Technical Report No: 2019-TR-05
GAME: Global Agile Model for Enterprises–Appendix

Mohammad Abdur Razzak

Lero–the Science Foundation Ireland Research Centre for Software

Faculty of Science and Engineering

Computer Science and Information System

University of Limerick

Submitted to the University of Limerick for the degree of

Doctor of Philosophy (PhD), 2020

05.02.2020

1. Supervisor: Prof Ita Richardson
Lero–the Science Foundation Ireland Research Centre for Software
University *of* Limerick
Ireland

2. Supervisor: Dr. Sarah Beecham
Lero–the Science Foundation Ireland Research Centre for Software
University *of* Limerick
Ireland

3. Supervisor: Dr. John Noll
Lero–the Science Foundation Ireland Research Centre for Software
University *of* Hertfordshire
United Kingdom

1. External examiner: Prof Tony Clear
Faculty of Design & Creative Technologies
Auckland University of Technology
New Zealand

1. Internal examiner: Prof Brian Fitzgerald
Lero–the Science Foundation Ireland Research Centre for Software
University *of* Limerick
Ireland

Day of the defense: 25 11 2019

Signature from head of PhD committee:

Contents

List of Tables

LIST OF TABLES

List of Figures

LIST OF FIGURES

Appendix A

Model in Global Software Development–Review protocol

A.1 Research Question

What are the process models (including capability, maturity, stage model, and frameworks) are in practice in globally distributed software development?

A.2 Search Terms

Table A.1: Search Terms.

Category	***Keywords***
Global software development	Distributed software development, Global software development, Collaborative software development, Global software engineering, Globally distributed work, Collaborative software engineering, Distributed development, Distributed teams, Global software teams, Globally distributed development, Geographically distributed software development, Offshore software development, Offshoring, Offshore, Offshore outsourcing, Dispersed teams
Process model	Process maturity, Process capability, Process evolution, Evolution sequence, Capability model, Maturity model
Agile	Scrum, Lean, XP, Kanban

A.3 Search String

(Global OR gsd OR gse OR DSD) AND (agile OR scrum AND plan-driven OR plan driven OR lean) AND (model OR framework OR method) AND (capability

OR process OR maturity)

A.3.1 Inclusion and Exclusion Criteria

A.3.1.1 Inclusion

- Publication year: 2010-2018
- Language: English
- Full text available and accessible
- Peer reviewed work
- Experience reports
- Answers one or more of the research questions
- Empirical studies and theoretical studies will be included if they meet the quality criteria.

A.3.1.2 Exclusion

- Exclude that not relate to the process of software development
- Exclude duplicated studies (where authors report similar results in two or more publications – e.g. a journal paper that is an extension of a conference paper). Exclude the least detailed paper, or if unclear exclude the paper that is published in the more notable venue.
- Exclude sources which did not discuss the concept of software development
- books, presentations, blogs

A. MODEL IN GLOBAL SOFTWARE DEVELOPMENT–REVIEW PROTOCOL

Appendix B

What do we know about scaling agile?–Review protocol

B.1 Research Question

What scaling agile process models or frameworks are used in practice?

B.2 Search String

("scaling agile") AND (framework OR method* OR model* OR mechanism OR principle* OR practice*)*

B.3 Where to search?

- IEEE Digital Library (Explore)
- ACM Digital Library
- Science Direct (Elsevier)
- Scopus

B.4 Source Selection

B.4.1 Inclusion and Exclusion Criteria

B.4.1.1 Inclusion

- Publication year: 1992-2018
- Language: English
- Full text available and accessible
- Peer reviewed work
- Experience reports
- Answers one or more of the research questions
- Empirical studies and theoretical studies will be included if they meet the quality criteria.

B.4.1.2 Exclusion

- Exclude that not relate to the process of software development
- Exclude duplicated studies (where authors report similar results in two or more publications – e.g. a journal paper that is an extension of a conference paper). Exclude the least detailed paper, or if unclear exclude the paper that is published in the more notable venue.
- Exclude sources which did not discuss the concept of software development
- books, presentations, blogs

B.5 Review Process

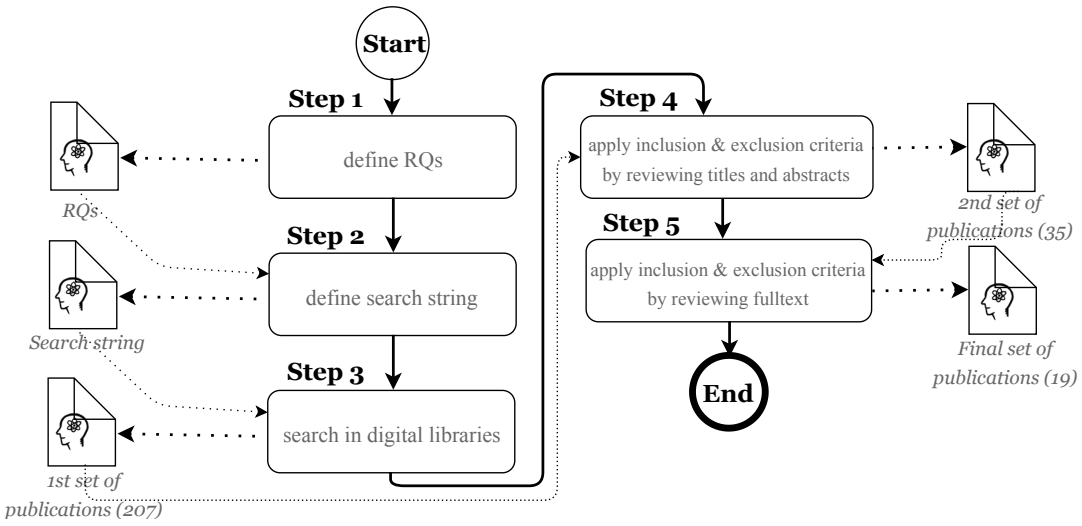

Figure B.1: Search Process

B.6 Final Set of peer-reviewed papers

1. Uludağ, Ö., Kleehaus, M., Xu, X., & Matthes, F. (2017, October). Investigating the role of architects in scaling agile frameworks. In Enterprise

Distributed Object Computing Conference (EDOC), 2017 IEEE 21st International (pp. 123-132). IEEE.

2. Ebert, C., & Paasivaara, M. (2017). Scaling agile. IEEE Software, 34(6), 98-103.

3. Putta, A. (2018, May). Scaling agile software development to large and globally distributed large-scale organizations. In Proceedings of the 13th Conference on Global Software Engineering (pp. 141-144). ACM.

4. Grewal, H., & Maurer, F. (2007, August). Scaling agile methodologies for developing a production accounting system for the oil & gas industry. In Agile Conference (AGILE), 2007 (pp. 309-315). IEEE.

5. Razzak, M. A., Richardson, I., Noll, J., Canna, C. N., & Beecham, S. (2018, May). Scaling agile across the global organization: an early stage industrial SAFe self-assessment. In 2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE) (pp. 116-125). IEEE.

6. Stojanov, I., Turetken, O., & Trienekens, J. J. (2015, August). A maturity model for scaling agile development. In Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on (pp. 446-453). IEEE.

7. Shameem, M., Kumar, C., Chandra, B., & Khan, A. A. (2017, December). Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective. In Software Engineering Conference Workshops (APSECW), 2017 24th Asia-Pacific (pp. 17-24). IEEE.

8. Heikkila, V., Rautiainen, K., & Jansen, S. (2010, September). A revelatory case study on scaling agile release planning. In Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on (pp. 289-296). IEEE.

9. Paasivaara, M. (2017, May). Adopting SAFe to scale agile in a globally distributed organization. In Global Software Engineering (ICGSE), 2017 IEEE 12th International Conference on (pp. 36-40). IEEE.

10. Pries-Heje, J., & Krohn, M. M. (2017, May). The safe way to the agile organization. In Proceedings of the XP2017 Scientific Workshops (p. 18). ACM.

11. Lal, R., & Clear, T. (2018, May). Enhancing product and service capability through scaling agility in a global software vendor environment. In Proceedings of the 13th Conference on Global Software Engineering (pp. 59-68). ACM.

12. Giray, G., Tüzün, E., Tekinerdogan, B., & Macit, Y. (2016, May). Systematic approach for mapping software development methods to the essence framework. In Theory-Oriented Software Engineering (TOSE), IEEE/ACM International Workshop on (pp. 26-32). IEEE.

13. Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. Journal of Software: Evolution and Process, 30(10), e1954.

14. Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M. (2018). Large-scale agile transformation at Ericsson: a case study. Empirical Software Engineering, 1-47.

15. IT governance in scaling agile frameworks, URL: `http://mkwi2018.leuphana.de/wp-content/uploads/MKWI_172.pdf`

16. Uludağ, Ö., Hauder, M., Kleehaus, M., Schimpfle, C., & Matthes, F. (2018, May). Supporting Large-Scale Agile Development with Domain-Driven Design. In International Conference on Agile Software Development (pp. 232-247). Springer, Cham.

17. Alqudah, M., & Razali, R. (2016). A review of scaling agile methods in large software development. International Journal on Advanced Science, Engineering and Information Technology, 6(6), 828-837.

18. Eckstein, J. (2016, May). Sociocracy: An Organization Model for Large-Scale Agile Development. In Proceedings of the Scientific Workshop Proceedings of XP2016 (p. 6). ACM.

19. Bick, S., Scheerer, A., & Spohrer, K. (2016, May). Inter-Team Coordination in Large Agile Software Development Settings: Five Ways of Practicing Agile at Scale. In Proceedings of the Scientific Workshop Proceedings of XP2016 (p. 4). ACM.

B.7 Scaling Agile Frameworks reported in different paper

Table B.1: List of Scaling Frameworks.

Paper ID	List of Framework
1	Crystal Family, DSDM, SoS, Enterprise Scrum, ASSF, LeSS, SAFe, DA, Spotify, Mega Framework, EADAGP, RAGE, Continuous Agile Framework, Scrum at Scale, Enterprise Transition Framwork, ScALed Agile Lean Development, eXponential Simple Continuous Autonomous Learning Ecosystem, Lean Enterprise Agile Framework, Nexus, FAST Agile
2	SoS, SAFe, LeSS, DAD, Lean Scalable Agility for Engineering (LeanSAFE)
3	SAFe, LeSS, DAD
4	SoS and eXtream Programming
5	SoS, LeSS, SAFe, DAD, Spotify, Nexus, Scrum at Scale
6	SAFe
7	SAFe, LeSS, DAD
8	SAFe
9	SAFe, LeSS, DAD
10	SoS, SAFe
11	DAD
12	Nexus
13	SoS, SAFe, LeSS, DAD, LeanSAFE, RAGE
14	SAFe, LeSS, DAD
15	DA, EA, EUP, laCoCa model, RAGE, SAFe, Scrum at Scale, Xscale, Crystal Family, DSDM, ES, FAST Agile, Goal Driven agile, LeSS, Nexus, Prince 2 Agile, SoS, Scrum Pattern Language for programs, Spotify, Matrix of Services, SCARE, SLIM
16	SAFe, LeSS, DAD
17	DAD, SAFe, LeSS, LeSS HUGE, Spotify, Nexus, RAGE
18	SoS, LeSS, SAFe, Nexus, LeSS HUGE
19	SAFe, LeSS, DAD

B.7 Scaling Agile Frameworks reported in different paper

Table B.2: Citation Frequency.

Paper ID	Framework Name	Paper ID	Total
1	Scaled Agile Framework (SAFe)	1,2,3,5,6,7,8,9,10,13,14,15,16,17,18,19	16
2	Large-Scale Scrum (LeSS)	1,2,3,5,7,9,13,14,15,16,17,18,19	13
3	Disciplined Agile Delivery (DAD)	1,2,3,5,7,9,11,13,14,15,16,17,19	13
4	Scrum of Scrums (SoS)	1,4,5,13,15,18	6
5	Nexus	1,5,12,15,17,18	6
6	Spotify	1,5,15,17	4
7	Recipes for Agile Governance in the Enterprise (RAGE)	1,13,15,17	4
8	Scrum at Scale	1,5,15	3
9	Crystal Family	1, 15	2
10	Dynamic Systems Development Method (DSDM)	1, 15	2
11	Enterprise Scrum	1, 15	2
12	AST Agile	1, 15	2
13	Lean SAFE	2, 13	2
14	LeSS HUGE	7, 18	2
15	Agile Software Solution Framework (ASSF)	1	1
16	Mega Framework	1	1
17	Enterprise Agile Delivery and Agile Governance Practice	1	1
18	Continuous Agile Framework	1	1
19	Enterprise Transition Framework	1	1
20	ScALeD Agile Lean Development	1	1
21	eXponential Simple Continuous Autonomous Learning Ecosystem	1	1
22	Lean Enterprise Agile Framework	1	1
23	XP	4	1
24	Enterprise Agile	15	1
25	Enterprise Unified Process (EUP)	15	1
26	laCoCa Model	15	1
27	XScale	15	1
28	Goal Driven Agile	15	1
29	PRINCE 2 Agile	15	1
30	Scrum Pattern Language of Programs (PloP)	15	1
31	Sustainable Cultural Agile Release in the Enterprise (SCARE)	15	1
32	Matrix of Services	15	1
33	Scrum Lean in Motion (SLIM)	15	1

B. WHAT DO WE KNOW ABOUT SCALING AGILE?–REVIEW PROTOCOL

Appendix C

GTM Recommendations and SAFe Practices

C.1 GTM Recommendations

Num	ID	Description
SG	1	Define Global Project Management
SP	1.1	Global Task Management,Goal: distribute tasks so that the advantages of GSE are leveraged and the negative factors which are inherent to its operation are minimised.
SSP	A	Determine team and organisational structure between locations.
R	A1	Create roles, relationships and rules to facilitate coordination and control over geographical, temporal and cultural distance.
R	A2	Structure global team and monitor operation to minimise fear and alienation in teams.
R	A2.1	Be aware of problems with unbalanced team sizes; e.g., smaller teams may be threatened and fear job loss.
R	A2.2	Team structure should cater for possibility of dual reporting to management at more than one location, e.g team structure could be cross divisional or multi-organisational and management remote.
R	A2.3	Ensure that the supervision, support and information needs of all team members are met regardless of location.
R	A2.4	Organisational structure should be documented and available to all team to allow a clear understanding of everyone's roles and responsibilities within the project.
R	B1	Identify and document reason for working with virtual team.
R	B2	Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.
R	B3	Retain tasks that require frequent communication between groups within collocated teams.
R	B4	Where GSE teams are subdivided into work modules (e.g. different parts of the life-cycle), management must allocate tasks based on core competencies of each sub-team, and clearly define which stages are carried out at which location.

R	B5	Confidential software development activities that provide competitive advantage should be developed within the organisation.
R	B6	Related non confidential development activities can be undertaken by external remote team colleagues.
SP	1.2	Knowledge and Skills
SSP	C	Identify business competencies required by global team members in each location
R	C1	Document and define customer base and functions relative to the application being developed.
R	C2	Provide training to ensure that global team has required understanding of the customer base and the business functions to take full advantage of the proximity of the team to the customer base.
SSP	D	Identify the cultural requirements of each local sub-team
R	D1	Cultural diversity: Each team member should be trained to understand the culture of the virtual team.
R	D2	Face-to-face meetings are recommended when and where possible, ideally at the start of the project and/or when a new member joins.
R	D3	Having individuals visit locations for extended periods can also be a successful strategy and should be fully leveraged at every possible opportunity.
SSP	E	Identify Communication Skills for GSE
R	E1	In order to develop the right practice, a new communication protocol needs to be set up. Policies should be put in place to support these new requirements to the satisfaction of all virtual team members. For example in synchronous communication, ensure that link up times are shared between core team working hours in each location.
SSP	F	Establish relevant criteria for training teams
R	F1	Effective knowledge transfer: Carry out evaluation of training needs to include cultural and linguistic issues.
R	F2	Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements.
R	F3	If working in regulatory domain, provide training on regulatory requirements and procedures.
SP	1.3	Global Project Management
SSP	G	Identify GSE project management tasks

R	G1	Define ability and potential productivity of team: Global project manager should allocate tasks and timescales that are realistic.
R	G2	Where possible, the project manager should be actively involved in the recruitment and selection of team members. Failing this, they should gather all information relating to the technical and professional experience of potential and existing team members.
R	G3	When teams are in place and project details reported project managers should understand and document how individuals contribute to that project along with their skills and knowledge.
R	G4	Plan for lessons learned to be recorded during the project and reviewed at the end by all project team members. Update procedures based on this experience
R	G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global change management agent.
SSP	H	Assign tasks to appropriate team members
R	H1	Assign according to one or more of three different approaches; Modularisation; Phase-based approach; and Integrated approach.
SSP	I	Ensure Awareness of cultural profiles
R	I1	National cultural differences should be identified and communicated to the management and team members. Cultural training can be communicated in following way:
R	I1.1	Provide training to give all team members an opportunity to learn and understand about each other's culture.
R	I1.2	Address national, religious and relevant ethnic issues, all team members should understand acceptable and unacceptable forms of behaviour.
R	I1.3	Training should be tailored to team member's specific needs and location.
R	I2	Project managers should ensure that cultural profiles for teams are established. E.g., Management and staff should show respect for gender-related cultural values of all colleagues. All employees' legal rights must be upheld.
SSP	J	Establish cooperation and coordination procedures between locations

R	J1	Ensure that a suitable infrastructure, process and management procedures are in place to help establish cooperation and coordination between locations.
R	J2	Achievable milestones should be planned and agreed. Projects should be monitored with reference to costs, time, productivity, quality and risk.
SSP	K	Establish reporting procedures between locations
R	K1	Regular formal reporting will help the project manager to remain aware of how project is progressing.
R	K2	Procedure should include and encourage team members to report whether or not they can take on that task in the given time and report any problems before it is too late.
SSP	L	Establish a Risk Management Strategy
R	L1	All potential risks should be identified and addressed to include: risks in misunderstanding cultural differences, misunderstanding requirements, feature volatility, schedules, budgets, personnel.
R	L2	In addition, risk associated with outsourcing activities to politically unstable locations needs to be identified.
R	L3	Identify and address local project demands that may pose a risk to the global project.
SG	2	Define Management Between Locations
SP	2.1	Operating Procedures
SSP	M	Define how conflicts and differences of opinion between locations are addressed and resolved
R	M1	Set up a strategy to handle, monitor and anticipate where conflict between remote locations may occur. The strategy should include how conflict will be resolved and how a person responsible for that resolution is selected.
R	M2	When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences.
R	M3	Establish criteria for global projects based on global strategy and vision to guide decision making.
SSP	N	Implement a communication strategy for the team
R	N1	Plan, facilitate, encourage and monitor communication between teams.

R	N2	Provide training on how best to communicate with remote colleagues, including the effective operation of communication tools and procedures.
R	N3	Consider linguistic and cultural implications inherent when communicating remotely.
SSP	O	Establish communication interface points between the team members
R	O1	Strategies need to be put in place which encourage both formal and informal reporting.
R	O2	Ensure that relevant team members are made aware of how and when they will receive inputs to products, needs to distribute outputs from and when complete work products are required.
R	O3	Ensure teams are aware of potential constraints such as legal restrictions and holidays in countries within which they are developing the product.
R	O4	Ensure that information about each team member is easily accessible by colleagues. Information of an individual's role within the team and their specific areas of responsibility should be combined with a photograph, their first name, surname, friendly name (if appropriate) and their preferred form of address.
R	O5	Intranets and wikis can be invaluable for this form of communication.
SSP	P	Implement strategy for conducting meetings between locations
R	P1	Identify appropriate virtual meeting technology is used.
R	P2	Try to ensure all participants are comfortable with virtual meeting and are given opportunity to agree or disagree with points raised, and offer new ideas.
R	P3	Circulate agenda prior to meeting, and clearly minute actions agreed a meeting
R	P4	Ensure that no delay occurs between the meeting and the circulation of minutes as people may be waiting for the minutes before implementing the actions.
SP	2.2	Collaboration between locations
SSP	Q	Identify common goals, objectives and rewards for the global team
R	Q1	Global Project manager sets project goals and objectives.
R	Q2	Goals at project level are common to all locations.

R	Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location.
R	Q4	The global team is viewed as an entity in its own right, regardless of the location of its team members and its performance should be judged and rewarded accordingly.
R	Q5	Acknowledging team success may require tailoring rewards to the needs of different cultures.
R	Q6	Project Managers need to understand the cultural motivation of the different team members and identify and apply appropriate rewards in each situation when and where relevant.
R	Q7	Consideration should be given to cultural issues, economic situation and income tax laws when planning rewards.
SSP	R	Collaboratively establish and maintain work product ownership boundaries
R	R1	Define product ownership boundaries through partitioning of work across GSE teams.
R	R2	Each location should understand their role within the life cycle of the product.
R	R3	Each location should understand how their modifications to the product unit can affect the other locations.
SSP	S	Collaboratively establish and maintain interfaces and processes
R	S1	Define common process goals across all locations.
R	S2	Define process ownership – placing ownership with those closest to process where possible.
R	S3	Seek and encourage input from team members at all locations.
R	S4	Let team members know their input to process development and ownership is valued.
R	S5	Processes should address specific challenges associated with GSE.
R	S6	Processes should take into account the relevant structures and procedures from all sites.
SSP	T	Collaboratively develop, communicate and distribute work plans.
R	T1	Achievable milestones should be planned and agreed.

R	T2	Within the commitments made, team members must explicitly include communication plans to include use synchronous and asynchronous communication tools.
R	T3	Contingency plans should be in place to address potential risks.
R	T4	Establish procedures to coordinate implementation of contingencies when and if required.
SG	3	Define Global Regulation Management
R	U	Sub Practice U. Establish a risk management strategy for regulation
R	V	Sub Practice V: Collaboratively plan, develop and validate systems and changes
R	W	Sub Practice W: Coordinate for traceability of requirements.

C.2 Team Level

Num	ID	Description
		Title: SAFe Team Practices V4 Author: Abdur Razzak and Dr. Sarah Beecham Specific Goal: Create a SAFe Team
1	[S4T1]	SP: Develop Agile Teams
1.1	[S4T2]	"Sub practice: Encourage intense collaboration within and between teams (recommendations):"
1.1.1	[S4T3]	Perform constant communication and collaboration
1.1.1.1	[S4T3.1]	Perform daily stand-ups
1.1.1.2	[S4T3.2]	Perform Iteration Planning before each iteration
1.1.1.3	[S4T3.3]	Perform Team Demo at the end of each iteration
1.1.1.4	[S4T3.4]	Perform retrospective at the end of each iteration
Moved	[S4T4]	Is now [S4T3.1] [S4T3.2] [S4T3.3] [S4T3.4]
1.1.2	[S4T5]	Manage dependencies and resolve impediments by continuously and actively engaging with other teams
1.1.3	[S4T6]	Facilitate trust among team members by having a common mission; common iteration goals and team Program Increment (PI) objectives
1.1.4	[S4T7]	Improve continuous collaboration by using regular feedback loops
1.2	[S4T8]	Sub-practice: Agree Agile Methodology
1.2.1	[S4T9]	Determine which methodology is used within the team: a combination of ScrumXP and Kanban and let each team know which method you are using
1.2.2	[S4T10]	For ScrumXP use the following practices
1.2.2.1	[S4T11]	Plan the Iteration
1.2.2.1.1	[S4T12]	Measure the team's capacity based on previous sprint to establish the velocity for the upcoming iteration
1.2.2.1.2	[S4T13]	Perform PI planning to understand and agree on one or more iteration goal(s) based on the team and PI Objectives
1.2.2.1.3	[S4T14]	Discuss each story in terms of relative difficulty; size; complexity and technical challenges; and establish acceptance criteria
1.2.2.1.4	[S4T15]	Break each story into tasks then estimate and identify dependencies of a specific task.
1.2.2.1.5	[S4T15.1]	Each team member takes responsibility for a specific task or tasks

1.2.2.1.6	[S4T15.2]	At end of session over committed members share out tasks with others who still have capacity
1.2.2.2	[S4T16]	Visualize the work
1.2.2.2.1	[S4T17]	Use big visual information radiator (BVIRs) to understand and track progress during iteration execution
1.2.2.2.2	[S4T18]	Use story board to visualize the stories and their progress throughout the iteration
1.2.2.2.3	[S4T19]	Create pull process within the iteration and continuously balance the work to increase throughput team by using Work-in-Process Limits (WIP)
1.2.2.3	[S4T20]	Perform daily Stand-up meetings for team coordination
1.2.2.3.1	[S4T21]	Perform Daily Stand-up to understand team's status; escalate problems; and get help from other team members
1.2.2.4	[S4T22]	Demonstrate value and process improvement
1.2.2.4.1	[S4T23]	Demonstrate completed story and summarize team's increment
1.2.2.4.2	[S4T24]	Perform Iteration Retrospective to identify way to improve
1.2.3	[S4T25]	For Kanban use the following practices
1.2.3.1	[S4T26]	Visualize Flow and Limiting WIP
1.2.3.1.1	[S4T27]	Build an approximation of team's current process and define some initial WIP limits
1.2.3.2	[S4T28]	Measure Flow
1.2.3.2.1	[S4T29]	Use objective measure that includes average lead time; WIP; and throughput by Cumulative Flow Diagram (CFD)
1.2.3.3	[S4T30]	Improve Flow with Classes of Service
1.2.3.3.1	[S4T31]	Use the mechanism of classes of service to help team to optimize theirs backlog items such as Standard; Fixed Date; and Expedite
1.2.3.4	[S4T32]	Estimate Work
1.2.3.4.1	[S4T33]	Estimate initial capacity to establish a common starting point for team estimation
1.2.3.4.2	[S4T34]	Calculate team derived velocity by multiplying the throughput by an average story size
1.2.3.4.3	[S4T35]	Break larger initiatives into stories to ease estimation of larger work items
1.2.4	[S4T36]	Agile teams are on the train
1.2.4.1	[S4T37]	Team plans and commits to a set of PI objectives together

1.2.4.2	[S4T38]	Team integrates and demonstrates together to deliver complex systems
1.2.4.3	[S4T39]	Team learn together and share best practices through inter-team communication supported by Communities of Practice (CoP).
2	[S4T40]	SP: Develop Team Backlog
2.1	[S4T41]	Establish team backlog by adding user stories, enabler stories and improvement stories
2.1.1	[S4T42]	Stories needed to fulfill features are identified written as enabler stories, estimated, and sequenced
2.1.2	[S4T43]	Local stories representing new functionality, refactors, defects, research spikes, and other technical debt are identified written as enabler stories, estimated, and sequenced
2.1.3	[S4T44]	Backlog on the release train will reflect certain stories in support of other teams' and stakeholders' objectives
2.2	[S4T45]	Optimize value delivery and system health with capacity allocation
2.2.1	[S4T46]	Balance the backlog of internally facing work with new user stories that deliver value
2.2.1.1	[S4T47]	Invest continuously in evolving the technical underpinnings of the solution as well as keeping existing customers happy with bug fixes and enhancements
2.2.2	[S4T48]	Sequence the work
2.2.2.1	[S4T49]	Compare value of three things: defects; refactors, redesigns, technology upgrades; new user stories
2.2.2.2	[S4T50]	Apply Capacity allocation technique to make a policy decision as to how much total effort apply to each of the three activities listed
2.2.2.3	[S4T51]	Select the highest-priority backlog items to implement in each iteration
2.2.2.4	[S4T52]	Use either size/value or Weighted Shortest Job First (WSJF) to prioritize the stories issued by product owner
2.3	[S4T53]	Perform backlog refinement
2.3.1	[S4T54]	Identify the mature stories that ready for implementation
2.3.2	[S4T55]	Identify upcoming stories (and features, as appropriate)
2.3.2.1	[S4T56]	Take a flow-based approach by having at least one team workshop per iteration
2.3.3	[S4T57]	Perform backlog refinement to underline the problems of current plan

3	[S4T58]	SP: Develop Program Increment (PI) objectives
3.1	[S4T59]	Build Team PI Objectives
3.1.1	[S4T60]	Build PI Objectives during PI Planning -ensuring the following is in place;
3.1.1.1	[S4T61]	Solid estimating and planning
3.1.1.2	[S4T62]	Well understood velocity
3.1.1.3	[S4T63]	Analysis of coming features
3.1.1.4	[S4T64]	Define stories for team backlog
3.1.1.5	[S4T65]	A synthesis of simple business terms
3.1.1.6	[S4T66]	Visibility of the Program Vision, new features, stories that need to be delivered
3.2	[S4T67]	Differentiate between Features and Objectives
3.2.1	[S4T68]	Identify features that can be delivered by individual teams
3.2.2	[S4T69]	Identify feature that requires collaboration
3.3	[S4T70]	Use Stretch objectives to provide a way to help that ensure the delivery time-box will be met
3.3.1	[S4T71]	Use stretch objectives to synchronize to delivery a cadence with capacity margins
3.3.2	[S4T72]	Team agree to do their best to deliver the both non-stretch and stretch objectives
3.3.3	[S4T73]	Stretch objectives are included in the capacity of the PI
3.3.4	[S4T74]	Allow 10-15% of the total capacity for the stretch objectives
3.3.5	[S4T75]	Constantly keep in mind that stretch objectives are used to identify what can be variable within scope of a plan
3.3.6	[S4T76]	Recognize stretch objectives are not the way for stakeholders to load the teams with more than they can do
3.4	[S4T77]	Write SMART Objectives (Specific, Measureable, Achievable, Realistic, Time-bound)
3.4.1	[S4T78]	(S) Provide a concise and simple description of the intended outcome (usually starting with an action verb)
3.4.2	[S4T79]	(M)easure what a team needs to do to achieve the team objectives (can be descriptive; yes or no; or quantitative; or within a range)
3.4.3	[S4T80]	(A) objectives should be achievable i.e; within the team's control and influence
3.4.4	[S4T81]	(R)ecognize factors that cannot be controlled
3.4.5	[S4T82]	(T)ime period for achievement must be within the PI, and all objectives must be scoped appropriately
3.5	[S4T83]	Communicate Business Value with Objectives

3.5.1	[S4T84]	Business owner assign business value to each of the teams individual objectives in face-to-face conversation with the teams
3.5.2	[S4T85]	Communicate strategy and context behind the weighting decisions
3.5.3	[S4T86]	Rank each objective according to the business owner ranking on a scale of 1 to 10
3.5.4	[S4T87]	Do not confuse business value with any other measure such as associated effort or total story point etc.
3.5.5	[S4T88]	Do not calculate business value but assign business value
3.5.6	[S4T89]	To serve as input to execution consideration
3.6	[S4T90]	Finalize the Team PI Objectives with business value and stretch objectives
3.7	[S4T91]	Commit to PI Objectives by agreeing to do everything in the team's power to meet the committed objectives
3.8	[S4T92]	Decrease excess WIP to reduce overhead and thrashing, and it increase productivity and velocity. [Shed Excess WIP with Realistic Objectives]
4	[S4T93]	SP: Develop Iteration
4.1	[S4T94]	Plan Iteration
4.1.1	[S4T95]	Establish velocity by quantifying team capacity
4.1.2	[S4T96]	Understand and agree on one or more iteration goals that are based on the team and PI planning objectives
4.1.3	[S4T97]	Story analysis and estimation
4.1.3.1	[S4T98]	Discuss each story by covering relative difficulty, size, complexity, technical challenges, and acceptance criteria
4.1.3.2	[S4T99]	Include enablers that could constitute infrastructure work, refactoring, research spikes, architectural improvement, and defects
4.1.3.3	[S4T100]	Prioritize and estimate the enablers
4.1.4	[S4T101]	Tasks
4.1.4.1	[S4T102]	Divide story into tasks
4.1.4.2	[S4T103]	Identify the best person(s) who would be the best to accomplish it
4.1.4.3	[S4T104]	Estimate how long it will take (typically in hours)
4.1.4.4	[S4T105]	Identify dependencies it may have on other tasks or stories
4.1.4.5	[S4T106]	Discuss with team if team members find themselves overcommitted
4.1.5	[S4T107]	Commitment

4.1.5.1	[S4T108]	Do not pull any story from team backlog if the team's collective capacity has been reached
4.1.5.2	[S4T109]	Agree on the final list of stories that will be achieved, and revisit and restate the iteration goals (Both product owner and team members)
4.2	[S4T110]	Execute Iteration
4.2.1	[S4T111]	Track iteration progress
4.2.1.1	[S4T112]	Visualize the status of the stories, defects, and other activities that the team is working o during the iteration
4.2.1.2	[S4T113]	Use big visible information radiator (BVIR) on a wall in the team room
4.2.1.3	[S4T114]	Use Kanban board for Kanban teams and story board for ScrumXP teams
4.2.1.4	[S4T115]	Use Agile Project Management Tools to capture stories and status, defects, test cases, estimates, actuals, assignments, burn-down chart
4.2.2	[S4T116]	Constant communication
4.2.2.1	[S4T117]	Use webcams, instant messaging, and other collaboration tools if teams are distributed
4.2.2.2	[S4T118]	Perform Daily Stand-up in front of the BVIR that highlights the stories
4.2.2.3	[S4T119]	Keep daily stand-up strictly time-boxed to 15 minutes
4.2.3	[S4T120]	Improving flow
4.2.3.1	[S4T121]	Managing Work-in-Progress (WIP)
4.2.3.1.1	[S4T122]	Apply WIP limits to prevent bottlenecks in development
4.2.3.1.2	[S4T123]	Adopt implicit WIP limit when team plan their own work (Usually Kanban teams)
4.2.3.1.3	[S4T124]	Adopt explicit WIP limit (ScrumXP team)
4.2.3.2	[S4T125]	Building Quality In
4.2.3.2.1	[S4T126]	Create high-quality system by adopting five quality and engineering practices prescribes in SAFe; Test-first, Continuous integration, Refactoring, Pair work, and collective ownership
4.2.3.3	[S4T127]	Continuously Accepting Stories
4.2.3.3.1	[S4T128]	Accept stories continuously to improve flow
4.2.3.3.2	[S4T129]	Perform demo as soon as the stories are ready
4.2.3.4	[S4T130]	Test Automation
4.2.3.4.1	[S4T131]	Adopt automated testing to quickly perform regression testing, enhancing continuous system-wide integration, refactoring, and maintenance
4.2.4	[S4T132]	Build stories serially and incrementally

4.2.4.1	[S4T133]	Avoiding the Intra-iteration Waterfall
4.2.4.1.1	[S4T134]	Avoid the tendency to Waterfall the iteration and instead ensure that they are completing multiple define-build-test cycle
4.2.4.2	[S4T135]	Building stories incrementally
4.2.4.2.1	[S4T136]	Enable a short feedback cycle and allows the team to operate with a smaller increment of the working systems for continuous integration and testing
4.2.5	[S4T137]	Focuses on Program execution
4.2.5.1	[S4T138]	Ensure that teams don't focus solely on local optimization
4.2.5.2	[S4T139]	SAFe teams plan together, integrate and demo together, and learn together
4.3	[S4T140]	Perform Team Demo
4.3.1	[S4T141]	Team show a tested increment of value to product owner
4.3.2	[S4T142]	Team receive feedback from product owner on what they produced
4.3.3	[S4T143]	Use outcome of meeting to shape the team backlog for the next iteration
4.3.4	[S4T144]	Rate stories either accepted or in need of refinement
4.3.5	[S4T145]	Participate in the integrated system demo
4.3.6	[S4T146]	Team integrate and evaluate as continuously as their system context allows
4.4	[S4T147]	Perform Iteration Retrospective as the check step for the overall iteration
4.4.1	[S4T148]	Team evaluates its process and any improvement stories it had from previous iteration
4.4.2	[S4T149]	Team identify problems and root causes as well as bright spots
4.4.3	[S4T150]	Team come up with improvement stories that enter the team for the next iteration
4.4.4	[S4T151]	Use iteration retrospective to drive program level changes to process either immediately or in the Inspect and Adapt workshop
4.4.5	[S4T152]	Refine the backlog before next planning to include the decision from demo and retrospective
4.5	[S4T153]	Product owner refactors and re-prioritize new and old backlog items
5	[S4T154]	SP: Develop Stories
5.1	[S4T155]	Develop Business feature by splitting into either user stories or enabler stories

5.1.1	[S4T156]	Develop user stories
5.1.1.1	[S4T157]	Develop user stories to express needed functionality (Replace the traditional requirement specification)
5.1.1.2	[S4T158]	Develop user stories that are value centric and focus on the user not the system
5.1.1.3	[S4T159]	Consider who is using the system and what specifically they doing with it and why they are doing it
5.1.1.4	[S4T160]	Consider the user might be the end user or a device or a another system
5.1.2	[S4T161]	Develop Enabler stories
5.1.2.1	[S4T162]	Develop enabler stories to develop technical functionality (stories that may not touch any end user)
5.1.2.2	[S4T163]	Develop enabler stories to support exploration, architecture, or infrastructure
5.1.2.3	[S4T164]	Use technical rather than user-centric language
5.1.3	[S4T165]	Demonstrate user and enabler stories by showing the artifacts produce or via UI stuff or mock
5.1.4	[S4T166]	Write Good Stories
5.1.4.1	[S4T167]	Use 3Cs Card, Conversation, Confirmation to write good story
5.1.4.1.1	[S4T168]	Use the Card to capture the statement of intent of the user story (Index card, sticky note or tool)
5.1.4.1.2	[S4T169]	Use the Conversation between the team customers user or other stakeholders to determine more detailed behaviors required to implement intent
5.1.4.1.3	[S4T170]	Use Conversation throughout the story life cycle to include backlog refinement planning implementation and demonstration
5.1.4.1.4	[S4T171]	Use Confirmation to provide the precision necessary to ensure the story is implemented correctly and cover both relevant functional and non-functional requirements
5.1.4.2	[S4T172]	Invest in Good stories
5.1.4.2.1	[S4T173]	Develop Independent (of all other stories), Negotiable (a flexible statement of intent, not a contract), Valuable (providing a valuable vertical slice to the Customer), Estimable (small and negotiable), Small (fits within an iteration), Testable (understood enough to know how to test it) stories
5.1.5	[S4T174]	Estimate Stories

5.1.5.1	[S4T175]	Estimate stories based on Volume –(How much is there?), Complexity – (How hard is it?), Knowledge - (What's known?), and Uncertainty – (What's not known?)
5.1.5.2	[S4T176]	Use estimating poker which combines expert opinion, analogy, and disaggregation for quick but reliable estimates (note that there are a number of other methods used as well).
5.1.5.3	[S4T177]	Measure Velocity
5.1.5.3.1	[S4T178]	Measure velocity to plan and limit WIP, so teams don't take on more stories than their prior velocity would allow
5.1.5.3.2	[S4T179]	Use velocity to estimate how long it takes to deliver larger epics, features, capabilities, and enablers, which are also estimated in story points
5.1.5.4	[S4T180]	Common starting baseline for estimation
5.1.5.4.1	[S4T181]	Estimate stories and velocity for the team, giving each team member eight points to start with (adjust for part-timers)
5.1.5.4.2	[S4T182]	Subtract one point for every team member vacation day and holiday
5.1.5.4.3	[S4T183]	Find a small story that would take about a half-day to code and a half-day to test and validate. Call it a one (1)
5.1.5.4.4	[S4T184]	Estimate every other story relative to that one (1)
5.1.6	[S4T185]	Split Stories
5.1.6.1	[S4T186]	Create smaller stories for faster, reliable implementation to reduce variability and manage risk
5.1.6.2	[S4T187]	Adopt any of the split story techniques e.g., Work flow steps, Business rule variations, Major effort, Simple/-complex, Variations in data, Data entry methods, Deferred system qualities, Operations (example: Create Read Update Delete, or CRUD), Use-case, scenarios, Break-out spike
6	[S4T188]	SP: Develop Innovation and Planning Iteration (IP)
6.1	[S4T189]	Understand the IP Iteration
6.1.1	[S4T190]	Consider allocating time for innovation and exploration
6.1.2	[S4T191]	Consider dedicating time for the PI System Demo, Inspect and Adapt workshop, PI Planning events, and backlog refinement
6.1.3	[S4T192]	Understand what is required for final integration of the Solution

6.1.4	[S4T193]	Understand requirements for working on technical infrastructure, tooling, and other systemic impediments
6.1.5	[S4T194]	Build in enablement of continuing education
6.2	[S4T195]	Allow time for Innovation
6.2.1	[S4T196]	Use IP iterations for research and design activities and hackathons
6.3	[S4T197]	Dedicate Time to PI Events
6.3.1	[S4T198]	Place PI system demo, inspect and adapt workshop, and PI planning in a dedicated IP iteration
6.4	[S4T199]	Integrate the Complete Solution
6.4.1	[S4T200]	Prepare a system demo and provide time for final testing of the Solution
6.4.2	[S4T201]	Final performance test includes Nonfunctional Requirements (NFR) testing, standards and security validations, user acceptance testing, final documentation, and any other readiness activities that are not feasible or economical to perform at every iteration.
6.4.3	[S4T202]	For hardware solutions and other tangible components that are more difficult to continuously integrate end to end, plan a full integration at the Program Level as well as the Value Stream Level during the IP iteration.
6.4.4	[S4T203]	Apply frequent integrations where possible, which–even when not fully end to end– can address specific aspects of the solution and validates the assumptions early
6.4.5	[S4T204]	Demo these integrations at the system demo at the end of every iteration
6.4.6	[S4T205]	Use the IP iteration as a placeholder for full, final solution integration that must happen at least once per PI
6.5	[S4T206]	Advance Development Infrastructure
6.5.1	[S4T207]	Lean delivery puts additional pressure on the development infrastructure, so in first instance try to address the following type of items in the course of each iteration:
6.5.1.1	[S4T208]	Erect new continuous integration environments within the development infrastructure
6.5.1.2	[S4T209]	Install and maintain new test automation frameworks
6.5.1.3	[S4T210]	Adopt project management tooling
6.5.1.4	[S4T211]	Consider upgrading and enhancing intra- and inter-team communications systems

6.5.2	[S4T212]	It may be more efficient to perform an upgrade or migration at a time when there isn't a critical demo just a few days away
6.6	[S4T213]	Enable Continuous Learning
6.6.1	[S4T214]	Adopt Communities of Practice to foster and support continuous learning
6.7	[S4T215]	Leverage the Build-in Estimation Buffer
6.7.1	[S4T216]	To address the situation when everyone is planned to full capacity (leaving no-one available to flex when problems occur) treat the one IP iteration as a guard band or estimating buffer
6.7.2	[S4T217]	Use these bands and buffers to allow teams extra time to respond to unforeseen events, delays in dependencies, and other issues
6.7.3	[S4T218]	Do not routinely use this time for completing work, as this defeats the primary purpose of the IP iteration, as innovation will likely suffer.
7	[S4T219]	SP: Develop Program Increment
7.1	[S4T220]	Execute the Program Increment
7.1.1	[S4T221]	PI Planning
7.1.1.1	[S4T222]	Teams estimate what will be delivered and when, and highlight their interdependencies
7.1.1.2	[S4T223]	Create the baseline for the integration and demo pull events by defining what will be built and demonstrated
7.1.1.3	[S4T224]	Detail what the ART will have ready for integration and demo at the end of the PI.
7.1.2	[S4T225]	Scrum of Scrums
7.1.2.1	[S4T226]	Continuously coordinate dependencies of the Agile Release Train and to provide visibility into progress and impediments by Release Train Engineer (RTE)
7.1.2.2	[S4T227]	Update progress toward Milestones, program PI objectives, and internal dependencies among the teams (by The RTE, Scrum Masters, and others (where appropriate)).
7.1.3	[S4T228]	PO Sync
7.1.3.1	[S4T229]	Shows visibility how well the ART is progressing toward meeting the program PI objectives
7.1.3.2	[S4T230]	Perform PO Sync to prepare for the next PI
7.1.4	[S4T231]	Release Management Meeting
7.1.4.1	[S4T232]	Provide governance for any upcoming Releases and also provide regular communication to management

7.1.4.2	[S4T233]	Team approve any scope, timing, or resource adjustments necessary to help ensure the release
7.1.5	[S4T234]	System Demo
7.1.5.1	[S4T235]	Provides feedback from the stakeholders about the efficacy and usability of the system under development
7.1.5.2	[S4T236]	Ensure that integration between teams on the same ART occurs on a regular basis and no less than every iteration
7.1.6	[S4T237]	Prepare next PI meeting
7.1.6.1	[S4T238]	Prepare for the upcoming PI as a continuous process, involving
7.1.6.2	[S4T239]	Management alignment and organizational readiness for planning
7.1.6.3	[S4T240]	Backlog readiness
7.1.6.4	[S4T241]	The actual logistics for the event–facility readiness
7.1.6.5	[S4T242]	Consider all three factors since any one of these can interfere with the potential outcome–an actual, specific, and committed PI
7.1.7	[S4T243]	Perform Inspect and Adapt
7.1.7.1	[S4T244]	To reflect, problem-solve, and take on improvement actions needed to increase the velocity, quality, and reliability of the next PI
7.1.8	[S4T245]	Pre- and Post-PI planning for the value stream
7.1.8.1	[S4T246]	Allows Agile Release Trains and Suppliers in large value streams to build an aligned plan for the next program increment
7.1.9	[S4T247]	Value stream increment and solution demo
7.1.9.1	[S4T248]	Build multiple increments of value, which accumulate into solution Capabilities through ARTs
7.1.9.2	[S4T249]	New capabilities must be designed, developed, tested, and validated holistically, along with the existing capabilities of the system
7.1.9.3	[S4T250]	Show progress that the solution has made during the past program increment to value stream stakeholders, Customers (or their internal proxies), and senior management
7.1.9.4	[S4T251]	Demonstrates its accomplishments in the past PI (Senior managers and stakeholders review the progress in the broader solution context)
7.1.9.5	[S4T252]	Make decisions about continuation, adjustment, or even cancellation of initiatives, as well as changes to the Budgets for the various value streams

8	[S4T253]	SP: Develop Cadence Principles
8.1	[S4T254]	Apply Cadence Principles
8.1.1	[S4T255]	Planning at regular Program Increment (PI) intervals limits variances to a single PI time-box, thereby increasing Agile Release Train and Value Stream predictability [Use a regular cadence to limit the accumulation of variance]
8.1.2	[S4T256]	In order to reliably meet PI objectives and planning iteration is unplanned and provides schedule margin [Provide sufficient capacity margin to enable cadence]
8.1.3	[S4T257]	If a feature doesn't make it into a PI (or release) and it remains high priority, its delivery can be anticipated to be on schedule in the next PI [Use cadence to make waiting time predictable]
8.1.4	[S4T258]	Short iterations help to control the number of stories in the iteration batch [Use a regular cadence to enable small batch sizes]
8.1.5	[S4T259]	PI planning, iteration planning, backlog refinement, inspect and adapt, architecture discussion, etc., all benefits from frequent meetings [Schedule frequent meetings using a predictable cadence]
8.2	[S4T260]	Apply Synchronization Principles
8.2.1	[S4T261]	Individual Agile Teams are aligned to common iteration lengths [Exploit economic of scale by synchronizing work from multiple projects]
8.2.2	[S4T262]	Teams plan with stretch objectives; these are sacrificed as necessary when plan meet reality [Capacity margin enables synchronization of deliverables]
8.2.3	[S4T263]	Value stream and program PI events synchronize customer feedback, resource and budget adjustments [Use synchronized events to facilitate cross-functional trade-offs]
8.2.4	[S4T264]	Teams are aligned to common time-boxes and similar batch sizes [To reduce queues, synchronize the batch size and timing of adjacent processes]
8.2.5	[S4T265]	Teams integrate and evaluate (at least) on iteration boundaries; program and value streams integrate and evaluate on PI boundaries. [Apply nested cadence harmonic multiples to synchronize work]
9	[S4T266]	SP: Develop Iteration Goals

9.1	[S4T267]	Develop iteration goals to align team members to a common purpose
9.2	[S4T268]	Develop iteration goals to align program teams to common PI Objectives and Manage dependencies
9.3	[S4T269]	Develop iteration goals to provide continuous management information
10	[S4T270]	SP: Build-in Quality
10.1	[S4T271]	Sub practices: Build in quality in the Software
10.1.1	[S4T272]	Perform Continuous Integration
10.1.1.1	[S4T273]	Implement feature and component level continuous integration
10.1.1.2	[S4T274]	Implement ART integration
10.1.1.3	[S4T275]	Solution Integration
10.1.1.3.1	[S4T276]	Synchronize with Supplier and Solution Context
10.1.1.4	[S4T277]	Consider challenges and trade-offs
10.1.1.5	[S4T278]	Enable continuous integration
10.1.1.5.1	[S4T279]	Common Cadence
10.1.1.5.2	[S4T280]	Infrastructure
10.1.1.5.3	[S4T281]	Engineering Techniques in Support of CI
10.1.1.5.4	[S4T282]	Support of the System Team
10.1.1.6	[S4T283]	Make Continuous Integration a Culture
10.1.1.6.1	[S4T284]	Integrate often
10.1.1.6.2	[S4T285]	Make integration results visible
10.1.1.6.3	[S4T286]	Make fixing a failing integration top priority
10.1.2	[S4T287]	Test-first
10.1.2.1	[S4T288]	Use the Agile Testing Matrix
10.1.2.2	[S4T289]	Apply Test-Driven Development
10.1.2.2.1	[S4T290]	Implement automated unit tests
10.1.2.2.2	[S4T291]	Implement component (subsystem) tests
10.1.2.3	[S4T292]	Use acceptance Test-Driven Development (ATDD)
10.1.2.3.1	[S4T293]	Write story acceptance tests
10.1.2.3.2	[S4T294]	Automate acceptance testing
10.1.2.3.3	[S4T295]	Use an acceptance test template/checklist
10.1.3	[S4T296]	Refactor
10.1.3.1	[S4T297]	Specify refactors using so that
10.1.3.2	[S4T298]	Split refactors using refactor splitting guidelines from Methods of splitting refactors table.
10.1.3.3	[S4T299]	Establish acceptance criteria for refactors
10.1.3.4	[S4T300]	Demonstrate refactors
10.1.3.5	[S4T301]	Adopt a refactoring culture
10.1.3.5.1	[S4T302]	Create a Design Community of Practice (CoP)

10.1.3.6	[S4T303]	Work in pairs
10.1.3.7	[S4T304]	Develop collective ownership
10.2	[S4T305]	Sub practice: Build in quality in the Firmware and Hardware
10.2.1	[S4T306]	Use exploratory early iterations
10.2.2	[S4T307]	try for early and frequent integration and testing of subsystems and systems including hardware/firmware
10.2.3	[S4T308]	Apply design verification to hardware/firmware designs

C.3 Program Level

Num	ID	Description
		Specific Goal: Define Program Level
1	[S4PR1]	SP: Define Vision
1.1	[S4PR2]	Define inputs to the solution vision
1.1.1	[S4PR3]	Fast and intimate feedback from customers
1.1.2	[S4PR4]	Contains some of the vision and destination for new elements
1.1.3	[S4PR5]	Indicates how the solution interact with the customer context
1.1.4	[S4PR6]	Direct and guide through value stream capabilities backlog to the vision
1.1.5	[S4PR7]	Continuous evolution of the architectural runway support current and near-term features
1.1.6	[S4PR8]	Strategic themes provides direction
1.1.7	[S4PR9]	Continuously communicate emerging requirements and opportunities back into the program vision through product owner
2	[S4PR10]	SP: Define Roadmap
2.1	[S4PR11]	Build the Program Increment (PI) Roadmap
2.1.1	[S4PR12]	Develop PI commitment
2.1.1.1	[S4PR13]	Teams commit to meet the program PI Objectives for the upcoming PI
2.1.2	[S4PR14]	Forecast PIs
2.2	[S4PR16]	Develop Long-Term Forecasting
2.2.1	[S4PR17]	Use Agile Story point physics to estimate larger initiatives at the Epic level
2.3	[S4PR18]	Estimate Longer-Term Initiatives
2.4	[S4PR20]	Avoid the Queue
2.4.1	[S4PR21]	Understand queuing theory and aware of the impact of long queues have on delivery time
3	[S4PR22]	SP: Define Program Metrics
3.1	[S4PR23]	Develop feature progress report
3.1.1	[S4PR24]	Represent the total number of stories planned for a feature
3.1.2	[S4PR25]	Represent number of stories completed for a feature
3.2	[S4PR26]	Develop program Kanban board
3.2.1	[S4PR27]	Analyze and prioritize the features prior to reaching a PI boundary appropriately

3.2.2	[S4PR28]	Develop acceptance criteria to guide a high-fidelity implementation
3.3	[S4PR29]	Develop program predictability measure
3.3.1	[S4PR30]	Measure Team PI Performance Report to calculate the program predictability measure (Team PI Performance Report compares actual business value achieved against planned business value)
3.4	[S4PR31]	Develop program performance metrics
3.4.1	[S4PR32]	Develop program performance by calculating Functionality and Quality at the end of each PI
3.5	[S4PR33]	Develop PI burn-down chart
3.5.1	[S4PR34]	Use iterations within the PI (horizontal axis) and amount of story points remaining at the start of each iteration (vertical axis) to develop PI burn-down chart
3.6	[S4PR35]	Develop cumulative flow diagram
3.6.1	[S4PR36]	Represent the progression of different areas such as Funnel, Backlog, Review, Analysis, Execution, and Done to develop cumulative flow diagram
4	[S4PR37]	SP: Develop Milestones
4.1	[S4PR38]	Develop PI Milestones
4.1.1	[S4PR39]	Create an objective measure of progress
4.1.2	[S4PR40]	Measure, assess, and evaluate the system by the relevant stakeholders throughout the development
4.2	[S4PR41]	Develop Learning Milestones
4.2.1	[S4PR42]	Provide the necessary meaning to understand the feasibility of the solution and frame the right set of capabilities
4.2.2	[S4PR43]	Plan milestones incrementally
4.3	[S4PR44]	Manage Fixed-Date Milestones
4.3.1	[S4PR45]	Use program backlog prioritization via WSJF
4.4	[S4PR46]	Other Milestones
4.4.1	[S4PR47]	Fill patent, certify the system, audit certain regulatory requirements for economic success of product development
5	[S4PR48]	SP: Develop releases
5.1	[S4PR49]	Release value more frequently
5.1.1	[S4PR50]	Release more frequently to gain the meaningful feedback about efficacy, deployability, and usability of the product (Each release helps assess value of product in the development environment)
5.2	[S4PR51]	Develop on Cadence, Release Any Time

5.2.1	[S4PR52]	Support the goals of synchronization, alignment, management of variability, and predictability of development velocity
5.2.2	[S4PR53]	Deliver the product whenever business needed
5.3	[S4PR54]	Building the Release
5.3.1	[S4PR55]	Develop Team Increment
5.3.1.1	[S4PR56]	Complete user stories from Team Backlog and assure that each story meets its (local) Definition of Done
5.3.1.2	[S4PR57]	Apply Continuous Integration practices with test automation whenever feasible to monitor and ensure progress
5.3.2	[S4PR58]	Build System Increment
5.3.2.1	[S4PR59]	Build a system increment every two weeks by summing up of all the team backlog items completed by all release train teams during current and all previous iterations
5.3.2.2	[S4PR60]	Perform system demo after each system increment
5.3.2.3	[S4PR212]	Perform verification and validation of key scenarios and nonfunctional aspects- usually with the help of the System Team
5.3.2.4	[S4PR213]	Perform a final system demo of all features developed in the PI during the Inspect and Adapt workshop at the PI boundary
5.3.3	[S4PR61]	Define Solution Increment [Ignored as it is part of Value Stream Level]
5.3.4	[S4PR62]	Define Release
5.3.4.1	[S4PR63]	Perform solution verification and validation as well as supporting activities before actually release to customers
5.3.4.2	[S4PR214]	Transition solution assets to deployment organization for actual delivery to customers
5.3.4.3	[S4PR215]	[Design] solution with 'release-ablity' in mind reflecting key release concerns and constraints
5.3.4.4	[S4PR216]	[Validate 'release-ability'] assumptions as early and frequently as possible using external releases at every PI at least
5.4	[S4PR64]	SP: Define scaled definition of done
5.4.1	[S4PR65]	Continuously buildup of system functionality, verification and validation of the elements of the solution as well as final solution itself, which must reflect in a scaled definition of done
6	[S4PR66]	SP: Develop Program Kanban

6.1	[S4PR67]	Develop program epics selection
6.1.1	[S4PR68]	Include all big initiatives in the program epic funnel as there is no WIP limit at this step
6.1.2	[S4PR69]	Perform initial exploration of epics and rank them roughly by using Weighted Shortest Job First (WSJF) to determine which epics should move to the next step for deeper exploration
6.1.3	[S4PR70]	Explore the epic in more depth; refine size and WSJF compared to other epics in this step; consider solution alternatives; identify possible paths of the incremental implementation strategy; and determine the costs involved, technology and architectural enablement, infrastructure, etc.
6.2	[S4PR71]	Develop feature section
6.2.1	[S4PR72]	Include all features into the feature funnel. These may include new functionality or enhancement of the existing system functions. New backlog items that enhance system qualities or produce architectural or infrastructure enablement also originate here.
6.2.2	[S4PR73]	Move features that align with the vision and support strategic themes to feature refinement for further exploration. General sizing of features also occurs at this step, as features are estimated in normalized story points. (The purpose of such estimation is to support economic estimating and forecasting of value delivery over a longer period of time based on scope.)
6.2.3	[S4PR74]	Elaborate and approve highest-priority features then move to program backlog where they are prioritized with WSJF relative to the rest of the backlog.
6.2.4	[S4PR75]	Pull top features from program backlog and moves them into the implementing step through the PI planning process, where selected features get broken down into stories and subsequently implemented by teams during the PI.
7	[S4PR76]	SP: Develop Program Backlog
7.1	[S4PR77]	Refine the backlog
7.1.1	[S4PR78]	Review and update backlog item definition and developing acceptance criteria
7.1.2	[S4PR79]	Work with the team to establish technical feasibility and scope estimates
7.1.3	[S4PR80]	Analyze ways to split backlog items into smaller chunks of incremental value

7.1.4	[S4PR81]	Determine the Enablers spawned by new features and capabilities, and establishing their capacity allocation
7.2	[S4PR82]	Prioritize the Backlogs
7.2.1	[S4PR83]	Apply Weighted Shortest Job First prioritization method for job sequencing
7.3	[S4PR84]	Prepare for PI planning
7.3.1	[S4PR85]	Prepare backlog, update vision briefings, and work with product owners to further socialize the backlog
7.3.2	[S4PR86]	Update enabler definitions and models to develop use cases that illustrate how the features and capabilities work together to deliver the end user value
7.4	[S4PR87]	Optimizing value and solution integrity with capacity allocation
7.4.1	[S4PR88]	Apply capacity allocation to make a decision about how much of the total effort can be applied for each type of activity for an upcoming PI. The decision can be revisited as part of backlog refinement in preparation of each PI planning
7.4.2	[S4PR89]	Establish an agreement to determine how the work is perform for each activity type
7.5	[S4PR90]	Define backlogs, Queues, Little's Law, and Wait times
7.5.1	[S4PR91]	Actively manage the backlog and keep them short by applying Little's Law
7.5.2	[S4PR92]	Limit the commitments to longer-term work, because some other item may come along that's more important than a prior commitment
8	[S4PR93]	SP: Define Weighted Shortest Job First (WSJF)
8.1	[S4PR94]	Calculate the Cost of Delay
8.1.1	[S4PR95]	Identify user-business value
8.1.2	[S4PR96]	Identify time criticality
8.1.3	[S4PR97]	Identify risk reduction-opportunity enablement value
8.2	[S4PR98]	Calculate Job Duration
8.2.1	[S4PR99]	Use job size to calculate the duration
9	[S4PR100]	SP: Develop PI planning
9.1	[S4PR101]	Develop activities that benefits business
9.1.1	[S4PR102]	Establish high-bandwidth communication across all team members and stakeholders
9.1.2	[S4PR103]	Build social network the ART depends upon
9.1.3	[S4PR104]	Align development to business via business context, vision, and Team and Program PI Objectives

9.1.4	[S4PR105]	Identify dependencies and forecast cross-team and cross-ART coordination
9.1.5	[S4PR106]	Provide the opportunity for just the right amount of Architecture and User Experience guidance
9.1.6	[S4PR107]	Match demand to capacity and eliminate excess WIP
9.1.7	[S4PR108]	Accelerate decision-making
9.2	[S4PR109]	Define PI planning's primary outputs
9.2.1	[S4PR110]	Create a set of SMART team PI objectives for each individual team with business value assigned
9.2.2	[S4PR111]	Create a Program board that highlights the new features
9.2.3	[S4PR112]	Vote of confidence/commitment from the entire program to these objectives
10	[S4PR113]	SP: Develop features and benefits matrix (Capabilities in the value stream level: Epics>Capabilities>Feature>Stories)
10.1	[S4PR114]	Create and Manage Features
10.1.1	[S4PR115]	Product manager create features in collaboration with product owner and other key stakeholders. Features are also created as a result of decomposition of epics
10.1.2	[S4PR116]	Create enabler features by architects or engineers to pave the architectural runway which is maintained in the program backlog alongside business features
10.1.3	[S4PR117]	Apply capacity allocation for enabler to work as a whole or to differentiate between various types of enablers
10.2	[S4PR118]	Prioritize Features
10.2.1	[S4PR119]	Apply WSJF for continuous prioritization of features in the program backlog
10.3	[S4PR120]	Estimate Features
10.3.1	[S4PR121]	Apply normalized estimation technique during the backlog refinement or equivalent approach used by the Agile Teams to estimate stories
10.3.2	[S4PR122]	Involve a subject matter expert in basic exploration and sizing
10.4	[S4PR123]	Accept Features
10.4.1	[S4PR124]	Develop acceptance criteria to determine whether the implementation is correct and delivers the business benefits
11	[S4PR125]	SP: Develop enabler features
11.1	[S4PR126]	Create and manage enablers
11.1.1	[S4PR127]	Create exploration enablers to validate the need or solution

11.1.2	[S4PR128]	Create architectural enablers to pave the runway
11.1.3	[S4PR129]	Create infrastructure enablers to be ready to develop, test, and integrate the initiatives
11.2	[S4PR130]	Use enablers
11.2.1	[S4PR131]	Apply enabler for exploration that provides a way for development teams to flesh out the details of requirements and design
11.2.2	[S4PR132]	Use architectural enablers to fix existing problems with solution such as enhance performance
11.2.3	[S4PR133]	Develop infrastructure in different level of SAFe to support frequent or continuous integration and testing
11.3	[S4PR134]	Implement architectural enablers
11.3.1	[S4PR135]	Break down the architectural enabler into small enabler stories that can fit in iterations
11.3.2	[S4PR136]	Organize planning enabler such that the system can run for most of the time on the old architecture or infrastructure
12	[S4PR137]	SP: Perform system demos
12.1	[S4PR138]	Timing of the System Demo
12.1.1	[S4PR139]	Perform system demo as near as possible to the end of the iteration
12.2	[S4PR140]	Balance integration effort and feedbacks
12.2.1	[S4PR141]	Integrate a subset of capabilities, components, or subsystems
12.2.2	[S4PR142]	Integrate to illustrate a particular feature, capability, or Nonfunctional Requirements
12.2.3	[S4PR143]	Integrate with the support of prototypes and mock-ups
12.2.4	[S4PR144]	Integrate every other iteration
12.3	[S4PR145]	Process and Agenda
12.3.1	[S4PR146]	Briefly review the business context and the PI Objectives
12.3.2	[S4PR147]	Briefly describe each new feature that will be demonstrated
12.3.3	[S4PR148]	Demonstrate each new feature in an end-to-end use case
12.3.4	[S4PR149]	Allow questions and comments by opening a forum
12.3.5	[S4PR150]	Identify current risks and impediments
12.3.6	[S4PR151]	Summarize progress, feedback, and action items
13	[S4PR152]	SP: Perform inspect and adapt (I&A) workshop
13.1	[S4PR153]	Perform PI System Demo
13.1.1	[S4PR154]	Show all the accumulaed features that have accrued over the course of the PI

13.1.2	[S4PR155]	Make it more formal and some additional preparation and staging is usually required
13.1.3	[S4PR156]	Engage stakeholders and provide feedback–time-boxed to an hour or less
13.2	[S4PR157]	Perform Qualitative measurement
13.3	[S4PR159]	Perform Retrospective and problem-solving workshop
13.3.1	[S4PR160]	Run a brief retrospective to identify whatever issues team would like to address
13.3.2	[S4PR161]	Perform a structured, root cause analysis-based, problem-solving workshop for large, systematic program-level problems
13.3.2.1	[S4PR162]	Agree on the problem(s) to solve
13.3.2.2	[S4PR163]	Perform root cause analysis
13.3.2.3	[S4PR164]	Identify the biggest root cause
13.3.2.4	[S4PR165]	Restate the new problem
13.3.2.5	[S4PR166]	Brainstorm solution
13.3.2.6	[S4PR167]	Create improvement backlog items
14	[S4PR168]	SP: Release any time
14.1	[S4PR169]	Continuously Deliver
14.2	[S4PR170]	Separating development concerns from release concerns
14.2.1	[S4PR171]	Release on the program increment cadence
14.2.2	[S4PR172]	Release less frequently
14.2.3	[S4PR173]	Release more frequently
15	[S4PR174]	SP: Develop architectural runway
15.1	[S4PR175]	Intentional Architecture Support the Bigger Picture
15.1.2	[S4PR177]	Provide guidance for cross-team design and implementation synchronization
15.1.3	[S4PR178]	These enablers create architectural runway needed to allow teams to deliver business value faster and more reliably
15.2	[S4PR179]	Enable Flow and Agility with Architecture
15.2.1	[S4PR180]	Split enabler epics into enabler features and/or capabilities, which are ultimately implemented by individual ARTs
15.2.2	[S4PR181]	Complete each enabler within a PI
15.3	[S4PR182]	Build the Architectural Runway
15.3.1	[S4PR183]	Continuously maintain and extend architectural runway to support stable velocity
15.3.2	[S4PR184]	Use capacity allocation to ensure continuous investments in enablers
16	[S4PR185]	SP: Organize Agile Release Train

16.1	[S4PR186]	Organizing Agile Release Train
16.1.1	[S4PR187]	Ensure Train Size is effective
16.1.1.1	[S4PR188]	Develop train size of between 50-125 people. However, trains with fewer than 50 people are also valid (found to be effective and provide many advantage over legacy Agile practices for coordinating Agile Teams)
16.1.2	[S4PR189]	ARTs Organization Depends on Value Stream Size
16.1.2.1	[S4PR190]	Multiple value streams can be realized by a single ART (This is usually the exceptional case; it tends to occur in smaller enterprises, or in enterprises that build a large number of products with a modest number of people)
16.1.2.2	[S4PR191]	Realize the value stream by a single ART (This is a common case and the easiest way to manage, as the ART and the value stream are one and the same)
16.1.2.3	[S4PR192]	Large value stream requires multiple ARTs (This is also quite a common case for those building systems, where additional work in required)
16.2	[S4PR193]	Splitting Large Value Streams
16.2.1	[S4PR194]	Organizing trains around capabilities and subsystems
16.2.1.1	[S4PR195]	Optimize capability ARTs for value flow and delivery speed
16.2.1.2	[S4PR196]	Optimize subsystem ARTs for architectural robustness, critical components, or components that are used by many other elements.
16.2.2	[S4PR197]	Organizing Teams on the Train
16.2.2.1	[S4PR198]	Develop a feature team that is organized around use-centered functionality. Each team is capable to deliver end-t-end value. Feature teams operate primarily with user stories, refactors, and Spike.
16.2.2.2	[S4PR199]	Develop a components team whose primary area of concern is restricted to a specific component or set of components.
17	[S4PR200]	SP: Develop an effective Solution (Ignored as it is part of Value Stream level)
17.1	[S4PR201]	Define solution capabilities, enablers, and NFRs
17.1.1	[S4PR202]	Use capabilities as end-to-end solution services that support the achievement of user goals
17.1.2	[S4PR203]	Enablers provide for exploration of new capabilities, contribute to solution infrastructure and architecture, and enhance NFRs
17.2	[S4PR204]	Define solution intent

17.2.1	[S4PR205]	Ignored
17.3	[S4PR206]	Define customer and solution context
17.3.1	[S4PR207]	Ignored
17.4	[S4PR208]	Define solution integration, testing, and demo
17.4.1	[S4PR209]	Adopt continuous integration and testing in order to progress configuration management, automation, and virtualization
17.5	[S4PR210]	Build an economically viable solution

C.4 Value Stream Level

Num	ID	Description
		Specific Goal: Define Value Stream Level
1	[S4VS1]	SP: Develop Economic Framework
1.1	[S4VS2]	Lean-Agile Budgeting
1.1.1	[S4VS3]	Provide fund to value streams, not project
1.1.2	[S4VS4]	Control investment over the course of PIs
1.1.3	[S4VS5]	Adjust value stream budget over time at PI boundaries, based on the relative value that each value stream provides to the portfolio
1.2	[S4VS6]	Epic Funding and Governance
1.2.1	[S4VS7]	Allocate fund to the value streams
1.2.2	[S4VS8]	Develop a lightweight business case and an explicit approval process for each epic
1.2.3	[S4VS9]	Establish thresholds by Program Portfolio Management (PPM) authority
1.3	[S4VS10]	Decentralize Economic Decision-Making
1.3.1	[S4VS11]	Empower people particularly Product and Solution Management with the relevant context, knowledge, and authority to make content decision at each level
1.3.2	[S4VS12]	Collaborate with large stakeholder community to determine the best course of action
1.4	[S4VS13]	Job Sequencing based on Cost of Delay
1.4.1	[S4VS14]	Adopt flow-based system economics that optimized by job-sequencing rather than by theoretical job ROI, or first-come, first-served job selection
1.4.2	[S4VS15]	Pull jobs into implementation based on WSJF, whereas estimate of job size is typically used as a proxy for duration
2	[S4VS16]	SP: Define Solution Intent
2.1	[S4VS17]	Provides a single source of truth as to the intended and actual behavior of the solution
2.2	[S4VS18]	Records and communicate requirements, design, and system architecture decisions
2.3	[S4VS19]	Facilitate further exploration and analysis activities
2.4	[S4VS20]	Aligns the customers, they system builders, and support to a common purpose
2.5	[S4VS21]	Supports compliance and contractual obligations
2.6	[S4VS22]	Nature of solution intent

2.6.1	[S4VS23]	Ensure to builders of complex system what exactly current system does now, and what changes are intended for a future state
2.6.2	[S4VS24]	Capture the knowledge of current and future state by including three primary elements: specification, design, and tests.
2.6.3	[S4VS25]	Adopt requirements and modeling tools such as Model-Based System Engineering (MBSE)
2.6.3.1	[S4VS26]	Support Leaning Cycles with Models
2.6.3.1.1	[S4VS27]	Facilitate early learning by testing and validating system characteristics, properties, and behaviors early, enabling early feedback on design decisions
2.6.3.1.2	[S4VS28]	Provides a different perspective into one or more system characteristics that enable the creation of Capabilities and Features
2.6.3.1.3	[S4VS29]	Predict performance (response time, reliability) or physical properties (heat, radiation, strength), or they may explore designs for user experience or response to an external stimulus
2.6.3.2	[S4VS30]	Use Traceability for Impact Analysis and Compliance
2.6.3.2.1	[S4VS31]	Simplifies and automates most regulatory and contractual compliance needs
2.6.3.2.2	[S4VS32]	Reduces time and effort for impact analysis needed to support new work and provides information for Inspect and Adapt activities
2.6.3.2.3	[S4VS33]	Facilitates cross-discipline collaboration by allowing traceability from a model in one discipline to a model in another
2.6.3.2.4	[S4VS34]	Encourages general knowledge discovery by making information, and related cross-discipline information, more accessible to teams
2.6.3.3	[S4VS35]	Record Models in Solution Intent
2.6.3.3.1	[S4VS36]	Organize and link information regarding different kinds of models
2.6.3.3.2	[S4VS37]	Specify the model information and organize to ensure its quality
2.6.3.3.3	[S4VS38]	Populate models with respective knowledge and information
2.6.3.3.4	[S4VS39]	Assign a model owner role (often a member of Systems Engineering), who is responsible for model content and structural integrity

2.6.3.4	[S4VS40]	Generate Documentation for Compliance
2.6.3.4.1	[S4VS41]	Generate documents for regulatory compliance (FAA, FDA, etc.) or contractual obligations (CDRLs in government contracting) when required
2.6.3.4.2	[S4VS42]	Generate document for stakeholders with different system perspectives or architectural framework standards (e.g., DoDAF, MODAF) that defines multiple stakeholder viewpoints
2.6.3.4.3	[S4VS43]	Generate documents to ensure consistency across all stakeholder views
2.6.3.4.4	[S4VS44]	Collaborate (System Engineers) with Customers and/or regulatory agencies on the minimum set sufficient to meet their obligations
2.6.3.4.5	[S4VS45]	Store information into repositories that can be used for inspections and formal reviews
2.6.3.5	[S4VS46]	Build Model Quality In
2.6.3.5.1	[S4VS47]	Define (by Model Owner) quality practices–model standards and model testing–and help to ensure that they are followed
2.6.3.5.2	[S4VS48]	Adopt quality practices that allow engineers to confidently and frequently make model changes and contribute to the system intent
2.6.3.5.3	[S4VS49]	Define document template early, if documents are being generated from the models as they influence many of these decisions
2.6.3.5.4	[S4VS50]	Run document generation tests early (against this configuration early) to ensure completeness. So, the teams do not have to rework (waste) the model with missed information
2.6.3.6	[S4VS51]	Create Testable and Executable Models
2.6.3.6.1	[S4VS52]	Adopt Test-First practices to help teams to build quality into their products early and facilitate the continuous and small changes
2.6.3.6.2	[S4VS53]	Allow developers to reliably make changes without causing an error in another part of the system by creating a rich suite of test cases
2.6.3.6.3	[S4VS54]	Testing Requirements Models
2.6.3.6.3.1	[S4VS55]	Use textual requirements in every system which typically reviewed manually

2.6.3.6.3.2	[S4VS56]	Seek executable specifications (ATDD- Acceptance Test-Driven Development), where requirements are specified in a form that can be executed for testing
2.6.3.6.3.3	[S4VS57]	Make requirements and tests one and the same where possible, and automate to the extent possible
2.6.3.6.4	[S4VS58]	Testing Analysis and Design Models
2.6.3.6.4.1	[S4VS59]	Apply static model analyzers or checkers that examine the model looking for anomalies
2.6.3.6.5	[S4VS60]	Models can also be dynamically tested
2.6.3.6.5.1	[S4VS61]	Use build-in solutions for assessing quality and leveraged as part of testing practice
2.6.3.6.6	[S4VS62]	Testing Traceability
2.6.3.6.6.1	[S4VS63]	Comply models with the linking structure to ensure proper queries, document generation, and compliance
2.7	[S4VS64]	Define the Dynamic Nature of Solution Intent
2.7.1	[S4VS65]	Adopt different approach that supports understanding the known and allows the unknowns to emerge over the course of development
2.7.2	[S4VS66]	Support and evolve solution intent throughout the entire development process
2.8	[S4VS67]	Define Fixed and Variable Solution Intent
2.8.2	[S4VS69]	Adopt variable intent that represents the elements for which system builders are free to explore the economic trade-offs of requirements and design alternatives that could meet the need
2.9	[S4VS70]	Develop Solution Intent
2.9.1	[S4VS71]	Provide sufficient guidance to the teams for initial PI Planning and execution
2.9.2	[S4VS72]	Use Features, capabilities, Stories, and Enablers to further define and realize the solution behavior
2.9.3	[S4VS73]	Invest in solution intent documentation in more complex and/or regulated environment
2.1	[S4VS74]	Collaborate on Solution Intent
2.10.1	[S4VS75]	Establish the solution intent's organizational structure and define where various types of information are managed to support analysis and compliance needs
2.10.2	[S4VS76]	Coordinate among Solution Management and Solution Architect/Engineering to delegate some solution intent requirements directly to the ARTs that build the capabilities and subsystems of the solution

2.10.3	[S4VS77]	Provide feedback through ARTs to solution-level decisions and raise issues and ideas to solution engineering
2.11	[S4VS78]	Moves from variable to Fixed Solution intent
2.11.1	[S4VS79]	Requires system builders to eventually know exactly what the system does and validate that it does exactly that and nothing else
2.11.2	[S4VS80]	Reuse elements from previous systems
2.12	[S4VS81]	Develop System of Systems Intent
2.12.1	[S4VS82]	Participate in a higher-level system of systems that provides system builders with unique knowledge and solution elements that accelerate development
2.13	[S4VS83]	Develop Minimal but Sufficient Documentation
2.13.1	[S4VS84]	Favor models over documents can provide a more easily maintained way to managing solution intent
2.13.2	[S4VS85]	Participate in the creation, feedback, and refinement of solution intent information
2.13.3	[S4VS86]	Adopt set-based design practices to avoid committing too early to design and requirements
2.13.4	[S4VS87]	Record any requirements and design decisions in only one place, a single source of truth that serves as the repository of record for all
2.13.5	[S4VS88]	Communicate at as high a level of abstraction as possible and decentralize requirements and design decision-making authority
2.13.6	[S4VS89]	Record what is needed because solution intent is the means to the end of building a product and meeting compliance and contractual obligation
3	[S4VS90]	SP: Develop Value Stream Kanban [The value stream Kanban generally repeats the structure and flow of the program Kanban system, so it won't be described further here. However, it operates with capabilities and value stream epics, respectively. The value stream Kanban is managed by the Solution Management team and supported by Solution Architects. Involvement of portfolio stakeholders is necessary to approve value stream epics.]
3.1	[S4VS91]	Synchronous Supporting Ceremonies
3.1.1	[S4VS92]	Epic Specification Workshop
3.1.1.1	[S4VS93]	Pull epics from funnel to review, others from review to analysis

3.1.1.2	[S4VS94]	Make a decision of a go or no-go followed by resultant features and capabilities being transitioned to the downstream part of the Kanban
3.1.2	[S4VS95]	Perform Value Stream Backlog Refinement
3.1.2.1	[S4VS96]	Build the backlog of capabilities
3.1.2.2	[S4VS97]	Size and elaborate the business benefits and acceptance criteria are performed at this time
3.1.2.3	[S4VS98]	Apply WSJF prioritization to sequences the features and capabilities that can be pulled into the backlog
3.1.3	[S4VS99]	PI Planning
3.1.3.1	[S4VS100]	Perform Pre-PI Planning session at the value stream level, where capabilities from the value stream Kanban are split into features and fed into the program Kanban systems for each train in the value stream
3.1.3.2	[S4VS101]	Bring the resultant features to their program Kanban system and place them in the feature funnel step
3.1.3.3	[S4VS102]	Bring back up to the value stream level (Post-PI Planning) for better understanding of which capabilities made it into the implementation
3.1.4	[S4VS103]	Solution/System Demo
3.1.4.1	[S4VS104]	Allow stakeholders to assess the amount of completed work and value delivered
4	[S4VS105]	SP: Develop Value Stream Capability
4.1	[S4VS106]	Use a similar matrix as the feature and benefits matrix and have associated acceptance criteria
4.2	[S4VS107]	Originate in the local value stream context or are derived from epics
4.3	[S4VS108]	Size to fit in a PI
4.4	[S4VS109]	Approve using the Value Stream Kanban and maintain approved capabilities in the value stream backlog
4.5	[S4VS110]	Associate enablers to describe and bring visibility to all the technical work necessary to support efficient development and delivery of business capabilities
4.6	[S4VS111]	Use the acceptance criteria to determine whether the functionality has been properly implemented
4.7	[S4VS112]	Split capabilities into features by applying following patterns-Work flow steps; Business rule variations; Major effort; Simple/complex; Variations in data; Data methods; Deferring system qualities; Operations; Use-case scenarios; Breaking out a spike
5	[S4VS113]	SP: Perform Solution Demo

5.1	[S4VS114]	Solution Demo as a Pull event
5.1.1	[S4VS115]	Pulls various aspects of the solution together to ensure that the Agile Release Trains and Suppliers are creating an integrated and tested solution that is fit for its intended purpose
5.1.2	[S4VS116]	Accelerates the integration, testing, and evaluation of the solution under development–something that is otherwise all too easy to defer until too late in the solution life cycle
5.1.3	[S4VS117]	Review the progress in the broader portfolio context and make decisions about the continuation or cancellation of initiatives or changes to the budgetary investment in the value streams
5.2	[S4VS118]	Overview of Solution Demo
5.2.1	[S4VS119]	Identify most recent System Demos from Pre- and Post-PI Planning that inform those who are staging the demo as to what specific capabilities and other aspects of the solution can be demonstrated
5.2.2	[S4VS120]	Includes following stakeholders to solution demo-Solution Management, Value Stream Engineer, Architects/Engineering, Customers, ART representatives, Program Portfolio Management representatives, Value Stream Level stakeholders, executive sponsors, senior management, and DevOps
5.2.3	[S4VS121]	Review the value stream PI Objectives that were agreed to at the beginning of the PI
5.2.4	[S4VS122]	Demonstrate each objective and capability in an end-to-end use case
5.2.5	[S4VS123]	Identify business value completed per objective
5.2.6	[S4VS124]	Open the forum for questions and comments
5.2.7	[S4VS125]	Wrap up by summarizing progress, feedback, and action items
5.2.8	[S4VS126]	Timebox the demo to one to two hours
5.2.9	[S4VS127]	Share demo responsibilities among lead engineers and team members who have new capabilities to demonstrate
5.2.10	[S4VS128]	Minimize PowerPoint slides; demonstrate only working, tested capabilities
5.2.11	[S4VS129]	Discuss the impact of the current PI on the solution NFRs and Solution Intent
5.2.12	[S4VS130]	Demonstrate in the solution context

5.3	[S4VS131]	Strategy, Investment, and Timing of Solution Demos
5.3.1	[S4VS132]	Invest in integration, testing, and supporting infrastructure to integrate big systems
5.3.2	[S4VS133]	Leverage virtualization, environment emulation, mocks, stubs, reduced test suites, etc to assist integration and testing
5.3.3	[S4VS134]	Allocate time and effort for integration and demonstrations during PI planning
6	[S4VS135]	SP: Develop Pre- and Post-PI Planning
6.1	[S4VS136]	Understand the Pre- and Post-PI Planning Meetings
6.1.1	[S4VS137]	Allow Agile Release Trains and Suppliers in large Value Streams to build an aligned plan for the next Program Increment (PI)
6.1.2	[S4VS138]	Use pre-PI planning event to coordinate input objectives, key Milestones, and Solution Context and business context for the ART planning sessions
6.1.3	[S4VS139]	Use Post-PI planning event to integrate the results of planning into the Vision and Roadmap for the value stream
6.1.4	[S4VS140]	Agree on a set of value stream PI Objectives to be implement by the end of the PI
6.1.5	[S4VS141]	Understand the Pre- and Post-PI Planning to provide for high-bandwidth communication through face-to-face alignment
6.1.6	[S4VS142]	Align ARTs to value streams via the ART and value stream PI objectives
6.1.7	[S4VS143]	Identify dependencies and foster cross-ART coordination
6.1.8	[S4VS144]	Provide the opportunity for just the right amount of Solution-level architectural (and,Provide the opportunity
6.1.9	[S4VS145]	Match solution demand to ART capacities
6.1.10	[S4VS146]	Build value stream-wide team to create the social fabric necessary to achieve high performance
6.2	[S4VS147]	Gain Context in the Solution Demo
6.2.1	[S4VS148]	Provides critical context for the pre- and post-PI planning meetings
6.3	[S4VS149]	Prepare for Pre- and Post-PI Planning
6.3.1	[S4VS150]	Bring together stakeholders from all parts of the value stream

6.3.2	[S4VS151]	Defines (The executive) current business, solution, and Customer context
6.3.3	[S4VS152]	Briefings prepared by Solution Management, including the top capabilities in the value stream backlog
6.3.4	[S4VS153]	Clear definitions of the upcoming milestones
6.4	[S4VS154]	Set Planning Context in Pre-PI Planning
6.4.1	[S4VS155]	Perform pre-PI planning meeting to build the context that allows the ARTs and Suppliers to create their plans
6.4.2	[S4VS156]	Perform following sessions-PI Summary Reports, Business and Solution Context, Value Stream Backlog, and Next PI Features
6.5	[S4VS157]	Summarize Results in Post-PI Planning
6.5.1	[S4VS158]	Perform Post-PI Planning to synchronize the ARTs and create the overall solution plan and roadmap
6.5.2	[S4VS159]	Include following agenda during Post-PI planning-PI Planning Report, Plan Review, Risk Analysis, and Confidence Vote, Plan Rework if Necessary, Planning Retrospective and Moving Forward
6.6	[S4VS160]	Create the Right Outcomes
6.6.1	[S4VS161]	Create a set of SMART objectives for the value stream
6.6.2	[S4VS162]	Identify stretch objectives to provide the flexible capacity and scope management options needed to increase reliability and quality of PI execution
6.6.3	[S4VS163]	Create a value stream planning board, which highlights the objectives, anticipated delivery dates, and any other relevant milestones, aggregated from the program boards
6.6.4	[S4VS164]	Develop a vote of confidence/commitment from the entire group to these objectives
7	[S4VS165]	SP: Develop Solution Context
7.1	[S4VS166]	Solution Context Drives the Solution Intent
7.1.1	[S4VS167]	Include solution context surface as part of the Definition of Done for a solution Increment
7.2	[S4VS169]	Fixed vs. Evolving Solution Contexts
7.2.1	[S4VS170]	Impose all solution context requirements on the solution via solution intent
7.2.2	[S4VS171]	Require evolution of the Customer's deployment environment for new solutions
7.2.3	[S4VS172]	Track changes as both the system and deployment environment have to evolve to a common state

7.2.4	[S4VS173]	Manage risk by adopting multiple potentially viable solution contexts
7.3	[S4VS174]	Types of Solution Contexts
7.3.1	[S4VS175]	Solution Context for a System of Systems
7.3.1.1	[S4VS176]	Specifies how the solution is packaged, deployed, and integrated
7.3.2	[S4VS177]	Solution Context for IT Deployment Environments
7.3.2.1	[S4VS178]	Deployment must consider specific interfaces, deployed OSs, firewalls, APIs to other applications, hosted or cloud infrastructure
7.3.3	[S4VS179]	Solution Context Includes Portfolio-Level Concerns
7.3.3.1	[S4VS180]	Work together to accomplish the system builder's larger objective
7.3.3.2	[S4VS181]	Collaborate and integrate with the others to provide the operational value stream with a seamless, end-to-end solution
7.4	[S4VS182]	Continuous Collaboration Ensures Deployability
7.4.1	[S4VS183]	Continuous collaboration ensures deployability through continuous feedbacks
7.4.2	[S4VS184]	Integrates the entire system-of-systems value stream to demonstrate progress toward the top-level context's Milestone and Release commitments
7.4.3	[S4VS185]	Continuous collaboration helps ensure that the solution can be deployed in the ultimate Customer's context
7.4.4	[S4VS186]	Effective Customer/system builder collaboration helps ensure that the system meets the Customers' needs
8	[S4VS187]	SP: Develop Solution
8.1	[S4VS188]	Overview of Solution Development in SAFe
8.1.1	[S4VS189]	ARTs operate synchronously and build the solution in increments, which are fully integrated and evaluable via the Solution Demo
8.1.2	[S4VS190]	Solution intent captures the goal of the solution and allows for exploring and defining fixed and variable requirements and designs, which are in part derived from the Solution Context
8.1.3	[S4VS191]	Clarify the intent, validate assumptions, and review progress
8.1.4	[S4VS192]	Drive development, make scope and priority decisions, and manage the flow of Features and Capabilities and Nonfunctional Requirements

8.1.5	[S4VS193]	Requires a holistic approach to definition, planning, implementation, and review of the solution to develop an economically viable solution
8.2	[S4VS194]	Effective Solution Development Requires Systems Thinking
8.2.1	[S4VS195]	Solution Capabilities, Enablers, and NFRs, and NFRs
8.2.1.1	[S4VS196]	Capabilities are the end-to-end solution services that support the achievement of user goals
8.2.1.2	[S4VS197]	Implement capabilities via vertical, end-to-end slices of value, which enable incremental solution development
8.2.1.3	[S4VS198]	Enablers provide for exploration of new capabilities, contribute to solution infrastructure and architecture, and enhance NFRs. That drives early value delivery and architectural robustness
8.2.2	[S4VS199]	Solution Intent
8.2.2.1	[S4VS200]	Drives and captures a holistic view of the solution and includes different aspects that govern value definition, including structural, behavioral, functional, and other views
8.2.2.2	[S4VS201]	Adopt Model-Based Systems Engineering that provides an effective way of reasoning about the solution and also serves as an efficient communication tool for sharing this knowledge
8.2.2.3	[S4VS202]	Use SAFe's fixed-variable solution intent paradigm enables value streams to enhance solution intent based on the objective knowledge that emerges over the course of many learning cycles
8.2.3	[S4VS203]	Customer and Solution Context
8.2.3.1	[S4VS204]	Take a systems view to ensure that the solution builder understands the solution context
8.2.3.2	[S4VS205]	Provides the additional pieces that determine operational requirements and constraints
8.2.4	[S4VS206]	Solution Integration, Testing, and Demo
8.2.4.1	[S4VS207]	Frequently evaluate integrated increments of the entire solution for effective solution development
8.2.4.2	[S4VS208]	Perform continuous Integration and testing while Solution Demonstration occurs on a fixed PI cadence
8.2.5	[S4VS209]	Building an Economically Viable Solution
8.2.5.1	[S4VS210]	Building a complex solution requires informed, effective decision-making

8.2.5.2	[S4VS211]	Perform continuous exploration that includes learning Milestones, Customer feedback loops, and Set-Based Design informs and streamlines the learning process by validating good options and eliminating less viable ones

C.5 Portfolio Level

Num	ID	Description
		Title: SAFe Portfolio Level Practices Author: Abdur Razzak and Dr. Sarah Beecham Specific Goal: Define Portfolio Level
1	[S4PO1]	SP: Define Strategic Themes
1.1	[S4PO2]	Formulate Strategic themes
1.1.1	[S4PO3]	Define strategic themes in the individual SAFe portfolio context
1.2	[S4PO4]	Influence portfolio vision
1.2.1	[S4PO5]	Economic Framework
1.2.1.1	[S4PO6]	Be aware that strategic themes could affect any of the major parameters including development/cycle time, product cost, product value, development expense, and risk
1.2.2	[S4PO7]	Value streams budgets
1.2.2.1	[S4PO8]	Provides the spending and personnel allocations necessary to accomplish the portfolio vision
1.2.3	[S4PO9]	Portfolio backlog
1.2.3.1	[S4PO10]	Provides decision-making filters in the portfolio kanban, thereby influence the portfolio backlog
1.2.3.2	[S4PO11]	Impacts the identification, success criteria, and prioritization of epics in the funnel and backlog states
1.2.3.3	[S4PO12]	Warrants consideration and discussion in the lightweight business case in the analysis state
1.2.3.4	[S4PO13]	Impact on how epics are split and implemented in the implementation state
1.2.4	[S4PO14]	Vision and Priorities
1.2.4.1	[S4PO15]	Apply strategic themes through Product and Solution management to influence the vision and roadmap
1.2.4.2	[S4PO16]	Apply WSJF to prioritize the items in the Value stream and Program backlogs
1.2.4.3	[S4PO17]	Provides an important means of conceptual alignment between the trains
1.3	[S4PO18]	Measure Progress
1.3.1	[S4PO19]	Success criteria provide a mechanism to understand progress towards the intent
1.3.2	[S4PO20]	Innovation accounting produce the desired long term results by implementing the tool, functionality, testing or other mechanisms to collect the data

1.3.3	[S4PO21]	Success criteria provide learning milestones that allow the portfolio to understand the solution involved, validate business and technical hypotheses, and pivot towards a better solution
1.3.4	[S4PO22]	PI cadence offers time box experimenting approach which later help to produce the desired and long term results
2	[S4PO23]	SP: Define Program Portfolio Management
2.1	[S4PO24]	Responsibilities of Program Portfolio Management
2.1.1	[S4PO25]	Define strategic themes that connect SAFe portfolio to evolving enterprise business strategy
2.1.2	[S4PO26]	Define and prioritize crosscutting portfolio backlog epics, and report to the business on investment spend and progress via KPIs and other aspects of portfolio context
2.2	[S4PO27]	Lean-Agile Program Portfolio management
2.2.1	[S4PO28]	Adopt SAFe's set of seven transformational patterns that can be used to move the organization to Lean-Agile Program Portfolio Management ([Decentralized decision-making], [Demand management; continuous value flow], [Lightweight; epic-only business cases], [Decentralized, rolling-wave planning], [Agile estimating and planning], [Lean-Agile budgeting and self-managing Agile Release Trains], [Objective, fact-based measure and milestones])
2.2.2	[S4PO29]	Helps to better understand how to fulfill the primary responsibilities–strategy and investment funding, program management, and governance–but in a more effective Lean-Agile fashion
2.3	[S4PO30]	Strategy and Investment funding
2.3.1	[S4PO31]	Lean-Agile budgeting
2.3.1.1	[S4PO32]	Allocate budget authority to the decision makers
2.3.1.2	[S4PO33]	Avoid overhead and enables the train to make fast and local decisions within the constraints of the allocated budget
2.3.2	[S4PO34]	Demand management and continuous value flow
2.3.2.1	[S4PO35]	Manage WIP and insure efficient product development flow by bringing visibility of existing program work and understating Agile program velocities
2.3.3	[S4PO36]	Epics and lightweight business cases

2.3.3.1	[S4PO37]	Define and analyze epics to provide visibility and economic justification for upcoming crosscutting work
2.3.3.2	[S4PO38]	Define lightweight business case
2.4	[S4PO39]	Program management
2.4.1	[S4PO40]	Self-managing Agile Release Trains
2.4.1.1	[S4PO41]	Adopt value-stream based, self-managing Agile Release Trains to provide a continuous of flow to its stakeholders and self-organizing Agile Release Trains
2.4.2	[S4PO42]	Decentralized, Rolling-Wave Planning
2.4.2.1	[S4PO43]	Adopt decentralized, program, and team-based rolling-wave planning via routine and cadence-based PI Planning activity
2.4.3	[S4PO44]	Agile Estimating and Planning
2.4.3.1	[S4PO45]	Adopt Agile estimating and planning by using the currency of story points
2.5	[S4PO46]	Governance
2.5.1	[S4PO47]	Portfolio context
2.5.1.1	[S4PO48]	Measure KPI's by including quantitative and financial measures, such as ROI, market share, customer net promoter score, innovation accounting, and more
2.5.1.2	[S4PO49]	Measure additional metrics
2.5.1.2.1	[S4PO50]	Lean Portfolio Metrics
2.5.1.2.1.1	[S4PO51]	Use Lean Portfolio Metrics to assess internal and external progress for an entire Portfolio.
2.5.1.2.1.2	[S4PO52]	Provides the leanest set that a few Lean-Agile Enterprises are using effectively to evaluate the overall performance of their transformations.
2.5.1.2.2	[S4PO53]	Portfolio Kanban board
2.5.1.2.2.1	[S4PO54]	Ensure that Epics and Enablers are reasoned and analyzed prior to reaching a Program Increment boundary, are prioritized appropriately, and have established acceptance criteria to guide a high-fidelity implementation
2.5.1.2.2.2	[S4PO55]	Track epics and enablers to understand which ones are being worked on and which have been completed
2.5.1.2.2.3	[S4PO56]	Review and Analysis states are WIP limited and reflect the limit set by the PPM Team
2.5.1.2.3	[S4PO57]	Epic Burn-up Chart
2.5.1.2.4	[S4PO59]	Epic Progress Measure
2.5.1.2.4.1	[S4PO60]	Represents the name of the epic
2.5.1.2.4.2	[S4PO61]	Represents the total current estimated story points for an epic's child features/stories

2.5.1.2.4.3	[S4PO62]	Represents the initial epic estimate, in story points, from the lightweight business case
2.5.1.2.4.4	[S4PO63]	Represents the current estimated story points, rolled up from the epic's child features/stories and the initial epic estimate
2.5.1.2.5	[S4PO64]	Epic Success Criteria
2.5.1.2.5.1	[S4PO65]	Use to establish scope and drive more detailed feature elaboration
2.5.1.2.6	[S4PO66]	Enterprise Balanced Scorecard
2.5.1.2.6.1	[S4PO67]	Provides four perspectives (Efficiency, Value Delivery, Quality, and Agility) to measure performance for each portfolio
2.5.1.2.7	[S4PO68]	Program Portfolio Management Self-Assessment
2.5.1.2.7.1	[S4PO69]	Program Portfolio Management (PPM) team continuously assesses and improves their processes using a structured, periodic Self-Assessment
2.5.1.3	[S4PO70]	Analyze qualitative data by including Strengths, Weaknesses, Opportunities and Threats (SWOT) analysis, and most importantly, the accumulated solution, market and business knowledge of the portfolio stakeholders.
2.5.2	[S4PO71]	Life-Cycle Governance
2.5.2.1	[S4PO72]	Encourage and facilitate incremental development and fast customer feedback
2.5.2.2	[S4PO73]	Facilitate continuous improvement by quantitative metrics, customer feedback, and the Inspect and Adapt retrospective cycle
3	[S4PO74]	SP: Define Portfolio Kanban
3.1	[S4PO75]	Make the strategic business initiative fully visible
3.2	[S4PO76]	Bring structure and visibility to the analysis and decision-making that moves these initiatives into implementation
3.3	[S4PO77]	Provide WIP limits to ensure that the teams responsible for analysis undertake into responsibly and do not create expectation for implementation or time frames that far exceed capacity and reality
3.4	[S4PO78]	Help to derive collaboration among the key stakeholders in the business
3.5	[S4PO79]	Provide a quantitative, transparent basis for economic decision-making for these, the most important business decisions

3.6	[S4PO80]	Develop a Kanban system to capture, analyze, approve, and track epics
4	[S4PO81]	SP: Define Portfolio Backlog
4.1	[S4PO82]	Portfolio Backlog Input
4.1.1	[S4PO83]	Determine feasibility and potential ROI of all epics in portfolio Kanban in order to reach the portfolio backlog
4.1.2	[S4PO84]	Identify, elaborate, estimate, and analyze all epics that reached in mature state to achieve a go recommendation from PPM
4.2	[S4PO85]	Managing the Portfolio Backlog
4.2.1	[S4PO86]	Apply WSJF prioritization to compare business and enabler epics against each other
4.3	[S4PO87]	Forecast
4.3.1	[S4PO88]	Use capacity allocation to estimate portfolio epic based on the given knowledge of program velocities
4.4	[S4PO89]	Move to implementation
4.4.1	[S4PO90]	Move prioritized epic from portfolio backlog to the implementation stage of portfolio kanabn
5	[S4PO91]	SP: Allocate Budgets
5.1	[S4PO92]	Lean-Agile Budgeting: Beyond Project Cost Accounting
5.1.1	[S4PO93]	Fund Value Streams, Not Projects
5.1.2	[S4PO94]	Empower Value Stream Content Authority
5.1.3	[S4PO95]	Provide object Evidence of Fitness for Purpose
5.1.4	[S4PO96]	Approve Epic Level Initiatives
5.1.5	[S4PO97]	Exercise Fiscal Governance with Dynamic Budgeting
6	[S4PO98]	SP: Define CapEx and OpEx
6.1	[S4PO99]	Operating expense (OpEx) records ongoing costs of running a product, business, or service
6.2	[S4PO100]	Capital Expense (CapEx) reflects the monies that required to purchase, upgrade, or fix tangible physical assets
7	[S4PO101]	SP: Define Portfolio Epics
7.1	[S4PO102]	Analyze epics before committed to implementation
7.2	[S4PO103]	Define Success criteria to validate the implementation
7.3	[S4PO104]	Develop Incremental implementation by keeping the epics in the portfolio backlog until there is implementation capacity available
7.4	[S4PO105]	Split epics into smaller pieces the represent incremental value
7.5	[S4PO106]	Approve the investment in Epics
8	[S4PO107]	SP: Define Value Streams

8.1	[S4PO108]	Identify value stream
8.1.1	[S4PO109]	Understand the actual flow of value by using set of questions in the specific business context
8.1.2	[S4PO110]	Characterize each value stream for more complete understanding
8.1.3	[S4PO111]	Understand how to form Agile Release Train to realize them when development of value streams in cross boundaries
8.1.4	[S4PO112]	Understand where in the organization that value is created because that is where the people, processes, and systems are
8.1.5	[S4PO113]	Focus on functional and local improvement because the result of which may be optimization of one function or step but sub-optimization of the end-to-end flow
8.2	[S4PO114]	Allocate budget for development of value stream
8.2.1	[S4PO115]	Identify and understand value streams flow through the organization
8.2.2	[S4PO116]	Manage budget for each train in accordance with Lean-Agile budgeting principles
8.3	[S4PO117]	Realize development value stream with Agile Release Train
8.3.1	[S4PO118]	Consider Multiple Value Stream ART in the case of small company where the employees are fewer than 50 people
8.3.2	[S4PO119]	Consider Single Value Stream ARTs in case of medium-size company where the staffs are fewer than 200 people
8.3.3	[S4PO120]	Consider Multiple-ART Value Stream in case of large-organization where more than 900 staffs are in development and deployment as well as hundreds more in operations
8.4	[S4PO121]	Coordinate Agile Release Train within value streams
8.4.1	[S4PO122]	Manage dependencies by applying extensive degree of cooperation; a common value stream backlog; implementation of new, crosscutting capabilities; additional system integration; additional roles and responsibilities; special considerations for pre-, post-, and PI planning activities; different degree and types of DevOps support
8.5	[S4PO123]	Coordinate value streams with a Portfolio
8.5.1	[S4PO124]	Implement value stream coordination to ensure that the enterprise moves forward with each value stream in lockstep with the enterprise objectives

8.6 8.6.1	[S4PO125] [S4PO126]	Reduce Time-to-Market with value stream mapping Apply ride management ART to reduce delays in the value stream

C.6 Roles and Responsibility

Num	ID	Description
		Title: Roles and Responsibilities in Different level of SAFe Author: Abdur Razzak and Dr. Sarah Beecham
1	[S4RR1]	Define Role and Responsibilities–Team Level
1.1	[S4RR2]	Define Product Owner role
1.1.1	[S4RR3]	Preparation and participation in PI planning
1.1.1.1	[S4RR4]	Participate in program refinement backlog and PI planning
1.1.1.2	[S4RR5]	Update and review team backlog and contribute to the vision, roadmap, and content presentations
1.1.1.3	[S4RR6]	Provides necessary clarifications to assist team with their story estimation and story sequencing for the upcoming program increment
1.1.2	[S4RR7]	Develop Iteration execution
1.1.2.1	[S4RR8]	Perform backlog refinement
1.1.2.2	[S4RR9]	Perform iteration planning
1.1.2.3	[S4RR10]	Elaborate Just-in-Time Story
1.1.2.4	[S4RR11]	Support acceptance test-driven development (ATDD)
1.1.2.5	[S4RR12]	Accept stories
1.1.2.6	[S4RR13]	Understand enabler work
1.1.2.7	[S4RR14]	PO Participate in team demo and retrospective
1.1.3	[S4RR15]	Develop Program execution
1.1.3.1	[S4RR16]	Coordinate content dependencies with other product owners by attending weekly PO sync meetings
1.1.3.2	[S4RR17]	Produce system demo for program and value stream stakeholders
1.1.4	[S4RR18]	Organize Inspect and Adapt workshop
1.1.4.1	[S4RR19]	Define and implement improvement stories that will increase the velocity and quality of the program
1.1.4.2	[S4RR20]	Produce the PI system demo for program stakeholders
1.1.4.3	[S4RR21]	Participate in the preparation of the PI system demo to make sure that they will be able to show the most critical aspects of the solution to the stakeholders
1.2	[S4RR22]	Define Scrum Master role
1.2.1	[S4RR23]	Responsibilities on the Team
1.2.1.1	[S4RR24]	Exhibits Lean-Agile leadership
1.2.1.2	[S4RR25]	Supports the rules

1.2.1.3	[S4RR26]	SM facilitates the team's progress toward the goal
1.2.1.4	[S4RR27]	Lead the team's efforts in relentless improvement
1.2.1.5	[S4RR28]	SM facilitates meetings
1.2.1.6	[S4RR29]	SM supports the product owner
1.2.1.7	[S4RR30]	SM eliminates impediments
1.2.1.8	[S4RR31]	Promotes SAFe quality practices
1.2.1.9	[S4RR32]	SM builds a high-performing team
1.2.1.10	[S4RR33]	Protect and Communicates
1.2.2	[S4RR34]	Responsibilities on the Train
1.2.2.1	[S4RR35]	SM coordinate with other teams
1.2.2.2	[S4RR36]	Facilitates preparation and readiness for ART ceremonies
1.2.2.3	[S4RR37]	Support estimation
2	[S4RR38]	Define Role and Responsibilities–Program Level
2.1	[S4RR39]	Define System Team responsibilities
2.1.1	[S4RR40]	Build development infrastructure
2.1.1.1	[S4RR41]	Create and maintain infrastructure, including continuous integration, automated builds, and automated build verification testing
2.1.1.2	[S4RR42]	Create platforms and environments for solution demonstration, QA, user testing, etc.
2.1.1.3	[S4RR43]	Create products, utilities, and scripts to automate deployment
2.1.1.4	[S4RR44]	Facilitate the technical aspects of collaboration with third parties, such as data or service providers, hosting facilities, etc.
2.1.2	[S4RR45]	Perform system integration
2.1.2.1	[S4RR46]	Participate in PI planning and the pre- and post- PI planning meetings at the value stream level, and in backlog refinement to define integration and test capabilities and features
2.1.2.2	[S4RR47]	Determine and help to maintain decisions and policies for appropriate branching models
2.1.2.3	[S4RR48]	Run solution-level integration scripts or integrate manually where automation is not possible or has not yet been applied
2.1.2.4	[S4RR49]	Assist component teams in defining inter-component interfaces
2.1.2.5	[S4RR50]	System Team attend other teams' stand-ups in support of daily activities
2.1.3	[S4RR51]	Perform end-to-end and solution performance testing

2.1.3.1	[S4RR52]	Create new automated test scenarios
2.1.3.2	[S4RR53]	Extend test scenarios to larger data sets
2.1.3.3	[S4RR54]	Organize test cases designed by individual teams into ordered suits
2.1.3.4	[S4RR55]	Perform manual testing and run automated tests for new features and stories
2.1.3.6	[S4RR57]	System Team assist teams in creating reduces test suites that they themselves can run
2.1.3.7	[S4RR58]	Perform test solution against NFRs and assist system and solution engineering to identify system shortfalls and bottlenecks
2.1.4	[S4RR59]	Perform system and solution demos
2.1.4.1	[S4RR60]	Ensure that the demo environments are adequate to the challenge of reliably demonstrating new solution functionality
2.1.5	[S4RR61]	Develop release solution
2.2	[S4RR63]	Define Release Management responsibilities
2.2.1	[S4RR64]	Ensure that the organization's release governance is understood
2.2.2	[S4RR65]	Communicate release status to external stakeholders
2.2.3	[S4RR66]	Ensure that an appropriate deployment/distribution plan is in place
2.2.4	[S4RR67]	Coordinate with marketing and with product and solution management on internal and external communications
2.2.5	[S4RR68]	Validate that the solution meets relevant quality and governance criteria
2.2.6	[S4RR69]	Participate in Inspect and Adapt to improve the release process, value stream productivity, and solution quality
2.2.7	[S4RR70]	Provide final authorization for the release
2.2.8	[S4RR71]	Act as a liaison with program portfolio management, as appropriate
2.3	[S4RR72]	Shared Services responsibilities
2.3.1	[S4RR73]	Participate in PI planning as well as Pre- and Post- PI planning
2.3.2	[S4RR74]	Collaborate to fulfill dependencies that occur during Program Increment (PI) execution
2.3.3	[S4RR75]	Participate in system demos, solution demos, and Inspect and Adapt workshops when appropriate
2.3.4	[S4RR76]	Conduct specialized training to keep up with advancements in their respective fields.

2.3.5	[S4RR77]	Involve with Agile team for short period of time
2.4	[S4RR78]	User Experience Designer (UX)
2.4.1	[S4RR79]	Work with stakeholders to understand the specific business targets behind the user-system interaction
2.4.2	[S4RR80]	Provide Agile teams with the next increment of UI design, User experience guidelines, and design elements in a just-in-time (but timely enough) fashion
2.4.3	[S4RR81]	Continuously validate user experience via user experience testing
2.4.4	[S4RR82]	Work with system and solution architect/engineering and teams to build and maintain the technical foundation for real-time user experience validation, feedback, tracking statistics, etc.
2.4.5	[S4RR83]	Share user experience guidelines across the program; educate developers on the best practices of maintaining good UI design
2.4.6	[S4RR84]	Assist test engineers and the System team in user experience testing and test automation
2.4.7	[S4RR85]	Lead UI design and User experience/UI community of practice workshop
2.4.8	[S4RR86]	Attend iteration planning, backlog refinement meetings, system demos, and solution demos whenever critical UI-related work is involved
2.5	[S4RR87]	System and Solution Architect /Engineer responsibilities
2.5.1	[S4RR88]	System and Solution Architect /Engineer participates in planning, definition, and high-level design of the solution and explore solution alternatives
2.5.2	[S4RR89]	System and Solution Architect /Engineer role involves four practices:
2.5.2.1	[S4RR89.1]	System and Solution Architect /Engineer defines subsystems and their interfaces;
2.5.2.2	[S4RR89.2]	System and Solution Architect /Engineer allocates responsibilities to subsystems;
2.5.2.4	[S4RR89.4]	System and Solution Architect /Engineer communicates requirements for interactions with solution context
2.5.3	[S4RR90]	Work with customers, stakeholders, and suppliers to establish high-level solution intent; help establish the solution intent information models and documentation requirements

2.5.4	[S4RR91]	Establish critical Nonfunctional requirements t the solution level; participate in the definition of others
2.5.5	[S4RR92]	Operate within the economic framework to validate the economic impact of design decisions
2.5.6	[S4RR93]	Work with portfolio stakeholders, particularly the enterprise architect, to develop, analyze, split, and realize the implementation of enabler epics
2.5.7	[S4RR94]	Participate in PI planning and pre- and post- PI planning, system and solution demos, and Inspect and Adapt events
2.5.8	[S4RR95]	System and Solution Architect /Engineer defines, explores, and support the implementation of value stream and program enablers to evolve solution intent; work directly with Agile Teams to implement, explore, or support them
2.5.9	[S4RR96]	Plan and develop the architectural runways in support of upcoming business features/capabilities
2.5.10	[S4RR97]	System and Solution Architect /Engineer works with product and solution management to determine capacity allocation for enablement work
2.5.11	[S4RR98]	Support technology/engineering aspects of program and value stream kanbans
2.5.12	[S4RR99]	Supervise and foster Built-in Quality
2.6	[S4RR100]	Define Release train engineer responsibilities
2.6.1	[S4RR101]	Manage and optimize the flow of value through the program using various tools, such as the Program and Value stream Kanbans and information radiators
2.6.2	[S4RR102]	Establish and communicate the annual calendars for iterations and Program increments
2.6.3	[S4RR103]	Facilitate PI planning readiness via fostering the preparation of vision and backlog, and via pre- and post-PI planning meetings
2.6.4	[S4RR104]	Facilitate PI planning
2.6.5	[S4RR105]	Aggregate Team PI Objectives into program PI Objectives (the RTE) and publish them for visibility and transparency
2.6.6	[S4RR106]	Aggregate Program PI Objectives into Value Stream PI Objectives (the VSE) and publish them for visibility and transparency
2.6.7	[S4RR107]	Assist with execution and Feature/Capability completion tracking

2.6.8	[S4RR108]	RTE facilitate periodic synchronization meetings, including the ART sync at the Program level and the Value Stream (VS) sync at the value stream level
2.6.9	[S4RR109]	Assist with economic decision-making by facilitating feature and capability estimation by teams and roll-up to the value stream level and Portfolio level
2.6.10	[S4RR110]	RTE escalates and tracks impediments
2.6.11	[S4RR111]	Encourage the collaboration between teams and System and Solution Architects, Engineering, and User Experience designers
2.6.12	[S4RR112]	Work with Product Management, Product Owners, and other value stream stakeholders to ensure strategy and execution alignment
2.6.13	[S4RR113]	RTE helps to manage risks and dependencies
2.6.14	[S4RR114]	Report status to Program portfolio management and Release management to support related activities
2.6.15	[S4RR115]	Understand and operate with the ART budget
2.6.16	[S4RR116]	RTE provides input on resourcing to address critical bottlenecks
2.6.17	[S4RR117]	Attend on system demos and solution demos
2.6.18	[S4RR118]	Drive continuous improvement via Inspect and Adapt workshops; assess the agility level of the program/value stream and help improve
2.6.19	[S4RR119]	Encourage team level, program level, and value stream level continuous integration and community of practices around SAFe, Agile and Lean. And, around engineering and quality practices
2.6.20	[S4RR120]	Coach leaders, teams, and Scrum masters in lean-Agile practices and mindsets
2.7	[S4RR121]	Define Product Management and Solution Management responsibilities
2.7.1	[S4RR122]	Understand customer needs and validate solutions
2.7.2	[S4RR123]	Understand and support portfolio work
2.7.3	[S4RR124]	Product Manager develops and communicates the program vision and roadmap
2.7.4	[S4RR125]	Manage and prioritize the flow of work
2.7.5	[S4RR126]	Participate in PI planning
2.7.6	[S4RR127]	Define releases and program increments
2.7.7	[S4RR128]	Work with system architect/engineering to understand enabler work
2.7.8	[S4RR129]	Participate in demos and Inspect and Adapt

2.7.9	[S4RR130]	Build an effective Product Manager/Product Owner team
2.7.10	[S4RR131]	Collaborate with solution management [Additional responsibility]
2.7.11	[S4RR132]	Participate in Pre- and Post-PI planning [Additional responsibility]
2.7.12	[S4RR133]	Participate in the Solution demo [Additional responsibility]
2.7.13	[S4RR134]	Collaborate with release management [Additional responsibility]
2.8	[S4RR135]	Define Business Owners responsibilities
2.8.1	[S4RR136]	Prior to PI planning
2.8.1.1	[S4RR137]	Participate in Pre-PI planning as applicable
2.8.1.2	[S4RR138]	Business Owner understands and assures that business objectives are understood and agreed to by key stakeholders of the train, including the Release Train Engineer (RTE), Product Management, and System Architects
2.8.1.3	[S4RR139]	Business Owner prepares to communicate the business context, including milestones and significant external dependencies, such as those of suppliers
2.8.2	[S4RR140]	During PI planning
2.8.2.1	[S4RR141]	Deliver relevant elements of the business context in the defined PI planning agenda time-box
2.8.2.2	[S4RR142]	Business Owner participate in certain key ceremonies, including presentation of vision, draft plan review, assigning final business value to PI Objectives, and approving final plans
2.8.2.3	[S4RR143]	Draft plan review to understand the bigger picture and how these plans taken together to fulfill the current business objectives.
2.8.2.4	[S4RR144]	Trace significant external commitments and dependencies
2.8.2.5	[S4RR145]	Actively circulate during planning, communicate business priorities to the teams, and maintain agreement and alignment among the stakeholders as to the key objectives of the train
2.8.2.6	[S4RR146]	Participate in the management review and problem-solving meeting.
2.8.2.7	[S4RR147]	Review and adjust scope, compromise as necessary
2.8.2.8	[S4RR148]	Assign business value

2.8.3	[S4RR149]	At Inspect and Adapt
2.8.3.1	[S4RR150]	Assess actual business value achieved vs. plan
2.8.4	[S4RR151]	During PI execution
2.8.4.1	[S4RR152]	Participate in release management, with focus on scope management, quality, deployment options, release, and market considerations
2.8.4.2	[S4RR153]	Actively participate in ongoing agreements to maintain business and development alignment as priorities and scope are inevitably changed
2.8.4.3	[S4RR154]	Attend the system demos to view progress and provide feedback
2.8.4.4	[S4RR155]	Business Owner attends occasional Agile team iteration planning and iteration retrospective meetings, as appropriate
2.8.5	[S4RR156]	Additional responsibilities
2.8.5.1	[S4RR157]	Participate and provide feedback from the solution demos relevant to the capabilities and subsystem being built by the ART
2.8.5.2	[S4RR158]	Business Owner actively address impediments, especially those that escalate beyond the authority of the key stakeholders on the train
2.8.5.3	[S4RR159]	Participate in Post-PI planning and assist the trains in adjusting the ART's PI plans as needed
2.8.5.4	[S4RR160]	Participate, in some cases, in program portfolio management, product management, and even release management and system architecture
2.9	[S4RR161]	Define Customer responsibilities
2.9.1	[S4RR162]	Participate as business owner in PI planning
2.9.2	[S4RR163]	Attend solution and possibly system demos; helps to evaluate solution increment
2.9.3	[S4RR164]	Participate in Inspect and Adapt workshops; assist in removing some systemic impediments
2.9.4	[S4RR165]	Interact with analyst and subject matter experts during specification workshops
2.9.5	[S4RR166]	Customer collaboratively manages scope, time, and other constraints with Product and Solution management
2.9.6	[S4RR167]	Define the roadmap, milestones, and releases
2.9.7	[S4RR168]	Communicate the economic logic behind the solution and help to validate assumptions in the economic framework

2.9.8	[S4RR169]	Review technical and financial status of the solution
2.9.9	[S4RR170]	Participate in beta testing, UAT, other forms of solution validation
3	[S4RR171]	Define Role and Responsibilities –Portfolio Level
3.1	[S4RR172]	Define Epic Owner responsibilities
3.1.1	[S4RR173]	Prior to Approval: Preparing the Epic
3.1.1.1	[S4RR174]	Work with stakeholders and subject matter experts to define the epic and its potential benefits
3.1.1.2	[S4RR175]	Epic Owner establish cost of delay and identify business sponsors
3.1.1.3	[S4RR176]	Epic Owner works with development team to size the epic and provide input for economic prioritization based on WSJF
3.1.1.4	[S4RR177]	Define epic success criteria
3.1.1.5	[S4RR178]	Shepherd the epics through the portfolio Kanban system and create the lightweight business case
3.1.1.6	[S4RR179]	Prepare to present the business case to program portfolio management for a go/no-go decision
3.1.2	[S4RR180]	Presenting the Epic
3.1.2.1	[S4RR181]	Present the epic to Program Portfolio Management
3.1.3	[S4RR182]	After Approval: Implementation
3.1.3.1	[S4RR183]	Work with product management and solution management to split the epic into value stream and/or program epics and features. Then prioritize them in the backlogs
3.1.3.2	[S4RR184]	Provide guidance to the release train on the epic context of the target features
3.1.3.3	[S4RR185]	Participate in PI Planning, System Demo, and Solution Demo whenever there is critical activity related to the epic
3.1.3.4	[S4RR186]	Work with Agile Team that perform research spikes, create proof of concepts, mock-ups, etc.
3.1.3.5	[S4RR187]	Coordinate and synchronize epic-related activities with functions in sales, marketing, and other business units
3.1.3.6	[S4RR188]	Epic Owner understands and reports on progress of the epic with key stakeholders
3.2	[S4RR189]	Define Enterprise Architect responsibilities
3.2.1	[S4RR190]	Maintain a high-level, holistic vision of enterprise and development initiatives
3.2.2	[S4RR191]	Define key technical initiatives that support Budgets via Enabler Epics

3.2.3	[S4RR192]	Participate in the strategy for building and maintaining the enterprise architectural runway
3.2.4	[S4RR193]	Understand and communicate Strategic Themes and other key business drivers for architecture to system architects and nontechnical stakeholders
3.2.5	[S4RR194]	Drive architectural initiatives in the Portfolio Kanban system and participate in Epic analysis where applicable
3.2.6	[S4RR195]	Influence common modeling, design, and coding practices
3.2.7	[S4RR196]	Collect, generate, and analyze innovative ideas and technologies that are applicable across the enterprise
3.2.8	[S4RR197]	Facilitate the reuse of ideas, components, and proven patterns
3.2.9	[S4RR198]	Synchronize System and data security and quality, Production infrastructure, Solution User Experience governance, Scalability, performance, and other NFRs across solution whenever applicable
4	[S4RR199]	Define Role and Responsibilities –Value Stream Level
4.1	[S4RR200]	Define Value Stream Engineer (VSE) responsibilities
4.1.1	[S4RR201]	Manage and optimize the flow of value through the program using various tools, such as the Program and Value Stream Kanbans and information radiators
4.1.2	[S4RR202]	Establish and communicate the annual calendars for Iterations and Program Increments (PI)
4.1.3	[S4RR203]	Facilitate PI Planning readiness via fostering the preparation of Vision and Backlogs, and via Pre- and Post-PI Planning meetings
4.1.4	[S4RR204]	Facilitate PI planning
4.1.5	[S4RR205]	Aggregate program PI objectives into Value Stream PI Objectives (the VSE) and publish them for visibility and transparency
4.1.6	[S4RR206]	Assist with execution and Feature/Capability completion tracking
4.1.7	[S4RR207]	Facilitate periodic synchronization meetings, including the ART sync at the Program Level and the VS sync at the value stream level
4.1.8	[S4RR208]	Assist with economic decision-making by facilitating feature and capability estimation by teams and the roll-up to the value stream level and Portfolio Level
4.1.9	[S4RR209]	VSE escalates and tracks impediments

4.1.10	[S4RR210]	Value Stream Engineer encourages the collaboration between teams and System and Solution Architects, Engineering, and User Experience designers
4.1.11	[S4RR211]	Work with Product Management, Product Owners, and other value stream stakeholders to help ensure strategy and execution alignment
4.1.12	[S4RR212]	VSE helps to manage risks and dependencies
4.1.13	[S4RR213]	Report status to Program Portfolio Management and Release Management and support related activities
4.1.14	[S4RR214]	Understand and operate within the ART Budget
4.1.15	[S4RR215]	Provide input on resourcing to address critical bottlenecks
4.1.16	[S4RR216]	Attend System Demos and Solution Demos
4.1.17	[S4RR217]	Drive continuous improvement via Inspect and Adapt workshops; assess the agility level of the program/value stream and help improve
4.1.18	[S4RR218]	Encourage Team Level, program level, and value stream level Continuous Integration and Communities of Practice around SAFe, Agile, and Lean and around Engineering and Quality Practices
4.1.19	[S4RR219]	Coach leaders, teams, and Scrum Masters in Lean-Agile practices and mindsets
4.2	[S4RR220]	Define System and Solution Architect/Engineering responsibilities
4.2.1	[S4RR221]	Participate in planning, definition, and high-level design of the solution and explore solution alternatives
4.2.2	[S4RR222]	Define subsystems and their interfaces; allocate responsibilities to subsystems; understand solution deployment, and communicate requirements for interactions with solution context
4.2.3	[S4RR223]	Work with Customers, stakeholders, and Suppliers to establish high-level Solution Intent; help establish the solution intent information models and documentation requirements
4.2.4	[S4RR224]	Establish critical Nonfunctional Requirements at the solution level; participate in the definition of others
4.2.5	[S4RR225]	Operate within the Economic Framework to validate the economic impact of design decisions
4.2.6	[S4RR226]	Work with portfolio stakeholders, particularly the Enterprise Architect, to develop, analyze, split, and realize the implementation of enabler Epics

4.2.7	[S4RR227]	Participate in PI Planning and Pre- and Post-PI Planning, System and Solution Demos, and Inspect and Adapt events
4.2.8	[S4RR228]	Define, explore, and support the implementation of value stream and program Enablers to evolve solution intent; work directly with Agile Teams to implement, explore, or support them
4.2.9	[S4RR229]	Plan and develop the Architectural Runway in support of upcoming business Features/Capabilities
4.2.10	[S4RR230]	System and Solution Architect/Engineer works with Product and Solution Management to determine capacity allocation for enablement work
4.2.11	[S4RR231]	Support technology/engineering aspects of Program and Value Stream Kanbans
4.2.12	[S4RR232]	Supervise and foster Built-in Quality
4.3	[S4RR233]	Define Solution Management responsibilities
4.3.1	[S4RR234]	Solution Management is responsible for the solution intent, which captures and documents fixed and variable solution level behaviors
4.3.2	[S4RR235]	Work with Release Management where applicable
4.3.3	[S4RR236]	Work with portfolio stakeholders, Customers, and ARTs to understand needs and build and prioritize the solution backlog
4.3.4	[S4RR237]	Plays a crucial role in pre- and post-PI planning, as well as value stream inspect and adapt workshops
4.3.5	[S4RR238]	Work with Suppliers, making sure the requirements for supplier-delivered capabilities are understood and assisting with the conceptual integration of these concerns
4.4	[S4RR239]	Define Supplier responsibilities
4.4.1	[S4RR240]	Working with Lean-Agile Suppliers
4.4.1.1	[S4RR241]	Treated like an Agile Release Train and works in the same cadence as the other ARTs
4.4.1.2	[S4RR242]	Participates in the PI Planning and Pre- and Post-PI Planning meetings, where they present what they plan to deliver in the next Program Increment, along with an indication of what will be delivered in each Iteration
4.4.1.3	[S4RR243]	Demos their subsystem or components in the System Demo, participates in the Solution Demo, and continually integrates their work with the rest of the value stream, providing feedback to other trains

4.4.1.4	[S4RR244]	Participates in Inspect and Adapt, both to improve the value stream as a whole and to help improve their own Lean-Agile practices
4.5	[S4RR245]	Define Customers responsibilities
4.5.1	[S4RR246]	Participate as Business Owner in PI Planning
4.5.2	[S4RR247]	Attend Solution and possibly System Demo; help evaluate the solution increment
4.5.3	[S4RR248]	Participate in Inspect and Adapt workshops; assist in removing some systemic impediments
4.5.4	[S4RR249]	Interact with analysts and subject matter experts during specification workshops
4.5.5	[S4RR250]	Collaboratively manage scope, time, and other constraints with Product and Solution Management
4.5.6	[S4RR251]	Help define the Roadmap, Milestones, and Releases
4.5.7	[S4RR252]	Communicate the economic logic behind the solution and help validate assumptions in the Economic Framework
4.5.8	[S4RR253]	Review technical and financial status of the solution
4.5.9	[S4RR254]	Participate in beta testing, UAT, other forms of solution validation

Appendix D

Global Agile Model for Enterprises (GAME)

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

The detail overview of GAME model incorporating above five layers is presents in the Table D.1.

Table D.1: SAFe practice mapping to GTM recommendations.

SG = GTM "Specific Goal"; SP = GTM "Specific Practice"; Sub P. = GTM "Sub Practice"; R = GTM "Recommendation"; T= SAFe Team-level practice; PR=SAFe Program-level practice; PO=SAFe Portfolio-level practice; VS=SAFe Value Stream-level practice; RR=SAFe Roles & Resp. practice;

ID	Description
SG 1	Define Global Project Management
SP 1.1	Global Task Management
Sub P. A	Determine team and organisational structure between locations
R A2.3	*Ensure that the supervision, support and information needs of all team members are met regardless of location.* **RR:** Product Owner provides necessary clarifications to assist team with their story estimation and story sequencing for the upcoming program increment **RR:** Scrum Master supports the product owner
R A2.4	*Organisational structure should be documented and available to all team to allow a clear understanding of everyone's roles and responsibilities within the project.* **RR:** Release Manager ensures that the organization's release governance is understood
Sub P. B	Determine the approach to task allocation between locations
R B2	*Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.* **T:** Decompose stories into tasks **T:** Prepare preliminary iteration goals **T:** Perform continuous integration **VS:** Collaborate with large stakeholder community to determine the best course of action
R B3	*Retain tasks that require frequent communication between groups within collocated teams.* **T:** Break larger initiatives into stories to ease estimation of larger work items **T:** Break each story into tasks then estimate and identify dependencies of a specific task.

Continued on next page.

ID	Description
R B4	*Where GSE teams are subdivided into work modules (e.g. different parts of the life-cycle), management must allocate tasks based on core competencies of each sub-team, and clearly define which stages are carried out at which location.* **PR:** Use Agile story point physics, based on normalized estimation, and estimate larger initiatives at the Epic level **PO:** Manage dependencies by applying extensive degree of cooperation; a common value stream backlog; implementation of new, crosscutting capabilities; additional system integration; additional roles and responsibilities; special considerations for pre-, post-, and PI planning activities; different degree and types of DevOps support **VS:** Adopt flow-based system economics that optimized by job-sequencing rather than by theoretical job ROI, or first-come, first-served job selection **VS:** Pull jobs into implementation based on WSJF, whereas estimate of job size is typically used as a proxy for duration **VS:** Work together to accomplish the system builder's larger objective **RR:** System Team assists component teams in defining inter-component interfaces **RR:** System and Solution Architect/Engineer defines subsystems and their interfaces **RR:** System and Solution Architect/Engineer allocates responsibilities to subsystems **RR:** System and Solution Architect/Engineer understands solution deployment **RR:** System and Solution Architect/Engineer communicates requirements for interactions with solution context
SP 1.2	Knowledge and Skills
Sub P. C	Identify business competencies required by global team members in each location
R C1	*Document and define customer base and functions relative to the application being developed.* **T:** Accept stories continuously to improve flow **T:** Use acceptance Test-Driven Development (ATDD) **PR:** Release (deliver) the product whenever business needs dictate. **PR:** Align releases to PI boundaries if business needs permit **PR:** Provide the necessary meaning to understand the feasibility of the solution and frame the right set of capabilities **PR:** Plan [learning] milestones incrementally **PR:** Perform system demos **PR:** Provide fast and intimate feedback

Continued on next page.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

ID	Description
	PR: Release more frequently to gain customer or user feedback about efficacy, deployability, and usability of the product **PO:** Bring structure and visibility to the analysis and decision-making that moves these initiatives into implementation **VS:** Perform Solution Demo **VS:** Supports compliance and contractual obligations **VS:** Empower people particularly Product and Solution Management with the relevant context, knowledge, and authority to make content decision at each level **VS:** Require evolution of the Customer's deployment environment for new solutions **RR:** Business Owner understands and assures that business objectives are understood and agreed to by key stakeholders of the train, including the Release Train Engineer (RTE), Product Management, and System Architects
Sub P. D	Identify the cultural requirements of each local sub-team
R D2	*Face-to-face meetings are recommended when and where possible, ideally at the start of the project and/or when a new member joins.* **T:** Team show a tested increment of value to product owner **T:** Integrate often
Sub P. E	Identify Communication Skills for GSE
R E1	*In order to develop the right practice, a new communication protocol needs to be set up. Policies should be put in place to support these new requirements to the satisfaction of all virtual team members. For example in synchronous communication, ensure that link up times are shared between core team working hours in each location.* **PR:** Provide Fast and intimate feedback from customers **VS:** Adopt Model-Based Systems Engineering that provides an effective way of reasoning about the solution and also serves as an efficient communication tool for sharing this knowledge **VS:** Encourages general knowledge discovery by making information, and related cross-discipline information, more accessible to teams
Sub P. F	Establish relevant criteria for training teams
R F2	*Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements* **VS:** Understand the Pre- and Post-PI Planning to provide for high-bandwidth communication through face-to-face alignment

Continued on next page.

ID	Description
R F3	*If working in regulatory domain, provide training on regulatory requirements and procedures.* **VS:** Generate documents for regulatory compliance (FAA, FDA, etc.) or contractual obligations (CDRLs in government contracting) when required
SP 1.3	Global Project Management
Sub P. G	Identify GSE project management tasks
R G1	*Define ability and potential productivity of team: Global project manager should allocate tasks and timescales that are realistic* **PR:** Use program backlog prioritization via weighted shorted job first (WSJF) **PR:** Apply WSJF for continuous prioritization of features in the program backlog **PO:** Provide WIP limits to ensure that the teams responsible for analysis undertake into responsibly and do not create expectation for implementation or time frames that far exceed capacity and reality
R G2	*Where possible, the project manager should be actively involved in the recruitment and selection of team members. Failing this, they should gather all information relating to the technical and professional experience of potential and existing team members.* **PR:** Work with the team to establish technical feasibility and scope estimates
R G3	*When teams are in place and project details reported project managers should understand and document how individuals contribute to that project along with their skills and knowledge.* **T:** Measure the team's capacity (based on previous sprint) to establish the velocity for the upcoming iteration **T:** Estimate initial capacity to establish a common starting point for team estimation **T:** Calculate team delivered velocity by multiplying the throughput by an average story size **T:** Use Capacity allocation technique to make a policy decision **T:** Build an approximation of team's current process and define some initial WIP limits **PR:** Apply capacity allocation to make a decision about how much of the total effort can be applied for each type of activity for an upcoming PI. The decision can be revisited as part of backlog refinement in preparation of each PI planning **PR:** Apply capacity allocation for enabler to work as a whole or to differentiate between various types of enablers **PR:** Apply normalized estimation technique during the backlog refinement or equivalent approach used by the Agile Teams to estimate stories

Continued on next page.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

ID	Description
	PO: Bring structure and visibility to the analysis and decision-making that moves these initiatives into implementation
	PO: Determine feasibility and potential ROI of all epics in portfolio Kanban in order to reach the portfolio backlog
R G4	*Plan for lessons learned to be recorded during the project and reviewed at the end by all project team members. Update procedures based on this experience*
	PR: Run a brief retrospective to identify whatever issues team would like to address
	RR: Product Owner participate in team demo and retrospective
	RR: Business Owner attends occasional Agile team iteration planning and iteration retrospective meetings, as appropriate
R G5	*Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global change management agent.*
	T: Perform retrospective at the end of each iteration
	T: Participate in Community of Practice (CoP)
	T: Perform Team Demo
	PR: Perform program backlog refinement
	PR: Develop program performance metrics
	PR: Release more frequently to gain customer or user feedback about efficacy, deployability, and usability of the product (Each release helps assess value of product in the development environment)
	PR: Create and Manage features
	PR: Perform system demos
	PR: Perform verification and validation of key scenarios and nonfunctional aspects, usually with the help of the System Team
	PR: Perform inspect and adapt (I&A) workshop
	PR: Run a brief retrospective to identify whatever issues team would like to address
	PR: Perform a structured, root cause analysis-based, problem-solving workshop for large, systematic program-level problems.
	PO: Facilitate continuous improvement by quantitative metrics, customer feedback, and the Inspect and Adapt retrospective cycle
	VS: Perform value stream backlog refinement
	VS: Elaborate capability
Sub P. H	Assign tasks to appropriate team members
R H1	*Assign according to one or more of three different approaches; Modularisation; Phase-based approach; and Integrated approach.*
	PR: Integrate a subset of capabilities, components, or subsystems

Continued on next page.

ID	Description
	PR: Integrate to illustrate a particular feature, capability, or Nonfunctional Requirements **PR:** Integrate with the support of prototypes and mock-ups **PR:** Integrate ever other iteration **PR:** Provide guidance for cross-team design and implementation synchronization **PO:** Encourage and facilitate incremental development and fast customer feedback **VS:** Work together to accomplish the system builder's larger objective **VS:** Collaborate and integrate with the others to provide the operational value stream with a seamless, end-to-end solution **RR:** System and Solution Architect/Engineer works with Product and Solution Management to determine capacity allocation for enablement work **RR:** System and Solution Architect /Engineer defines, explores, and support the implementation of value stream and program enablers to evolve solution intent; work directly with Agile
Sub P. J	Establish cooperation and coordination procedures between locations
R J1	*Ensure that a suitable infrastructure, process and management procedures are in place to help establish cooperation and coordination between locations.* **T:** Measure the team's capacity (based on previous sprint) to establish the velocity for the upcoming iteration **T:** Perform Program Increment (PI) planning to understand and agree on one or more iteration goal(s) based on the team and PI Objectives **T:** Build an approximation of team?s current process and define some initial WIP limits **T:** Estimate initial capacity to establish a common starting point for team estimation **T:** Use objective measure that includes average lead time (how long it takes, on average to get through the system), WIP (the average number of items in the system), and throughput (the number of stories completed per a period of time) by Cumulative Flow Diagram (CFD) **T:** Differentiate between Features and Objectives to reduce the confusion **T:** Stretch objectives to provide a way to help that ensure the delivery time-box will be met **T:** Write SMART Objectives **T:** Assign business value to each of the teams individual objectives [Communicate Business Value with Objectives] **T:** Finalize the Team PI Objectives with business value and stretch objectives

Continued on next page.

ID	Description
	T: Commit to PI Objectives by agreeing to do everything in the team's power to meet the committed objectives **T:** Decrease excess WIP to reduce overhead and thrashing, and it increase productivity and velocity. [Shed Excess WIP with Realistic Objectives] **T:** Planning at regular PI intervals limits variances to a single PI time-box, thereby increasing Agile Release Train and Value Stream predictability [Use a regular cadence to limit the accumulation of variance] **PR:** Understand queuing theory and aware of the impact of long queues have on delivery time **PR:** Build a system increment every two weeks comprising all team backlog items completed by all release train teams during current and all previous iterations **PR:** Perform system demo after each system increment **PR:** Perform system demos **PR:** Release on the program increment cadence **PR:** Release less frequently **PR:** Release more frequently **PR:** Perform verification and validation of key scenarios and nonfunctional aspects, usually with the help of the System Team **PR:** Perform a final system demo of all features developed in the PI, during the Inspect and Adapt workshop at the PI boundary. **PR:** Perform inspect and adapt (I&A) workshop **PR:** Perform solution verification and validation prior to actual release to customers **PR:** Document system prior to actual release to customers **PR:** Transition solution assets to deployment organization for actual delivery to customers **PR:** [Design] solution with 'release-ablity' in mind reflecting key release concerns and constraints **PR:** [Validate 'release-ability'] assumptions as early and frequently as possible using external releases at every PI at least **VS:** Provide fund to value streams, not project **RR:** Value Stream Engineer encourages the collaboration between teams and System and Solution Architects, Engineering, and User Experience designers **RR:** RTE encourages the collaboration between teams and System and Solution Architects, Engineering, and User Experience designers
R J2	*Achievable milestones should be planned and agreed. Projects should be monitored with reference to costs, time, productivity, quality and risk.*

Continued on next page.

ID	Description
	PR: Measure Team PI Performance Report to calculate the program predictability measure (Team PI Performance Report compares actual business value achieved against planned business value) **PR:** Create an objective measure of progress **PR:** Measure, assess, and evaluate the system by the relevant stakeholders throughout the development **PR:** Measure Program Predictability at the end of each PI **PR:** Measure Program Features planned vs. accepted at the end of each PI. **PR:** Measure Program Enabler Features planned vs. accepted at the end of each PI. **PR:** Measure Program stories planned vs. accepted at the end of each PI. **PR:** Measure Program unit test coverage at the end of each PI **PR:** Measure Program defects at the end of each PI **PR:** Measure Program total tests at the end of each PI **PR:** Measure Program automated test fraction at the end of each PI **PR:** Measure number of Program Non Functional Requirements (NFR) tests at the end of each PI **PR:** Use capacity allocation to ensure continuous investments in enablers **PO:** Provides the spending and personnel allocations necessary to accomplish the portfolio vision **PO:** Identify, elaborate, estimate, and analyze all epics that reached in mature state to achieve a go recommendation from PPM **VS:** Review the progress in the broader portfolio context and make decisions about the continuation or cancellation of initiatives or changes to the budgetary investment in the value streams **RR:** Product Manager develops and communicates the program vision and roadmap **RR:** Business Owner prepares to communicate the business context, including milestones and significant external dependencies, such as those of suppliers **RR:** Customer collaboratively manages scope, time, and other constraints with Product and Solution management
Sub P. K	Establish reporting procedures between locations
R K1	*Regular formal reporting will help the project manager to remain aware of how project is progressing.* **PR:** Use the PI burn-down chart to track the work planned vs. accepted for a PI **VS:** Develop a lightweight business case and an explicit approval process for each epic

Continued on next page.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

ID	Description
R K2	*Procedure should include and encourage team members to report whether or not they can take on that task in the given time and report any problems before it is too late.* **T:** Demonstrate completed story and summarize team's increment **PR:** Represent the progression of features through stages (Funnel, Backlog, Review, Analysis, Execution, and Done) using a Cumulative Flow Diagram **VS:** Records and communicate requirements, design, and system architecture decisions **VS:** Record any requirements and design decisions in only one place, a single source of truth that serves as the repository of record for all **RR:** Epic Owner understands and reports on progress of the epic with key stakeholders
Sub P. L	Establish a Risk Management Strategy
R L1	*All potential risks should be identified and addressed to include: risks in misunderstanding cultural differences, misunderstanding requirements, feature volatility, schedules, budgets, personnel.* **PR:** Identify current risks and impediments **PO:** Impact on the economic framework from strategic themes could affect any of the major parameters including development/cycle time, product cost, product value, development expense, and risk **VS:** Manage risk by adopting multiple potentially viable solution contexts **RR:** Scrum Master eliminates impediments **RR:** Value Stream Engineer helps to manage risks and dependencies **RR:** Release Train Engineer helps to manage risks and dependencies
R L2	*In addition, risk associated with outsourcing activities to politically unstable locations needs to be identified.* **RR:** Value Stream Engineer helps to manage risks and dependencies **RR:** Release Train Engineer provides input on resourcing to address critical bottlenecks
R L3	*Identify and address local project demands that may pose a risk to the global project.* **PR:** Teams commit to meet the program PI Objectives for the upcoming PI **PR:** Support the goals of synchronization, alignment, management of variability, and predictability of development velocity **PR:** Work with the team to establish technical feasibility and scope estimates **PR:** Apply capacity allocation to make a decision about how much of the total effort can be applied for each type of activity for an upcoming PI

Continued on next page.

ID	Description
	PR: Establish an agreement to determine how the work is performed for each activity type **PR:** Limit the commitments to longer-term work, because some other item may come along that's more important than a prior commitment **PR:** Identify risk reduction-opportunity enablement value **PR:** Identify dependencies and forecast cross-team and cross-ART coordination. **PR:** Match demand to capacity and eliminate excess WIP **PR:** Perform a structured, root cause analysis-based, problem-solving workshop for large, systematic program-level problems. **PO:** Impact on the economic framework from strategic themes could affect any of the major parameters including development/cycle time, product cost, product value, development expense, and risk **PO:** Allocate budget authority to the decision makers **PO:** Facilitate continuous improvement by quantitative metrics, customer feedback, and the Inspect and Adapt retrospective cycle **PO:** Provide a quantitative, transparent basis for economic decision-making for these, the most important business decisions **PO:** Exercise Fiscal Governance with Dynamic Budgeting. **PO:** Manage dependencies by applying extensive degree of cooperation; a common value stream backlog; implementation of new, crosscutting capabilities; additional system integration; additional roles and responsibilities; special considerations for pre-, post-, and PI planning activities; different degree and types of DevOps support. **RR:** Release Train Engineer escalates and tracks impediments **RR:** Business Owner actively address impediments, especially those that escalate beyond the authority of the key stakeholders on the
SG 2	Define Management Between Locations
SP 2.1	Operating Procedures
Sub P. M	Define how conflicts and differences of opinion between locations are addressed and resolved
R M1	*Set up a strategy to handle, monitor and anticipate where conflict between remote locations may occur. The strategy should include how conflict will be resolved and how a person responsible for that resolution is selected.* **PO:** Define strategic themes in the individual SAFe portfolio context **RR:** Value Stream Engineer escalates and tracks impediments

Continued on next page.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

ID	Description
R M2	*When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences* **T:** Facilitate trust among team members by having a common mission, common iteration goals, and team Program Increment (PI) objectives **T:** Develop program increment objectives **T:** Prepare preliminary iteration goals **T:** Perform Daily Stand-up to understand team's status, escalate problems, and get help from other team members **T:** Create a Community of Practices (CoP) for software quality/continuous integration/test first/refactoring **T:** Perform Continuous Integration **PO:** Define strategic themes in the individual SAFe portfolio context **PO:** Define strategic themes that connect SAFe portfolio to evolving enterprise business strategy
R M3	*Establish criteria for global projects based on global strategy and vision to guide decision making* **PO:** Define strategic themes in the individual SAFe portfolio context **VS:** Adjust value stream budget over time at PI boundaries, based on the relative value that each value stream provides to the portfolio **RR:** Epic Owner establish cost of delay and identify business sponsors
Sub P. N	Implement a communication strategy for the team
R N1	*Plan, facilitate, encourage and monitor communication between teams.* **T:** Perform constant communication and collaboration (for fast, effective, and empowered decision making, and to help individuals meet their responsibilities). **T:** Perform Daily Stand-up to understand team's status, escalate problems, and get help from other team members **T:** Use "big visual information radiator" (BVIRs) to understand and track progress during iteration execution **T:** Use story board to visualize the stories and their progress throughout the iteration **T:** PI planning, iteration planning, backlog refinement, inspect and adapt, architecture discussion, etc., all benefits from frequent meetings [Schedule frequent meetings using a predictable cadence] **RR:** Scrum Master coordinate with other teams **RR:** System Team attend other teams' stand-ups in support of daily activities **RR:** System Team assist teams in creating reduces test suites that they themselves can run

Continued on next page.

ID	Description
	RR: System and Solution Architect /Engineer defines, explores, and supporst the implementation of value stream and program enablers to evolve solution intent; work directly with Agile Teams to implement, explore, or support them
Sub P. O	Establish communication interface points between the team members
R O1	*Strategies need to be put in place which encourage both formal and informal reporting.* **PR:** Establish high-bandwidth communication across all team members and stakeholders **PO:** Bring structure and visibility to the analysis and decision-making that moves these initiatives into implementation
R O2	*Ensure that relevant team members are made aware of how and when they will receive inputs to products, needs to distribute outputs from and when complete work products are required.* **T:** Perform continuous integration **T:** Visualize and Limit Work in Progress **T:** Participate in community of practice **PR:** Perform system demo **PR:** Develop PI planning **PR:** Perform inspect and adapt (I&A) workshop **PO:** Track epics and enablers to understand which ones are being worked on and which have been completed **PO:** Develop a Kanban system to capture, analyze, approve, and track epics **RR:** Scrum Master builds a high-performing team **RR:** Epic Owner works with development team to size the epic and provide input for economic prioritization based on WSJF
R O4	*Ensure that information about each team member is easily accessible by colleagues. Information of an individual's role within the team and their specific areas of responsibility should be combined with a photograph, their first name, surname, friendly name (if appropriate) and their preferred form of address.* **RR:** Scrum Master supports the product owner
R O5	*Intranets and wikis can be invaluable for this form of communication.* **PR:** Build social network the ART depends upon
Sub P. P	Implement strategy for conducting meetings between locations
R P2	*Try to ensure all participants are comfortable with virtual meeting and are given opportunity to agree or disagree with points raised, and offer new ideas.* **RR:** RTE facilitate periodic synchronization meetings, including the ART sync at the Program level and the Value Stream (VS) sync at the value stream level

Continued on next page.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

ID	Description
	RR: Participate in certain key ceremonies, including presentation of vision, draft plan review, assigning final business value to PI Objectives, and approving final plans
R P3	*Circulate agenda prior to meeting, and clearly minute actions agreed a meeting* **RR:** Facilitates meetings **RR:** Actively circulate during planning, communicate business priorities to the teams, and maintain agreement and alignment among the stakeholders as to the key objectives of the train
SP 2.2	Collaboration between locations
Sub P. Q	Identify common goals, objectives and rewards for the global team
R Q3	*Project goals and objectives communicated, understood and agreed across all team members regardless of location.* **T:** Develop program increment objectives **T:** Prepare preliminary iteration goals **PR:** Develop PI planning **PO:** Define and analyze epics to provide visibility and economic justification for upcoming crosscutting work **PO:** Develop lightweight business cases to provide reasoning, analysis, and prioritization while avoiding over specificity **PO:** Ensure that the enterprise moves forward with each value stream in lockstep with the enterprise objectives **PO:** Reduce time-to-market with value stream mapping **VS:** Use the acceptance criteria to determine whether the functionality has been properly implemented
R Q6	*Project Managers need to understand the cultural motivation of the different team members and identify and apply appropriate rewards in each situation when and where relevant* **RR:** Facilitates the team's progress toward the goal
Sub P. R	Collaboratively establish and maintain work product ownership boundaries
R R1	*Define product ownership boundaries through partitioning of work across GSE teams* **VS:** Include solution context surface as part of the Definition of Done for a solution Increment
R R2	*Each location should understand their role within the life cycle of the product.* **T:** Team plan together to create a common vision and roadmap **PR:** Develop PI planning

Continued on next page.

ID	Description
R R3	*Each location should understand how their modifications to the product unit can affect the other locations.* **T:** Team integrates and demonstrates together **T:** Teams integrate and evaluate (at least) on iteration boundaries; program and value streams integrate and evaluate on PI boundaries. [Apply nested cadence harmonic multiples to synchronize work] **T:** Perform Team Demo **T:** Place PI system demo, inspect and adapt workshop, PI planning, and backlog refinement in a dedicated IP iteration **T:** Implement feature and component Level continuous integration **T:** Enable continuous integration **T:** Common Cadence **T:** Infrastructure **T:** Engineering Techniques in Support of CI **T:** Support of the System Team **T:** Make Continuous Integration a Culture **T:** Integrate often **T:** Make integration results visible **T:** Make fixing a failed integration attempt the top priority **PR:** Apply Continuous Integration practices with test automation whenever feasible to monitor and ensure progress. **PR:** Perform program backlog refinement **PR:** Elaborate feature **PR:** Develop infrastructure in different level of SAFe to support frequent or continuous integration and testing **VS:** Perform value stream backlog refinement **VS:** Elaborate capability **VS:** Use the acceptance criteria to determine whether the functionality has been properly implemented **VS:** ARTs operate synchronously and build the solution in increments, which are fully integrated and evaluable via the Solution Demo **VS:** Allow developers to reliably make changes without causing an error in another part of the system by creating a rich suite of test cases **RR:** Update and review team backlog and contribute to the vision, roadmap, and content presentations
Sub P. S	Collaboratively establish and maintain interfaces and processes
R S1	*Define common process goals across all locations.*

Continued on next page.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

ID	Description
	T: Individual Agile Teams are aligned to common iteration lengths [Exploit economic of scale by synchronizing work from multiple projects]
	T: Teams are aligned to common time-boxes and similar batch sizes [To reduce queues, synchronize the batch size and timing of adjacent processes]
	T: Perform Team Demo
	T: Place PI system demo, inspect and adapt workshop, PI planning, and backlog refinement in a dedicated IP iteration
	T: Test-first
	T: Use the Agile Testing Matrix
	T: Apply Test-Driven Development
	T: Perform Unit Test
	T: Perform Component Tests
	T: Use acceptance Test-Driven Development
	T: Perform Functional Tests
	T: Automate Acceptance Testing
	T: Use acceptance test template/checklist
	T: Refactor
	T: Specify refactors
	T: Split refactors
	T: Establish acceptance criteria
	T: Demonstrate refactors
	T: Adopt the Culture
	T: Create a Community of Practices (CoP) for software quality/continuous integration/test first/refactoring
	T: Work in Pairs
	T: Develop a collective ownership
	T: Build in quality in the Firmware and Hardware
	T: Use exploratory early iterations
	T: Frequent integrate and test at System-Level
	T: Apply design verification
	PR: Perform program backlog refinement
	PR: Elaborate feature
	VS: Perform value stream backlog refinement
	VS: Elaborate capability
	VS: Allocate fund to the value streams
	RR: Update and review team backlog and contribute to the vision, roadmap, and content presentations

Continued on next page.

ID	Description
R S2	*Define process ownership – placing ownership with those closest to process where possible.* **VS:** Collaborate and integrate with the others to provide the operational value stream with a seamless, end-to-end solution **RR:** Provide necessary clarifications to assist team with their story estimation and story sequencing for the upcoming program increment
R S3	*Seek and encourage input from team members at all locations.* **T:** Improve continuous collaboration by using regular feedback loops. **PO:** Encourage and facilitate incremental development and fast customer feedback **RR:** Provide necessary clarifications to assist team with their story estimation and story sequencing for the upcoming program increment
R S4	*Let team members know their input to process development and ownership is valued.* **T:** Perform Iteration Retrospective to identify way to improve **T:** Team learn together and share best practices through inter-team communication supported by Communities of Practice (CoP). **PR:** Represent the progression of different areas such as Funnel, Backlog, Review, Analysis, Execution, and Done to develop cumulative flow diagram **VS:** Participate in the creation, feedback, and refinement of solution intent information **RR:** Update and review team backlog and contribute to the vision, roadmap, and content presentations **RR:** Lead the team's efforts in relentless improvement
R S6	*Processes should take into account the relevant structures and procedures from all sites.* **VS:** Store information into repositories that can be used for inspections and formal reviews
Sub P. T	Collaboratively develop, communicate and distribute work plans.
R T1	*Achievable milestones should be planned and agreed.* **T:** In order to reliably meet PI objectives and planning iteration is unplanned and provides schedule margin [Provide sufficient capacity margin to enable cadence] **T:** Short iterations help to control the number of stories in the iteration batch [Use a regular cadence to enable small batch sizes] **PO:** Success criteria provide learning milestones that allow the portfolio to understand the solution involved, validate business and technical hypotheses, and pivot towards a better solution **PO:** Define Success criteria to validate the implementation

Continued on next page.

ID	Description
	VS: Integrates the entire system-of-systems value stream to demonstrate progress toward the top-level context's Milestone and Release commitments
	VS: Clarify the intent, validate assumptions, and review progress
	VS: Perform continuous exploration that includes learning Milestones, Customer feedback loops, and Set-Based Design informs and streamlines the learning process by validating good options and eliminating less viable ones
	RR: Prepare to communicate the business context, including milestones and significant external dependencies, such as those of suppliers
	RR: Define the roadmap, milestones, and releases
R T2	*Within the commitments made, team members must explicitly include communication plans to include use synchronous and asynchronous communication tools.*
	VS: Integrates the entire system-of-systems value stream to demonstrate progress toward the top-level context's Milestone and Release commitments
	VS: Facilitate further exploration and analysis activities
	RR: Facilitates preparation and readiness for ART ceremonies
	RR: Coordinate and synchronize epic-related activities with functions in sales, marketing, and other business units
R T3	*Contingency plans should be in place to address potential risks.*
	T: If a feature doesn't make it into a PI (or release) and it remains high priority, its delivery can be anticipated to be on schedule in the next PI [Use cadence to make waiting time predictable]
	T: Teams plan with stretch objectives; these are sacrificed as necessary when plan meet reality [Capacity margin enables synchronization of deliverables]
	T: Consider challenges and trade-offs
R T4	*Establish procedures to coordinate implementation of contingencies when and if required.*
	T: Perform backlog refinement to underline the problems of current plan
	T: Synchronize with Supplier and Solution Context
	VS: Specifies how the solution is packaged, deployed, and integrated
	VS: Clarify the intent, validate assumptions, and review progress
	VS: Drive development, make scope and priority decisions, and manage the flow of Features and Capabilities and Nonfunctional Requirements
	VS: Requires a holistic approach to definition, planning, implementation, and review of the solution to develop an economically viable solution
	RR: Influence common modeling, design, and coding practices
	RR: Collect, generate, and analyze innovative ideas and technologies that are applicable across the enterprise
	RR: Facilitate the reuse of ideas, components, and proven pattern

Continued on next page.

ID	Description
	RR: Synchronize System and data security and quality, Production infrastructure, Solution User Experience governance, Scalability, performance, and other NFRs across solution whenever applicable **RR:** Create and maintain infrastructure, including continuous integration, automated builds, and automated build verification testing
SG 3	Define Global Regulation Management
Sub P. U	*Establish a risk management strategy for regulation*
R U	*Establish a risk management strategy for regulation* **VS:** Manage risk by adopting multiple potentially viable solution contexts **VS:** Collaborate (System Engineers) with Customers and/or regulatory agencies on the minimum set sufficient to meet their obligations
Sub P. V	*Collaboratively plan, develop and validate systems and changes*
R V	*Collaboratively plan, develop and validate systems and changes* **T:** Perform Continuous Integration **PR:** Develop PI commitment **PR:** Forcase PIs **PR:** Develop Program Backlog **PR:** Estimate larger initiatives at the Epic level using Agile story point physics, based on normalized estimation. **PR:** Use short "committed queues" to reduce delivery time for new initiatives. **PR:** File patent, certify the system, audit certain regulatory requirements for economic success of product development **PO:** Influence vision **VS:** Develop Solution Intent **VS:** Continuous Collaboration Ensures Deployability **VS:** Frequently evaluate integrated increments of the entire solution for effective solution development **RR:** Facilitate the technical aspects of collaboration with third parties, such as data or service providers, hosting facilities, etc.
Sub P. W	*Coordinate for traceability of requirements.*
R W	*Coordinate for traceability of requirements.* **PR:** Continuous evolution of the Architectural Runway supports current and near-term Features. **PR:** [Agile] teams continuously communicate emerging requirements and opportunities back into the program vision.

Continued on next page.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

ID	Description
	PR: Use a Feature Progress Report to show planned and actual stories for a feature. **PR:** Use a Program Kanban Board to ensure that features are reasoned and analyzed prior to reaching a PI boundary **PR:** Use a Program Kanban Board to ensure that acceptance criteria have been established to guide a high-fidelity implementation **PR:** Use a Program Kanban Board to track features that are being worked on or completed. **VS:** Track changes as both the system and deployment environment have to evolve to a common state **RR:** Trace significant external commitments and dependencies

D.1 Summary

There were three categories I have identified as a result of the mapping exercise. However, I have only presented SAFe practices that implement GTM recommendations. SAFe practices that are not implemented GTM recommendations and GTM recommendations that are not implemented were excluded from the model development. In total, there are 321 SAFe practices that mapped onto 50 GTM recommendations. Of these, 10 SAFe practices were found duplicate, and 834 SAFe practices did not map onto any GTM recommendation.

D. GLOBAL AGILE MODEL FOR ENTERPRISES (GAME)

Appendix E

GTM Assessment and Cost-benefit analysis

E.1 GTM Assessment

This appendix presents results derived from participant responses to the GSD Process Assessment questionnaire C.1. Respondents were asked to rate Ocuco's degree of implementation of 70 GSD practices from the Global Teaming Model, using a 10-point scale as follows:

1. Implemented, works well for me.
2. Implemented, just working this way, not sure of its value though.
3. Implemented, but needs improving as it is not effective in its current form.
4. Implemented, but informally–wouldn't be able to tell you the precise process.
5. Started implementing, then stopped, planning to start when time (sporadic implementation).
6. Planning to implement in near future. –i.e. would like to implement but haven't yet.
7. Would like to implement but can't: Not implemented because my organisation does not have resources, or management won't approve (i.e. in an ideal world would like to, and know about it already, but cannot afford to do this).
8. Not implemented because I didn't know about the practice (i.e. now I know, it seems a good idea and appropriate for my needs; I will think about introducing this practice).
9. Not Needed: Not implemented because it is not necessary for my type of business (e.g., business is too small, don't have any problems that this practice would solve).
10. I don't really understand what the practice is advocating so cannot comment.

Each practice was then assigned to one of five categories as follows:

Fully Implemented If the number of responses with value 1 was greater than the number of responses with value 2-8, the practices was categorized as *fully implemented*.

Largely Implemented If the practice was not categorized as fully implemented, and the number of responses with value 1, 2 or 4 was greater than the number of responses with value 3, or 5-8, the practice was categorized as *largely implemented*.

Partially Implemented If the practice was not categorized as fully or largely implemented, and the number of responses with value 1-5 was greater than the number of responses with values 6, 7, or 8, the practice was categorized as *partially implemented*.

Not Adequately Implemented If the practice did not qualify as fully, largely, or partially implemented, and the number of responses with value 1-8 was greater than the number of responses with value 9, 10, or 11, the practice was categorized as *not adequately implemented*.

Not Needed Practices not meeting the above criteria, but having more responses with value 9 than value 10 or 11, were deemed to be *not needed*. These practices don?t apply to Ocuco's business or product domain.

Other Practices not meeting any of the above criteria were assigned to category *other*.

In Tables below, practices are grouped according to the category assigned using the procedure above, based on the totality of responses. The columns indicate the score computed for each sub-group, as well as the difference between the highest and lowest scores among sub-groups ("agreement gap").

Table E.1: Practices fully implemented.

ID	Practice	Team A	Team B	Team C	Agre. gap
A1	Create roles, relationships and rules to facilitate coordination and control over geographical, temporal and cultural distance.	Fully	Fully	Fully	0

Continued on next page

Table E.1 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
A2	Structure global team and monitor operation to minimize fear and alienation in teams	Fully	Largely	Fully	1
B6	Related non confidential development activities can be undertaken by external remote team colleagues.	Fully	Largely	Not Needed	4
D2	Face-to-face meetings are recommended when and where possible, ideally at the start of the project and/or when a new member joins.	Fully	Fully	Largely	1
E1	In order to develop the right practice, a new communication protocol needs to be set up. Policies should be put in place to support these new requirements to the satisfaction of all global team members. For example in synchronous communication, ensure that link up times are shared between core team working hours in each location.	Fully	Fully	Partly	2
G1	Define ability and potential productivity of team: Global project manager should allocate tasks and timescales that are realistic.	Partly	Fully	Fully	2
G2	Where possible, the project manager should be actively involved in the recruitment and selection of team members. Failing this, they should gather all information relating to the technical and professional experience of potential and existing team members.	Fully	Largely	Fully	1

Continued on next page

Table E.1 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
I2	Project managers should ensure that cultural profiles for teams are established. E.g., Management and staff should show respect for gender-related cultural values of all colleagues. All employees? legal rights must be upheld.	Fully	Fully	Partly	2
J1	Ensure that a suitable infrastructure, process and management procedures are in place to help establish cooperation and coordination between locations.	Fully	Largely	Fully	1
J2	Achievable milestones should be planned and agreed. Projects should be monitored with reference to costs, time, productivity, quality and risk.	Fully	Largely	Fully	1
K1	Regular formal reporting will help the project manager to remain aware of how project is progressing.	Fully	Largely	Fully	1
K2	Procedure should include and encourage team members to report whether or not they can take on that task in the given time and report any problems before it is too late.	Fully	Largely	Fully	1
M3	Establish criteria for global projects based on global strategy and vision to guide decision making.	Partly	Fully	Fully	2
N1	Plan, facilitate, encourage and monitor communication between teams.	Fully	Fully	Partly	2

Continued on next page

Table E.1 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
N3	Consider linguistic and cultural implications inherent when communicating remotely.	Largely	Fully	Not Adequate	3
O1	Strategies need to be put in place which encourages both formal and informal reporting.	Fully	Largely	Fully	1
O2	Ensure that relevant team members are made aware of how and when they will receive inputs to products they are working on, and when they need to distribute outputs from these products and when complete work products are required.	Partly	Fully	Fully	2
O3	Ensure teams are aware of potential constraints such as legal restrictions and holidays in countries within which they are developing the product.	Fully	Fully	Partly	2
O4	Ensure that Information about each team member is easily accessible by colleagues. Information of an individual?s role within the team and their specific areas of responsibility should be combined with a photograph, their first name, surname, friendly name (if appropriate) and their preferred form of address.	Fully	Fully	Largely	1
O5	Intranets and wikis can be invaluable for this form of communication.	Fully	Fully	Partly	2
P1	Identify appropriate global meeting technology is used.	Fully	Fully	Fully	0

Continued on next page

Table E.1 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
P2	Try to ensure all participants are comfortable with global meeting and are given opportunity to agree or disagree with points raised, and offer new ideas.	Partly	Fully	Fully	2
P3	Circulate agenda prior to meeting, and clearly minute actions agreed a meeting.	Partly	Fully	Fully	2
Q1	Global Project manager sets project goals and objectives.	Largely	Fully	Fully	1
Q2	Goals at project level are common to all locations.	Largely	Fully	Fully	1
Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location.	Partly	Fully	Fully	2
Q4	The global team is viewed as an entity in its own right, regardless of the location of its team members and its performance should be judged and rewarded accordingly.	Fully	Fully	Fully	0
R1	Define product ownership boundaries through partitioning of work across GSE teams.	Partly	Fully	Partly	2
R2	Each location should understand their role within the life cycle of the product.	Partly	Fully	Largely	2
S1	Define common process goals across all locations.	Largely	Fully	Fully	1
S2	Define process ownership – placing ownership with those closest to process where possible.	Fully	Fully	Largely	1
S3	Seek and encourage input from team members at all locations.	Fully	Fully	Fully	0
S5	Processes should address specific challenges associated with GSE.	Fully	Largely	Fully	1

Continued on next page

Table E.1 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
S6	Processes should take into account the relevant structures and procedures from all sites.	Partly	Largely	Fully	2
T1	Achievable milestones should be planned and agreed.	Partly	Fully	Fully	2
T2	Within the commitments made, team members must explicitly include communication plans to include use synchronous and asynchronous communication tools.	Partly	Fully	Fully	2
T3	Contingency plans should be in place to address potential risks.	Partly	Fully	Fully	2

Table E.2: Practices largely implemented.

ID	Practice	Team A	Team B	Team C	Agre. gap
B1	Identify and document reason for working with global team.	Fully	Partly	Fully	2
B4	Where GSE teams are subdivided into work modules (e.g. different parts of the life-cycle), management must allocate tasks based on core competencies of each sub-team, and clearly define which stages are carried out at which location.	Largely	Fully	Not Needed	4
B5	Confidential software development activities that provide competitive advantage should be developed within organization.	Partly	Partly	Not Needed	2
C1	Document and define customer base and functions relative to the application being developed.	Partly	Largely	Fully	2

Continued on next page

Table E.2 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
C2	Provide training to ensure that global team has required understanding of the customer base and the business functions to take full advantage of the proximity of the team to the customer base.	Partly	Largely	Fully	2
D1	Cultural diversity: Each team member should be trained to understand the culture of the global team.	Largely	Largely	Fully	1
D3	Having individuals visit locations for extended periods can also be a successful strategy and should be fully leveraged at every possible opportunity.	Largely	Fully	Largely	1
F1	Effective knowledge transfer: Carry out evaluation of training needs to include cultural and linguistic issues.	Partly	Fully	Largely	2
F3	If working in regulatory domain, provide training on regulatory requirements and procedures.	Partly	Largely	Fully	2
G3	When teams are in place and project details reported project managers should understand and document how individuals contribute to that project along with their skills and knowledge.	Largely	Partly	Largely	1
G4	Plan for lessons learned to be recorded during the project and reviewed at the end by all project team members. Update procedures based on this experience.	Largely	Partly	Fully	2

Continued on next page

Table E.2 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
H1	Assign according to one or more of three different approaches; Modularization; Phase-based approach; and Integrated approach.	Other	Largely	Largely	4
L1	All potential risks should be identified and addressed to include: risks in misunderstanding cultural differences, misunderstanding requirements, feature volatility, schedules, budgets, personnel.	Partly	Partly	Fully	2
L2	In addition, risk associated with outsourcing activities to politically unstable locations needs to be identified.	Not Needed	Largely	Not Needed	3
L3	Identify and address local project demands that may pose a risk to the global project.	Partly	Fully	Fully	2
M1	Set up a strategy to handle, monitor and anticipate where conflict between remote locations may occur. The strategy should include how conflict will be resolved and how a person responsible for that resolution is selected.	Partly	Fully	Fully	2
M2	When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences.	Partly	Fully	Not Adequate	3

Continued on next page

Table E.2 – *Continued from previous page*

ID	Practice	Team A	Team B	Team C	Agre. gap
N2	Provide training on how best to communicate with remote colleagues, including the effective operation of communication tools and procedures.	Partly	Fully	Partly	2
P4	Ensure that no delay occurs between the meeting and the circulation of minutes as people may be waiting for the minutes before implementing the actions.	Partly	Fully	Partly	2
Q5	Acknowledging team success may require tailoring rewards to the needs of different cultures.	Partly	Fully	Largely	2
Q6	Project Managers need to understand the cultural motivation of the different team members and identify and apply appropriate rewards in each situation when and where relevant.	Partly	Partly	Largely	1
Q7	Consideration should be given to cultural issues, economic situation and income tax laws when planning rewards.	Largely	Partly	Largely	1
R3	Each location should understand how their modifications to the product unit can affect the other locations.	Partly	Largely	Fully	2
S4	Let team members know their input to process development and ownership is valued.	Largely	Fully	Partly	2
T4	Establish procedures to coordinate implementation of contingencies when and if required.	Partly	Largely	Fully	2
U	Sub Practice U. “Establish a risk management strategy for regulation”	Other	Largely	Not Adequate	4

Table E.3: Practices partially implemented.

ID	Practice	Team A	Team B	Team C	Agre. gap
B2	Base task allocation on the organizational requirement, e.g., if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Fully	Partly	Largely	2
B3	Retain tasks that require frequent communication between groups within collocated teams.	Largely	Partly	Not Adequate	2
F2	Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements.	Partly	Fully	Largely	2
G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global change management agent.	Partly	Partly	Largely	1
I1	National cultural differences should be identified and communicated to the management and team members.	Partly	Partly	Partly	0
V	Sub Practice V: "Collaboratively plan, develop and validate systems and changes"	Partly	Partly	Fully	2

We have not found any practice that is not adequately implemented or not relevant.

E.2 Cost-benefit Analysis

The cost-benefit analysis assessed the costs and benefits of making improvements.

Table E.4: Cost and Benefit Values.

Rating	Cost	Benefit
Definitely High	Equivalent to adding another staff member	Save a sprint of team-effort
Somewhat High	Equivalent to Dev-Con	Save a sprint of developer effort
Somewhat Low	Equivalent to one-day team on-site	Save a developer-day of effort
Definitely Low	Equivalent to a new PC	Save a developer-hour of effort

First, each practice was rated on the relative cost of implementing a practice on a four-point scale from low to high as shown in table "Cost and Benefit Values." Then, a similar assessment of the benefit of implementing a practice was made, again on a four-point scale from low to high.

The resulting score is computed as *Score=cost* x *benefit*, where cost is a value from 4 (low) to 1 (high), and benefit is a value from 1 (low) to 4 (high).

Table E.5: Cost-benefit ranking of practices in need of improvement.

ID	Practice	Cost	Benefit	Impact
Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location.	Definitely Low	Definitely High	16
O2	Ensure that relevant team members are made aware of how and when they will receive inputs to products they are working on, and when they need to distribute outputs from these products and when complete work products are required.	Definitely Low	Definitely High	16

Continued on next page

Table E.5 – *Continued from previous page*

ID	Practice	Cost	Benefit	Impact
B2	Base task allocation on the organizational requirement, e.g., if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Definitely Low	Definitely High	16
V	Sub Practice V: "Collaboratively plan, develop and validate systems and changes"	Somewhat Low	Definitely High	12
G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global change management agent.	Somewhat Low	Somewhat High	9
M2	When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences.	Somewhat Low	Somewhat Low	6
F2	Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements.	Somewhat High	Somewhat High	6
C1	Document and define customer base and functions relative to the application being developed.	Somewhat High	Somewhat High	6
I1	National cultural differences should be identified and communicated to the management and team members.	Somewhat Low	Definitely Low	3
B3	Retain tasks that require frequent communication between groups within collocated teams.	Somewhat High	Definitely Low	2

Appendix F

Interview Protocol

F.1 Overview

The purpose of this interview is to help us understand how Ocuco implements SAFe at the Team, Program, and Portfolio Level. We will use this understanding, combined with our observations, to create an "As is" model of Ocuco's processes at the Team, Program, and Portfolio Level. Some of the questions may seem obvious or redundant, but it is important to get each participant's perspective on each question.

F.2 Confidentiality

With your permission we would like to record the interview, and transcribe your responses so that we can analyse and aggregate the data along with all the other responses we collect from your team.

The recording will not be shared with Ocuco.

The transcripts will be anonymised so that your name will not be associated with any of your responses, and your responses will be difficult to trace back to the source.

Any data relating to the interviews will be securely stored.

Your identity will remain confidential. Your name will not be published and will not be disclosed to anyone outside the Lero team.

Please be assured that you have the right to withdraw from being interviewed at any time during the interview, and do not have to participate in this study if you do not wish to do so.

We anticipate that this interview will take about an hour.

F.3 Permission

This research project has been approved by the Faculty of Science and Engineering Ethics Committee at the University of Limerick, and is being facilitated by Dr Sarah Beecham, Dr. John Noll, MOhammad Abdur Razzak and Prof Ita Richardson from Lero -the Irish Software Research Centre, University of Limerick.

F.4 Demographics

1. Time at Ocuco.
2. Time on current project.
3. Current position.
4. Current location.
5. Previous position & company.
6. Total development experience.
7. Total domain experience.
8. Education and formal qualifications.
9. Gender.
10. Nationality.

F.5 Team level

F.5.1 Project

1. How would you describe your current project?
2. How would you describe your project's current domain?
3. What is your role?
4. Have you met any of your remote colleagues?
5. Does geographic separation hinder the project?
6. Why do you think Ocuco is employing distributed development for this project?
7. Have you had any training in Agile methods?
8. Have you had any training in distributed development?

F.5.2 Process

1. Describe your dev process.
    - How do devlopers test changes?
    - How are builds created for QA?
    - Is build machine a bottleneck?
    - How does "outside of sprint" work?
    - Are there separate QA tasks?
    - Are spikes time-boxed? How is effort accounted for?
    - Is sprint too short?
    - Does Dan ever make estimates?
2. Does the application domain restrict the process in any way?
3. How would you describe the previous process (before Scrum)?
4. What advantages does Scrum offer over that process?
5. What advantages did the previous process have over Scrum?
6. What is working well with Scrum?
7. What is not working well with Scrum?
8. What obstacles exist that prevent Scrum from working well?
9. What is the best aspect of Scrum?
10. What is the worst aspect of Scrum?
11. If there was one thing you could change, what would it be?

For the next two questions, rate your motivation on the following scale:
Very low, Somewhat low, Neither low nor high, Somewhat high, Very high

1. How would you rate your motivation now?
2. How would you rate your motivation prior to introduction of Scrum?

F.6 Project/Program level

F.6.1 Project

1. How would you describe your current project?
2. How would you describe your project's current domain?
3. What is your role?
4. Have you met any of your remote colleagues?
5. Does geographic separation hinder the project?
6. Why is Ocuco employing distributed development?
7. Have you had any training in SAFe or Agile methods?
8. Have you had any training in distributed development?

F.6.2 Process

1. Describe your development process from the PMO perspective.
    - How do you develop the (product) "vision?"
    - How do you develop the product "roadmap?"
        - What do your program metrics comprise?
        - How do you specify your milestones?
        - How do you specify releases?
        - Do you develop a program Kanban? If so, how?
    - (How) do you develop a program backlog?
        - How do you prioritise the backlog? Do you use "Weighted Shortest Job First?"
        - Do you use a "features and benefits matrix?"

  - Do you use "enabler features?" (Features that don't provide direct user-value but rather enable the creation of user-valuable features. Example: test scaffolding to enable Test-First Design & development.)
- (How) do you do Program Increment (PI) planning?
- (How) do you organize your Agile Release Train (ART)? (If they even have ART)
  - Do you use an "architectural runway?"
- Other questions:
  - Do you perform system demos (of work on the ART, rather than the team increment)?
  - Do you conduct an "inspect and adapt workshop?"
  - Can you release any time?
  - How do you develop an "effective solution?"

2. Does your application domain restrict the process in any way?
3. How would you describe your previous process (before SAFe), at the Program Level?
4. What advantages does SAFe offer over that process? *(Omit if SAFe not being applied at Program Level.)*
5. What advantages did the previous process have over SAFe? *(Omit if SAFe not being applied at Program Level.)*
6. What is working well with SAFe from Program perspective?
7. What is not working well with SAFe from Program perspective?
8. What obstacles exist that prevent SAFe from working well from Program perspective?
9. What is the best aspect of SAFe from Program perspective?

10. What is the worst aspect of SAFe from Program perspective?
11. If there was one thing you could change, what would it be?

For the next two questions, rate your motivation on the following scale:
Very low, Somewhat low, Neither low nor high, Somewhat high, Very high

1. How would you rate your motivation now?
2. How would you rate your motivation prior to introduction of SAFe?

F.7 Portfolio level

F.7.1 Project

1. How would you describe your current project?
2. How would you describe your project's current domain?
3. What is your role?
4. Have you met any of your remote colleagues?
5. Does geographic separation hinder the project?
6. Why is Ocuco employing distributed development?
7. Have you had any training in SAFe or Agile methods?
8. Have you had any training in distributed development?

F.7.2 Process–Strategics

Portfolio Manager How do you determine strategic themes?

- Do you drive growth through new product revenues?
- Do you reduce investment in customer project?
- Do you lead market through innovation?

- Do you maintain share in customer segment?
- Do you drive long term growth through enterprise investments?
- Do you leverage common services across segments?
- How do you formulate solution portfolio strategy? (Do you get any input from anyone?)

Portfolio Manager How do strategic themes drive creation of portfolio backlogs and epics?

- Do you define portfolio hierarchy?

Portfolio Manager How do you allocate budget to each strategic theme to ensure balanced delivery against organizational priorities?

- Epic pipeline status
- Projected R&D spend by innovation types
- Epic risk vs. reward
- Risk-adjusted ROI by strategic theme

Portfolio Manager How do you prioritize features and epics within each strategic theme?

- Do you use any scoring model to prioritize epics?

Portfolio Manager How do you manage and adjust strategic themes in a sustainable fashion throughout the budgeting cycle?

Portfolio Manager How do you interlock strategic themes with other operational processes across the enterprise?

Portfolio Manager How do you manage or allocate funding for distributed team or newly acquired company in different country?

Epic Owner How do you develop portfolio Kanban?

Epic Owner How do you develop portfolio Backlogs? Whom do you collaborate in preparation of portfolio backlogs?

Epic Owner How do you help product management team to split the epic into features and prioritize them in the backlogs?

Epic Owner How do you participate in PI planning, System Demo, and Solution Demo?

Epic Owner How do you help Agile Teams during research spikes?

Enterprise Architect As an Enterprise Architect what is your primary reponsibility?

Enterprise Architect What are the aspects do you take consideration while developing enterprise architectural strategy?

- Who decide about new technology?
- Do decide about development and deployment infrastructure strategy?
- Do you involve in interprogram collaboration such as Community of Practice (CoP)?
- Do you take decision about the implementation strategy?

F. INTERVIEW PROTOCOL

Appendix G

Ocuco's As-is Models

G.1 Team Level

Team Backlog Grooming As-is

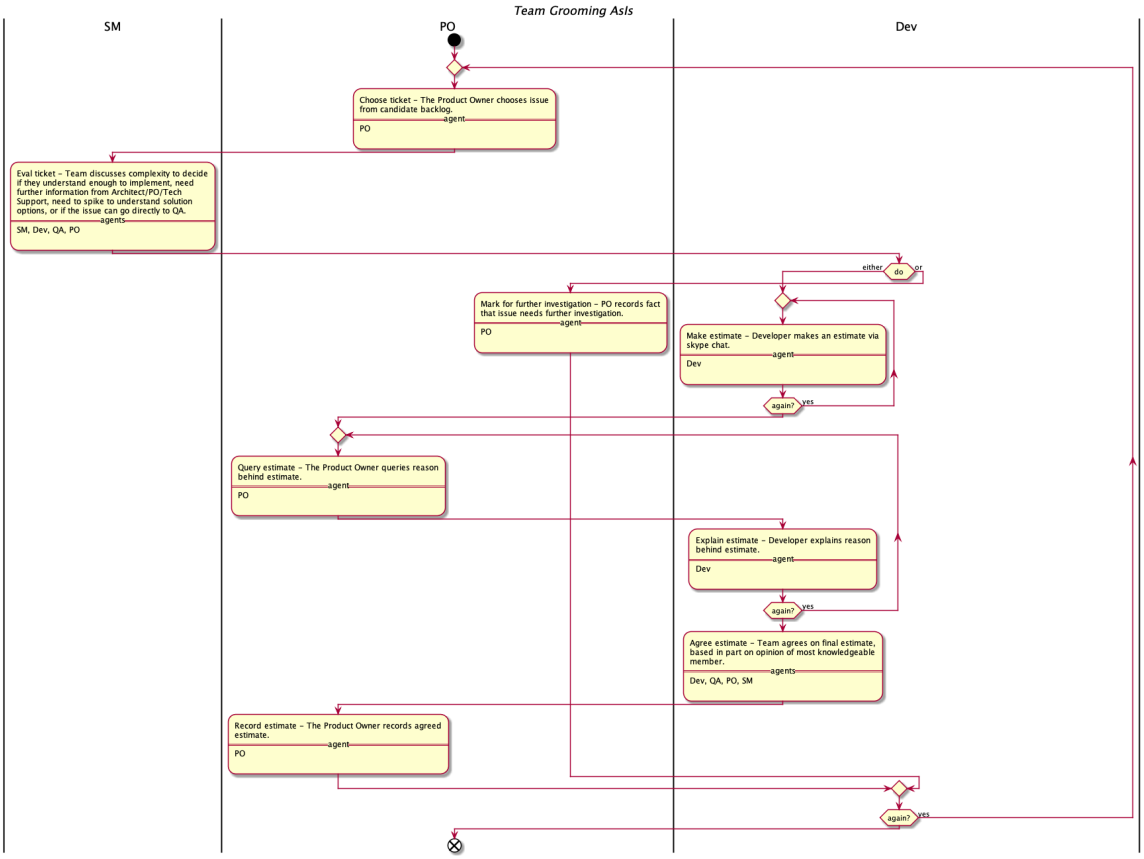

Figure G.1: Ocuco Ltd. Team Level Backlog Grooming As-is.

Team New Issue As-is

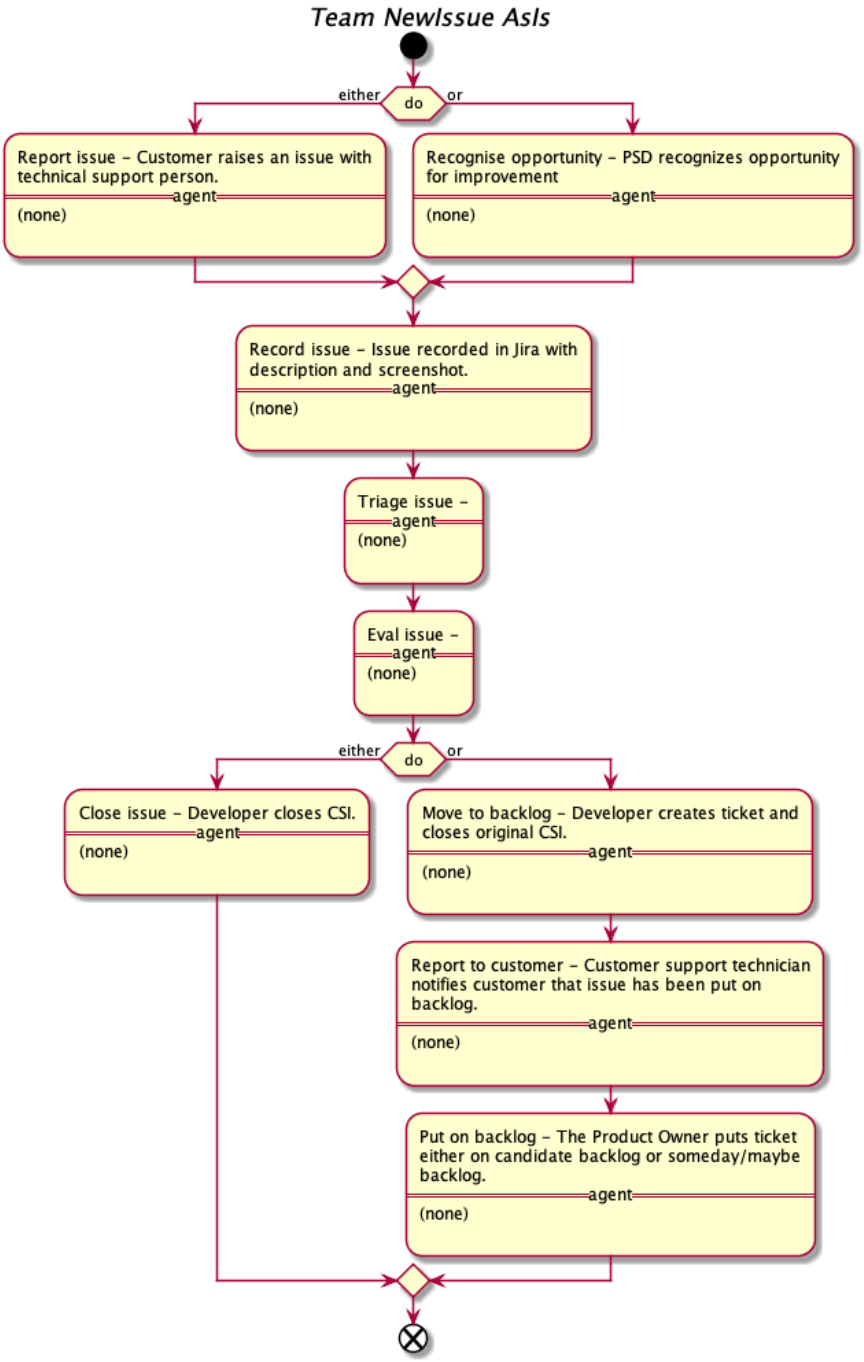

Figure G.2: Ocuco Ltd. Team Level New Issue As-is.

Team Sprint As-is

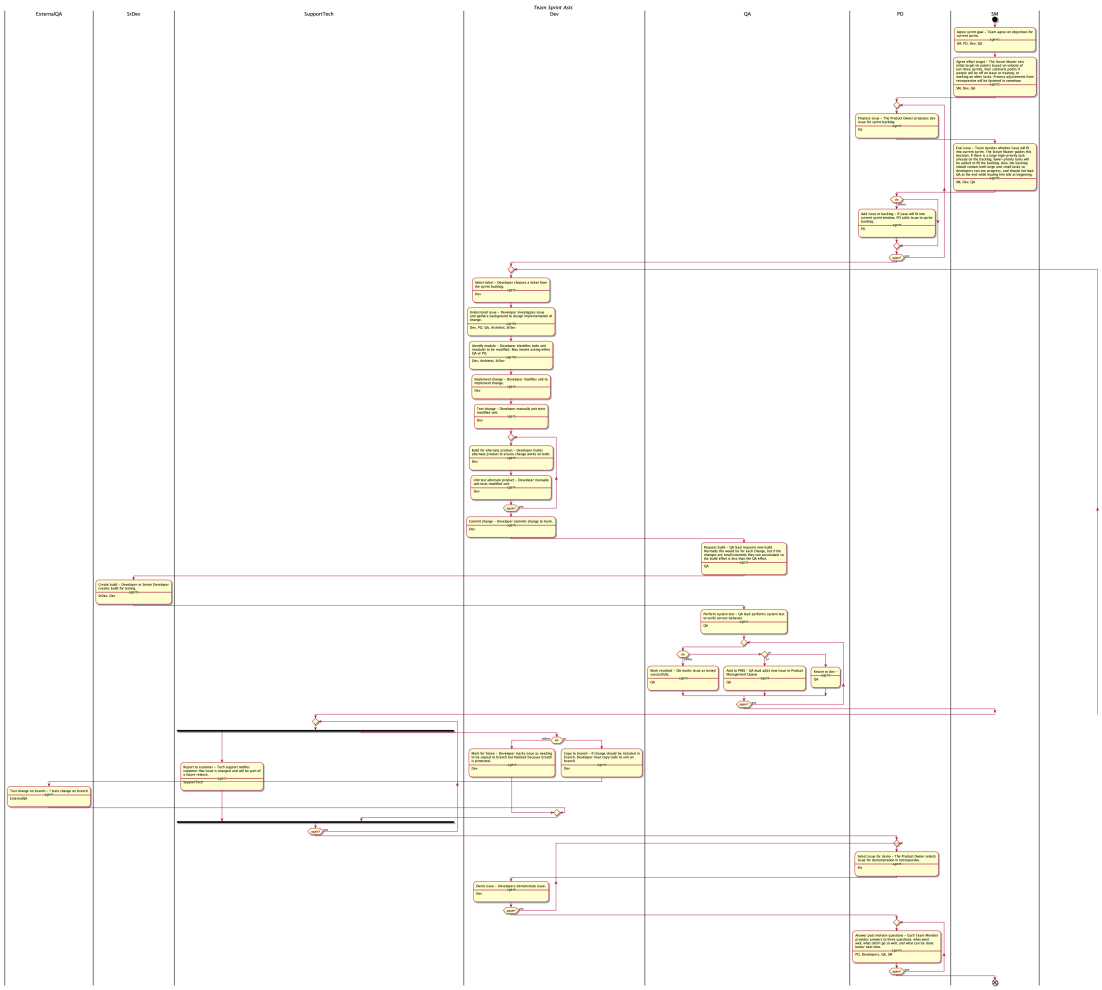

Figure G.3: Ocuco Ltd. Team Level Sprint As-is.

G.2 Program Level

Program As-is

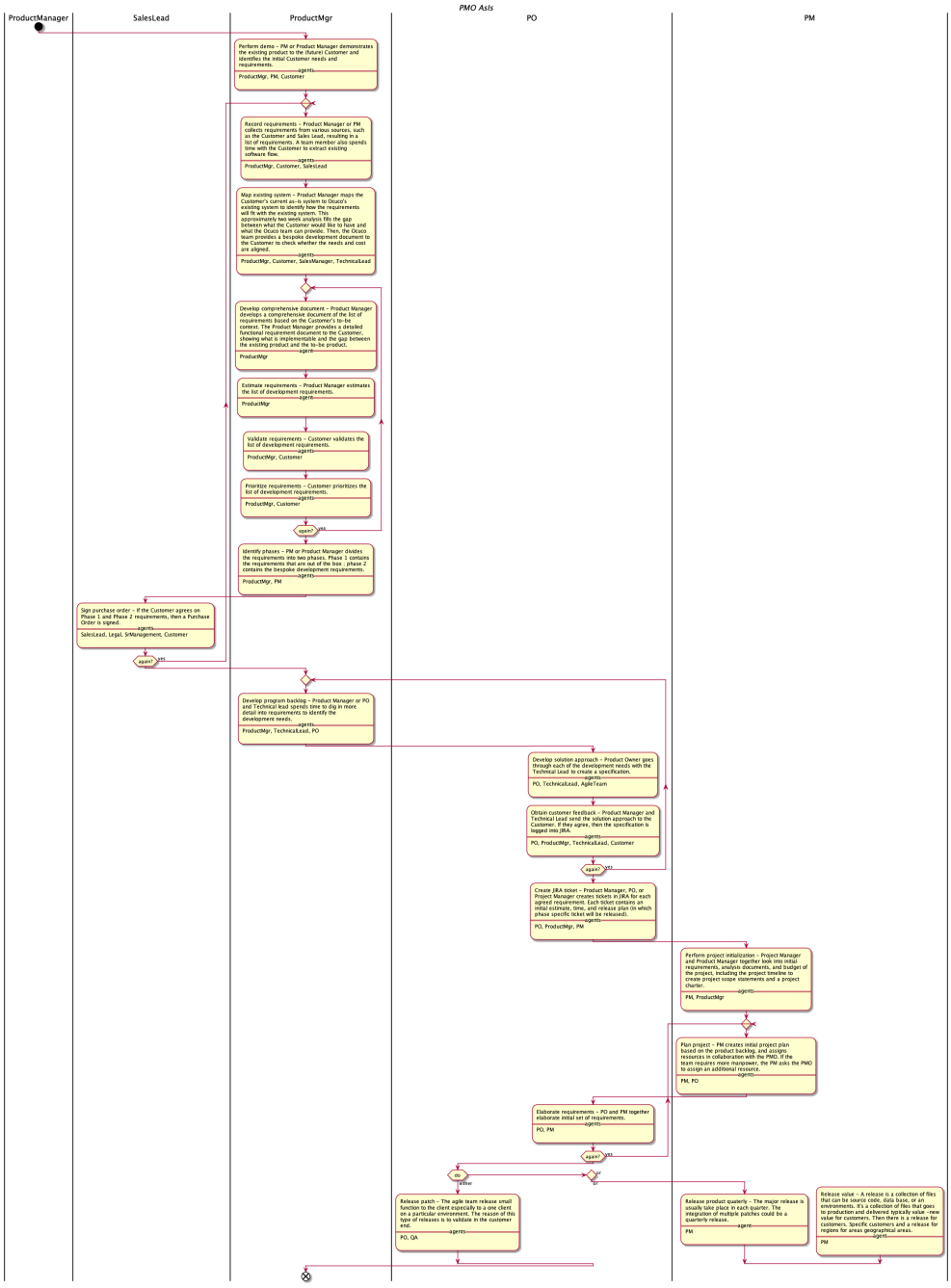

Figure G.4: Ocuco Ltd. Program Level As-is.

Resource Allocation

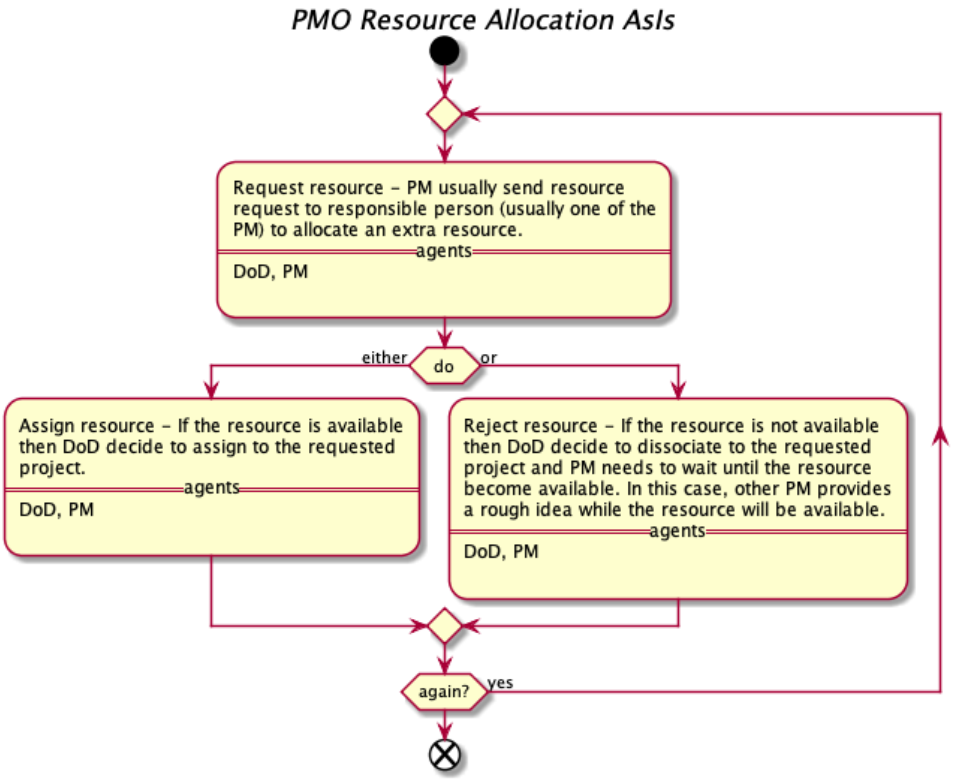

Figure G.5: Ocuco Ltd. Program Level Resource Allocation As-is.

G.3 Portfolio Level

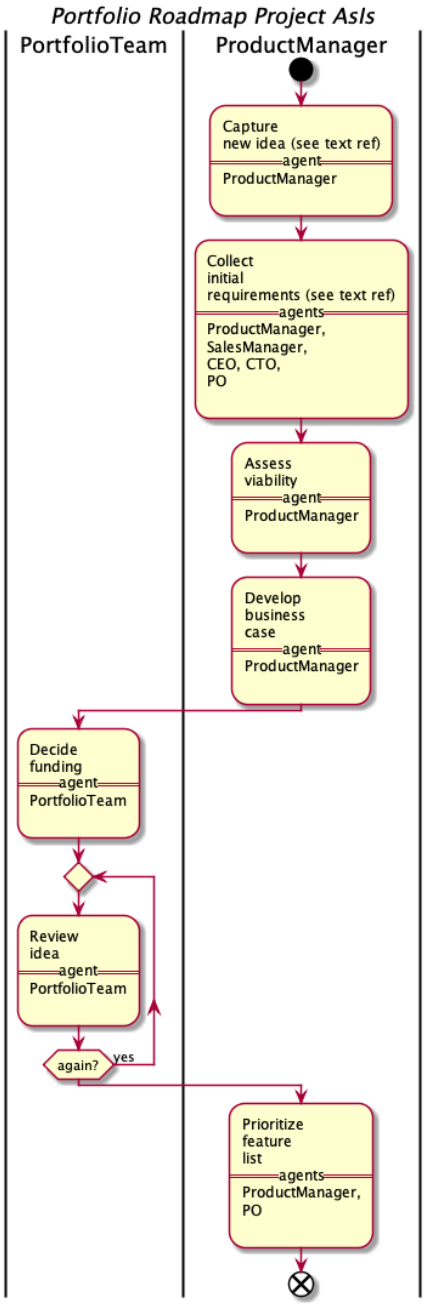

Figure G.6: Ocuco Ltd. Portfolio Level Roadmap Project Project As-is.

Appendix H

Why to perform a practice?–Mapping Ocuco's issues onto GTM recommendations

H.1 Why do it?

ID	Practice	Issue	Quote
B2	Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Task allocation, geographical distance	The challenge is, if the customer (in the US) raised a question obviously the guy in Dublin won't be able to fix it immediately. So, now we are missing one full day for them to look at them again.
B2	Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Geographic separation	Completely, its been quite difficult. It doesn't helped the fact that the customer actually in West Coast of the States.
B2	Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Busy/tight schedule	The difficulty is there, when I[PO] am here on-site I only get couple of hours in morning to deal with Dublin stuffs. If I do not get the things that I need from them even though these two hours I am pretty much isolated for the day. Even I have Technical Lead here is Portland but he is very heavily involved with other things.

B2	Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Missing important updates due to divided time between two roles.	Currently I am working on X project and then Y project. Those two are client facing. Then I also have the Z for USA Product, which is Acuitas integrated in through ActiveEHR. So that's another one that taking as well.
B2	Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Lack of synchronisation in the product line.	We really have to get you know to a common place, it is very difficult with lots of different paid development and changes in the same product is the independence that's difficulties as well.
B2	Base task allocation on the organisational requirement, e.g, if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.	Uncertain requirement adds high risk.	Sometime customers are asking for to sign off and provide estimates and a statement of work on something that we still don't have a full understanding. There could be high risk of adding scope and spending more budget than originally fore-casted.
B3	Retain tasks that require frequent communication between groups within collocated teams.	Geographic distance can create communication problems and breakdown in support even if teams are located in the same timezone and country.	Completely, it's been quite difficult. It doesn't help the fact that the customer is actually in the West Coast
B3	Retain tasks that require frequent communication between groups within collocated teams	Dedicated team members in a different timezone.	Major obstacle would be with the dedicated team and then if you have people cross country.

B3	Retain tasks that require frequent communication between groups within collocated teams	Geographical distribution of the team members create delays.	Inter-locations is a challenge because there are inevitable delays due to the geographical distribution of the team.
C1	Document and define customer base and functions relative to the application being developed	Poor coordination of resources and artifacts means that testing is difficult (e.g. need to have both the file and the sample data).	Even [the customer] needs to have sample data ...and I also need to have sample data to test. In case of South CAL, the classic example would be, having file from [XY]. If we don't have file then there is no way to test it. So, this are the things cause challenges and delay within the sprint.
C1	Document and define customer base and functions relative to the application being developed	Pressure from the customer can drive project managers to give incorrect, rushed estimates about the application that is being developed.	The client would wanna know a plan–they wanna have a plan for everything that we gonna do. Lets say, there are five change requests for example that is in the progress as we are still talking to them about what the requirement is. So, there is no estimate on this. They wanna know exactly when all gonna get done.

C1	Document and define customer base and functions relative to the application being developed	Pressure from the customer can drive project managers to give incorrect, rushed estimates about the application that is being developed.	In Agile world there is no way that I could tell them when they gonna get done until the estimate there, until we started a sprint planning. We don't know this will going to this sprint or not, product owner needs to be prioritize the things and product owner has other five epics on the top list – so thing like that which is very challenging and obviously with a client it's hard because I cannot just tell them we are doing Agile. So, it won't work this way. After come back internally and pressure the guys–can you first track change request.
C1	Document and define customer base and functions relative to the application being developed	Regional differences impose additional requirements, which in turn delays the development process.	They want much more changes and then the main problem they have is localization.
C1	Document and define customer base and functions relative to the application being developed	Lack of local market knowledge causes delays. For example how insurance is handled in the US varies from region to region.	It is big part in US and I think its most difficult part in our software.

H. WHY TO PERFORM A PRACTICE?–MAPPING OCUCO'S ISSUES ONTO GTM RECOMMENDATIONS

C1	Document and define customer base and functions relative to the application being developed	Lack of synchronisation in the product line.	We really have to get you know to a common place, it is very difficult with lots of different paid development and changes in the same product is the independence that's difficulties as well.
C1	Document and define customer base and functions relative to the application being developed	Risk identification is crucial especially when we are dealing with third parties.	Risk identification is also very crucial. And, especially when dealing with third parties is kind of challenging.
C1	Document and define customer base and functions relative to the application being developed	Third party integration.	A partner organisation, part of the company really.
C2	Provide training to ensure that global team has required understanding of the customer base and the business functions to take full advantage of the proximity of the team to the customer base.	Lack of knowledge about insurance/market required training	Is big part in US and I think its most difficult part in our software.
F1	Effective knowledge transfer: Carry out evaluation of training needs to include cultural and linguistic issues.	None	None
F2	Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements.	Lack of training could demotivate team members.	In this particular project Neither low nor high because of the learning curve. And, it's been very difficult to wrap my head around the roles itself.

F2	Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements.	Proper training required to get to know about the role and responsibilities.	Though myself and [X] did bit of training on PO stuff but I don't feel team had enough training on what our role was. So, I think our roles wasn't done best of theirs ability and I don't think I was able to do my role best of my ability because I didn't had the experience that I needed for it.
F2	Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements.	Lack of technical knowledge could cause of wrong requirements.	It is very difficult thing to do, to write technical specification when you are not the actual developer that's going to be develop a pieces of material
G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global management agent	Joint roles, responsibilities, and dependencies, mean that information flow is delayed when working across geographic distance.	So, unfortunately, when I am here on-site with the customer [X] has his/her own number of Epics working remotely, which I can't answer when I am here. And, same for him–when he is here he can't answer the epics I worked on
G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global management agent	Geographic distance can create communication problems and breakdown in support even if teams are located in the same timezone and country.	Completely, it's been quite difficult. It doesn't help the fact that the customer is actually in the West Coast

H. WHY TO PERFORM A PRACTICE?–MAPPING OCUCO'S ISSUES ONTO GTM RECOMMENDATIONS

G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global management agent	Geographic separation can leave the customer without support.	The challenge is, if the customer (in the US) raised a question obviously the guy in Dublin won't be able to fix it immediately. So, now we are missing one full day for them to look at them again.
G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global management agent	Poor streamlining of process leads to delays; processes that worked in a collocated setting don't always work when distributed, as in the case of locking the development unit, and unlocking when developer has finished.	I think there is a problem or probably not. When we do our coding we have to lock our unit that we are working on. When we finish the job we commit the code then release the lock. So, someone else can check that unit about what changes. But, sometimes well its happen regularly in every week that someone forgets to unlock the unit. So, when that happens–previously all developers were in Dublin and that shouldn't be a problem. Now, we have a problem.
G5	Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global management agent	Multiple versions of the build machines can lead to confusion and delays, as developers are uncertain which machine to use and which is available.	It's a disruption for us on a daily basis. We have five active versions in build machine. So, if someone is accessing the build machine right now then I won't be able to know when they will be finish. I don't know when I will get the access.

I1	National culture differences should be identified and communicated to the management and team members. Cultural training can be communicated in the following ways.	None	None
L11	Identify and address local project demands that may pose risk to the global project	Lack of Domain knowledge can create bottlenecks.	Insurance is big part of the software in the US and I think it's the most difficult part in our software.
L3	Identify and address local project demands that may pose risk to the global project	Confusion over what is defined as 'done' risks under utilization of personnel	I think what would help is proper scheduling of the completed tasks –Definition of Done. If I am done, if we have an good idea to finish something by Tuesday then at least QA will be doing something within that day within the sprint.
L3	Identify and address local project demands that may pose risk to the global project	The QA is blocked due to the build machine not being available, and developer is blocked because, they cannot build – the build machine bottleneck is causing a risk to the project.	The Build machine does have an effect in some cases. For example, if I send something to QA and QA will come back and saying, What would be starting point for the test? In some cases, I don't even know at what point the QA is testing it. So, I have to setup the testing environment and I have to make sure that the setting is turned on that environment for her to test it.

L3	Identify and address local project demands that may pose risk to the global project	Poor coordination of resources and artifacts means that testing is difficult (e.g. need to have both the file and the sample data).	Even the customer needs to have sample data as well as me. In the case of our Project, a classic example would be, having a file from Company X. If we don't have data then there is no way to test it. So, these are the things that cause challenges and delays within the sprint.
L3	Identify and address local project demands that may pose risk to the global project	Dependencies and task allocation need to be finely tuned otherwise, delays in development will occur due to temporal distance.	The difficulty is there, when I am here on-site I only get couple of hours in morning to deal with Dublin staff
L3	Identify and address local project demands that may pose risk to the global project	Sharing resources across projects can result in loss of information that can lead to project delays.	When you got 50% time on this project then you do tend to miss of bit both.
L3	Identify and address local project demands that may pose risk to the global project	Over reliance on key personnel can result in bottlenecks when they are over-stretched.	Even I have [X] here in [A location] but he is very heavily involved with other things

L3	Identify and address local project demands that may pose risk to the global project	Pressure from the customer can drive project managers to give incorrect, rushed estimates.	The client would wanna know a plan–they wanna have a plan for everything that we gonna do. Lets say, there are five change requests for example that's in progress as we are still talking to them about what the requirement is. So, there is no estimate on this. They wanna know exactly when all gonna get done.
L3	Identify and address local project demands that may pose risk to the global project	Travel can impact on project, when key personnel are out of communication	The PO has to travel to the customer and in 5 working days if he is away 2 days then everything blocked in our side.
M2	When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences.	Two people holding the same role leads to conflict and governance problems; for example, a team might find it difficult to know what the priorities are, for find themselves working on two epics simultaneously.	We also had lot of conflict in sort of decisions–we had [X] who is the (A role), (Y) is (B role)–it was quite difficult.
M2	When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences.	Small group of people are involved in strategic decision-making.	Strategical decision are taken by a small group of people.

H. WHY TO PERFORM A PRACTICE?–MAPPING OCUCO'S ISSUES ONTO GTM RECOMMENDATIONS

M3	When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences.	Fear of decision making leads to frustration and slipped schedules.	From my own point of view if [X] says ticket takes five days and as a developer if I think it will take 10 days then I probably won't say 10 days because other people may think I am taking more time than I should. The reality is everyone takes a longer time than the initial estimation.
M4	When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account, for example conflict due to fear as well as cultural differences.	Conflicting feedback on requirements from different sources, lead to confusion and frustration. Poor governance meant that it was unclear who was responsible for approving the requirement.	I need to get approval from too many people. So far I have mentioned [X] and [Y] before putting into the backlog. That's not all of it. Some things have to go to [Z] and specially any primary screens of Acuitas. So, I need to get approval from too many people–one person says okay good check with X or no rubbish takes it back. It is too difficult to satisfy that many people.
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Project is not well defined at the early stage of the project.	We are kind of defining stuff as we go. So, I haven't got strong reference point for why we are doing.

O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Missing important updates due to divided time between two roles.	I was working with the (A) team and there is also (B) project as well. So, its like 50 per of my time designated for project (B) and other 50 per for other projects such as (A) and (C).
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Missing important updates due to divided time between two roles.	Currently I am working on X project and then Y project. Those two are client facing. Then I also have the Z for USA Product, which is Acuitas integrated in through ActiveEHR. So that's another one that taking as well.
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Missing important updates due to divided time between two roles.	There'll be days like I only spend minimum time just on our stand-up for example there is any blockers that they need my help. I do not really would be able to go and then look at up of our backlog.
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Missing important updates due to divided time between two roles.	I think in that sense normally I noticed that the internal projects will always kind of suffer, sometimes it showing because the team just there on auto pilots kind of thing and they are moving along but then a kind of sometimes over feel that things would get left behind just because no one there to get back thing are about it or just following those type of things.

H. WHY TO PERFORM A PRACTICE?–MAPPING OCUCO'S ISSUES ONTO GTM RECOMMENDATIONS

O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Customer collaboration is a challenge due to geographical distance.	I think the biggest challenge when its come to the client and things we need to know from them and the follow up we need to do more than little bit more internally you know.
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	There is no visibility of the release planning at the Program level.	I think from program perspective that the plans for release or the very specific release plan that would be the basic benefits. And everybody would be able to understand well and advance what's happening for the next. I don't know six months or a year and what's getting released when and what type of thing.
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Dedicated team members in a different timezone.	Major obstacle would be with the dedicated team and then if you have people cross country.
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	There is no Agile Release Train (ART) at the Program level.	It's difficult because we are still in our transition phase in putting all of this. It's kind of difficult to have with so many different teams.

O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Lack of synchronisation in the product line.	We really have to get you know to a common place, it is very difficult with lots of different paid development and changes in the same product is the independence that's difficulties as well.
O2	Ensure that relevant team members are made aware of how and when they will receive input to product they are working on, and when they need to distribute outputs from these products and when complete work products are required	Third party integration.	Risk identification is also very crucial. And, especially when we are dealing with third parties is kind of challenging.
Q1	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Poor communication between roles leads to rework and frustration, e.g. interface between business owner and Tech lead.	It is very difficult thing to write technical specification when you are not the actual developer who is going to develop a component
Q1	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Likely to repeat the same mistake again, as no time allowed for the review of each sprint. This may be demotivating and it is a loss of opportunity to share knowledge and lessons learned.	We haven't really been keeping the Sprint Review up-to-date as we should do at the end of the sprint

H. WHY TO PERFORM A PRACTICE?–MAPPING OCUCO'S ISSUES ONTO GTM RECOMMENDATIONS

Q2	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Lack of agreement across team, lead to poor estimation, and slipped schedules.	We kind of overcommitted the first time. We overcommitted to a lot of stuff, work to be done within the sprint which I think was too much. It is a result of bad or wrong estimation. So, whatever we committed to do in two weeks in reality the job took four weeks.
Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Ad hoc processes for new work lead to confusion, with the result that the project is not well defined.	Although I have experienced people to work with, this work doesn't exist in Acuitas yet. So in that respect, I find it difficult. We kind of define stuff as we go along so, I haven't got a strong reference point for why we are doing [the work in the project]
Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Overcommitment leads to late of delivery of promised product.	For the first few sprints we completely overcommitted to a lot of stuff which we just couldn't deliver. So, we are trying to fit in with the velocity that is based on the size of the team.
Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Internal projects are always kind of suffer due to multiple responsibilities.	I think in that sense normally I noticed that the internal projects will always kind of suffer, sometimes it showing because the team just there on auto pilots kind of thing and they are moving along but then a kind of sometimes over feel that things would get left behind just because no one there to get back thing are about it or just following those type of things.

Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Lack of synchronisation in the product line.	We really have to get you know to a common place, it is very difficult with lots of different paid development and changes in the same product is the independence that's difficulties as well.
Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Risk identification is crucial especially when dealing with third parties.	Risk identification is also very crucial. And, especially when we are dealing with third parties is kind of challenging.
Q3	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Requirements from customer are misunderstood by the development team or recorded the wrong request.	I remember a few weeks ago we were with a customer from X country that's asked for a particular improvement in a particular area and the whole issue had gotten completely misunderstood by development. Then we had been sitting in the backlog for a very long time and me and X set up a call with the customer to try and clear up exactly what they want cause a lot of the time there is this Chinese whispers thing going on with information getting. People misunderstanding things, recording the wrong request and so on.

H. WHY TO PERFORM A PRACTICE?–MAPPING OCUCO'S ISSUES ONTO GTM RECOMMENDATIONS

Q4	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Developer doesn't follow or understand objectives, and as a result the project suffers lower productivity. E.g. Does not take responsibility to take on another due to lack of ownership;	I personally do not go to the engineering backlog but I know some other developers do, X finishes his task earlier than I do, and then gets items from engineering backlog to work on.
Q4	Project goals and objectives communicated, understood and agreed across all team members regardless of location	Lack of authority, autonomy, and over bureaucracy, lead to re-work and delays in work flow.	Yah, when I say rework its not re-edit. Its kind of starting from scratch. So, that is one thing I need to change because I don't think its follows agile to require approval from so many different points.
R1	Define product ownership boundaries through partitioning of work across GSE teams	Team members do not take responsibility to take on another task due to lack of ownership	I personally do not go to the engineering backlog but I know some other developers such as [X]–who finish task earlier and they do get items on the backlog and work on it. So, me personally answer your question, No–I haven't done that.
R1	Define product ownership boundaries through partitioning of work across GSE teams	Joint roles, responsibilities, and dependencies, means that information flows is delayed when working across geographic distance.	So, unfortunately, when I am here on-site [X] has own number of epics, which I can't answer when I am here. And, same for him–when he is here he can't answer the epics I worked on.

R1	Define product ownership boundaries through partitioning of work across GSE teams	Project is not well defined at the early stage of the project.	So, in that respect I find it difficult. We are kind of defining stuff as we go.
S6	Process should take into account the relevant structures and procedures from all sites	Poor streamlining of process, leads to QA carry over to next Sprint	QA side was not really had the time to finish the task within the sprint–it poses carry over. Which shouldn't be the case where proper estimation including the QA time, it should stick within the sprint. It should not go over or go beyond.
S6	Process should take into account the relevant structures and procedures from all sites	Pressure to QA at the end of sprint and it's create bottleneck to the QA end	On day 8th QA receive multiple tickets from multiple developers and there is only one QA.
S6	Process should take into account the relevant structures and procedures from all sites	Pressure to QA at the end of sprint and it's create bottleneck to the QA end	You could expect there will be a bottleneck during that time. Because, if it was schedule properly, job and tasks are completed. In reality, what happen, all developers finish their tasks around the same time then it's required a lot of responsibilities and time for QA to finish.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Moving from Plan driven to Agile can create problems	So, we worked on Waterfall fashion in past and I think this is difficult for people to move from the Waterfall way to Scrum way.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Moving from Plan driven to Agile can create problems	It looks like we have Waterfall. But, we don't have waterfall and we are trying to be Agile.

H. WHY TO PERFORM A PRACTICE?–MAPPING OCUCO'S ISSUES ONTO GTM RECOMMENDATIONS

T4	Establish procedures to coordinate implementation of contingencies when and if required	Moving from Plan driven to Agile can create problems	We are build and doing a lot of things but its not fully agile.
T4	Establish procedures to coordinate implementation of contingencies when and if required	The QA is blocked due to the build machine not being available, and developer is blocked because, they cannot build – the build machine bottleneck is causing a risk to the project.	If I send something to QA and QA will come back and saying, What would be starting? In some cases, I don't even know where the QA stuffs are testing it. So, I have to setup the testing environment and I have to make sure that the setting is turned on that environment for her to test it.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Lack of coordination with QA	QA side was not really had the time to finish the task within the sprint–it poses carry over. Which shouldn't be the case where proper estimation including the QA times. It should stick within the sprint. It should not go over or go beyond.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Lack of coordination with QA	Due to dependencies we carried over the old sprint tickets to the next sprint. So, if I couldn't finish any ticket in this sprint then we move it to the next sprint along with new one. We have the new one and also we have to make sure we finished the old one.

T4	Establish procedures to coordinate implementation of contingencies when and if required	Lack of coordination with QA	We still try to complete the QA within the sprint. Because this is one of the big issues we are having certainly earlier of the sprint. Most of the tickets delivered to the QA second or last day of the sprint then QA doesn't have enough time to close them. So, that was the problem we are having.
T4	Establish procedures to coordinate implementation of contingencies when and if required	No time allowed for the review of each sprint.	We haven't really been keeping it up-to-date, as we should do at the end of the sprint.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Geographic distance can create coordination problems and breakdown in support	The challenge is, if they [Customer] raised a question on it obviously the guy in Dublin won't be able to fixed it. So, now we are missing one full day for them to look at them again.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Geographic distance can create coordination problems and breakdown in support	When we do our coding we have to lock our unit that we are working on. When we finish the job we commit the code then release the lock. So, someone else can check that unit about what changes. But, sometimes well its happen regularly in every week that someone forget to unlock the unit.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Geographic distance can create coordination problems and breakdown in support	Due to time zone barrier things could be block. Sometimes I get the answer at 9pm.

T4	Establish procedures to coordinate implementation of contingencies when and if required	Rework due to misunderstanding	Possibly things would be late if you do not have face-to-face communication we can interpret the things in different way.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Blocked due to PO in Customer site/PO handover	The PO has to travel to customer and in 5 working days if he away 2 days then everything blocked in our side.
T4	Establish procedures to coordinate implementation of contingencies when and if required	Multiple version of Build machine can create coordination problems	It's a disruption for us on a daily basis. We have five active versions in build machine. So, if someone is accessing the build machine right now then I won't be able to know when he or she will be finish. I don't know when I will get the access.
V	Collaboratively plan, develop and validate systems and changes	There is no visibility of the release planning at the Program level.	I think from program perspective that the plans for release or the very specific release plan that would be the basic benefits. And everybody would be able to understand well and advance what's happening for the next. I don't know six months or a year and what's getting released when and what type of thing.
V	Collaboratively plan, develop and validate systems and changes	Lack of synchronisation in the product line	We really have to get you know to a common place, it is very difficult with lots of different paid development and changes in the same product is the independence that's difficulties as well.

V	Collaboratively plan, develop and validate systems and changes	Country specific regulations	When we tried to observe HIPPA and because although that's only on the USA at the moment, we might observe everywhere.
V	Collaboratively plan, develop and validate systems and changes	Country specific regulations	Data protection, so you know you have to consider what user see on the screen and you know for the people might be able to see it. And you have to consider how for example the text reminder we sent. You know they gotta be secure, nobody can hacking or steal them.
V	Collaboratively plan, develop and validate systems and changes	Geographical distance hinder regulation especially functionality	So the integration required in the in X is complex and it's not only the standard Acuitas patient work journey. But is the patient journey customize for X in which there is insurance and there are membership you know everything is based on insurance and membership. Insurances plan and so it is a lot more complex than the standard Acuitas. And that process was painful because of sometime the geographical distance.
V	Collaboratively plan, develop and validate systems and changes	Country specific regulations	Plus we have 3 or 4 micro project to manage because there are another thing specific for France for your understanding is the many regulation regarding the healthcare system.

V	Collaboratively plan, develop and validate systems and changes	Regulation has an impact on technical requirements	If I talk about data team we have GDPR regulations coming in, general data protection from the EU which means we need to start encrypting databases. That's a regulation or legislative regulations coming in means we must do that by a period of time.

Appendix I

Process Roadmap

I.1 Roadmap B2

Base task allocation on the organizational requirement, e.g., if proximity to market is reason development team is located in a particular country, then customer-related tasks should be allocated to that team.

I.1.1 GTM Context

Fig. I.1 provides GTM context of the recommendation B2.

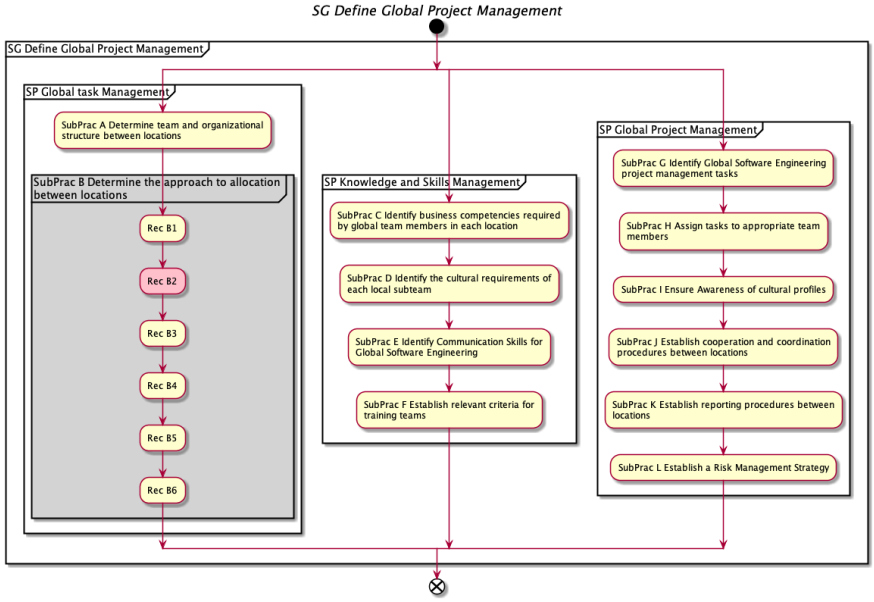

Figure I.1: B2 GTM context.

I.1.2 Why implement this recommendation?

From observations and interviews:

- task allocation, geographical distance
- Geographic separation
- Busy/tight schedule

- Missing important updates due to divided time between two roles
- Lack of synchronisation in the product line
- Uncertain requirement adds high risk

I.1.3 Decompose stories into tasks (S4T15)

I.1.3.1 Purpose

The purpose of this practice is to break each story into tasks, then estimate and identify dependencies of a specific task.

I.1.3.2 Background

Prior to the planning and iteration meeting, the Product Owner will have prepared some preliminary iteration goals, based on the team's progress in the Program Increment (PI). Typically, the Product Owner starts the meeting by reviewing the proposed iteration goals and the higher-priority stories in the team backlog. During the meeting, the Agile Team discusses implementation options, technical issues, Nonfunctional Requirements (NFRs), and dependencies, then plans the iteration. The Product Owner defines the *what*; the team defines *how* and *how much*.

Throughout the meeting, the team elaborates the acceptance criteria and estimates the effort to complete each story. Based on their velocity, the team then selects the candidate stories. The team breaks each story down into tasks and estimate them in hours to confirm that they have the capacity and skills to complete them. Once confirmed, the team commits to the work and records the iteration backlog in a visible place, such as a storyboard or tooling. This meeting is timeboxed to a maximum of four hours.

I.1.3.3 As-is Context

Existing As-Is actions do not change.

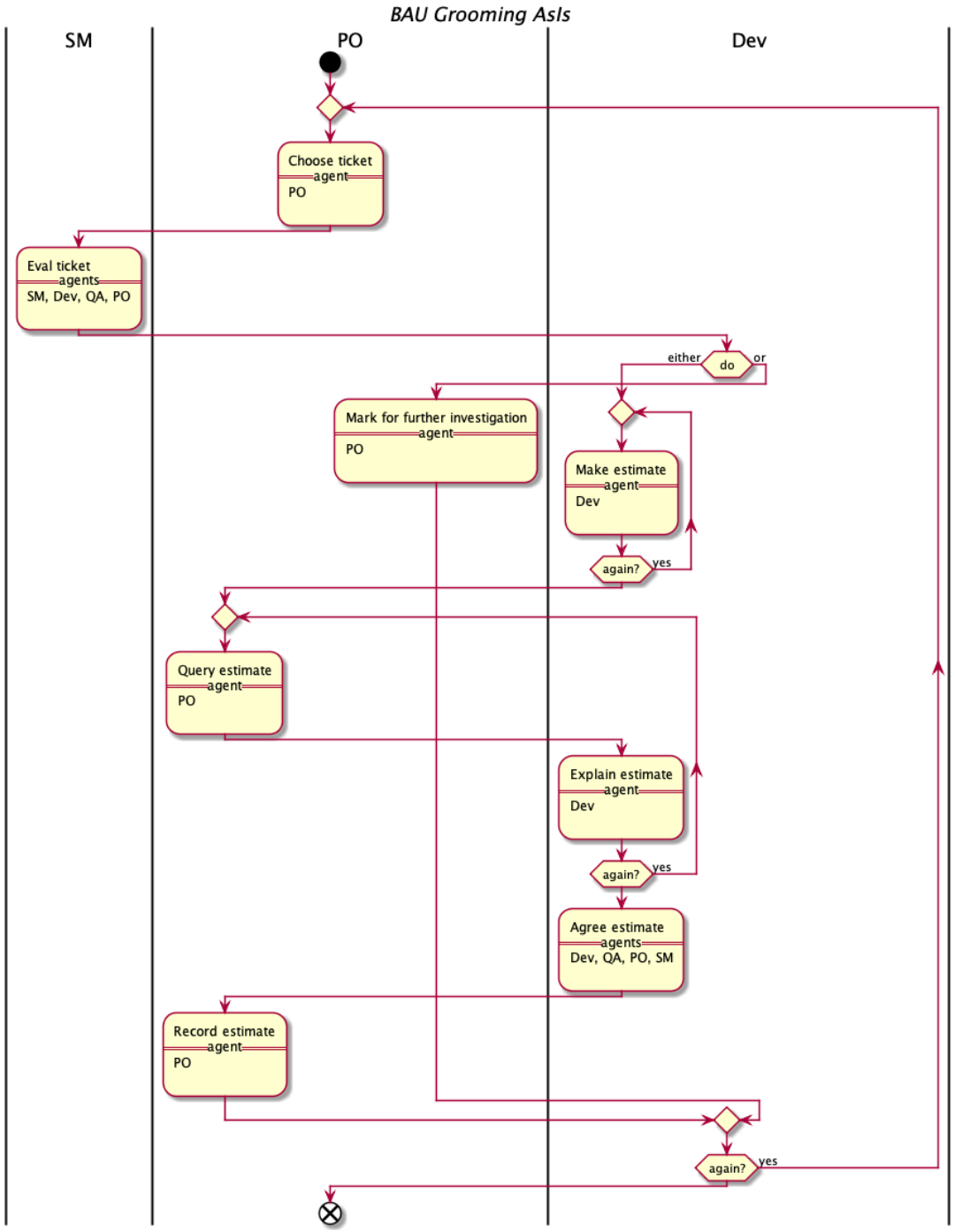

Figure I.2: S4T15 As-is context.

I.1.3.4 To-be Context

Two new actions capture possible decomposition of stories into smaller tasks.

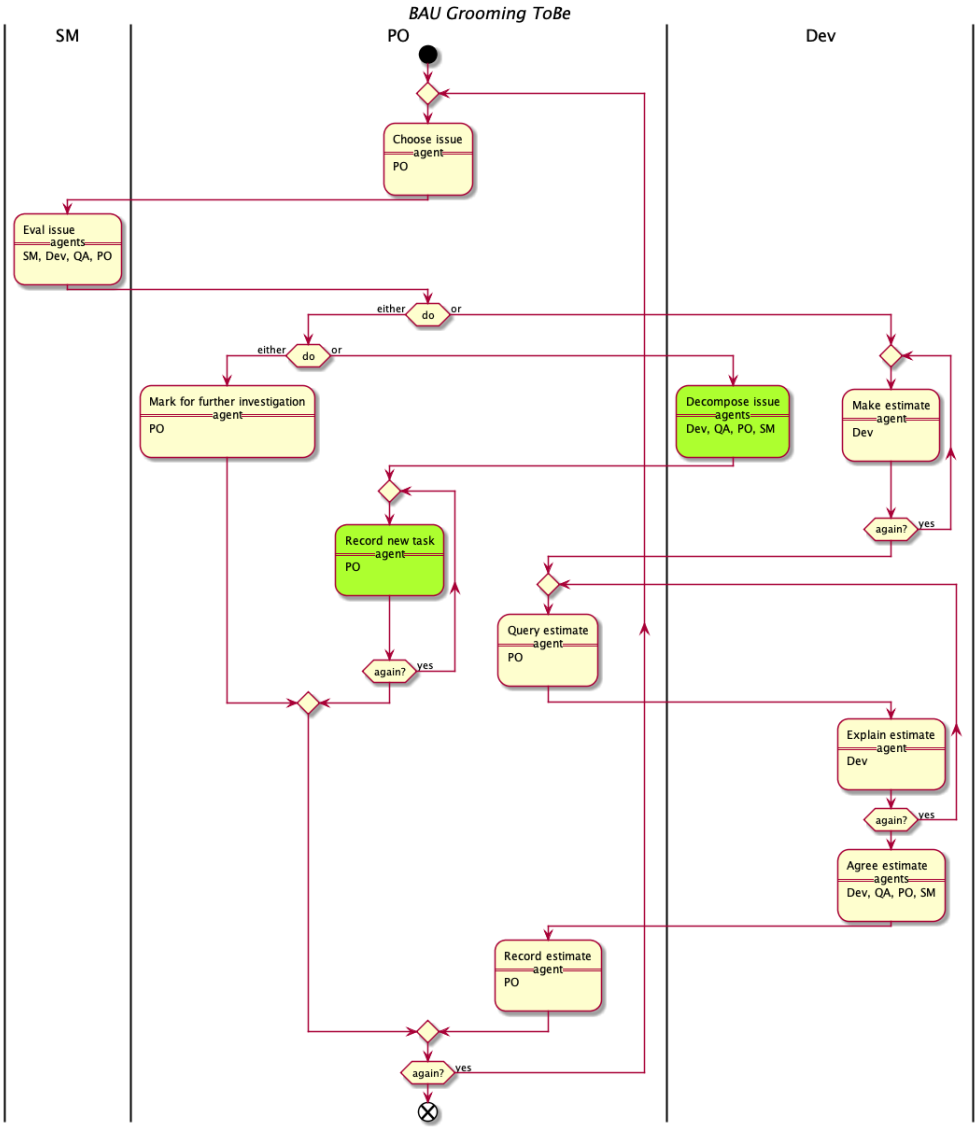

Figure I.3: S4T15 To-be context.

I.1.3.5 Stakeholders

PO, SM, Dev, QA.

I.1.3.6 Entry and Exit Criteria

- Entry Criteria: On candidate backlog ticket
- Exit Criteria: Estimated ticket, Needs further investigation ticket, On product backlog ticket

I.1.3.7 Implementation details

Table I.1: Implementation details for practice S4T15 – Decompose stories into tasks.

Step	Description	Role(s)
Choose issue	The Product Owner chooses issue from candidate backlog.	PO
Eval issue	Team discusses complexity to decide if they understand enough to implement, need further information from Architect/PO/Tech Support, need to spike to understand solution options, or if the issue can go directly to QA.	SM, Dev, QA, PO
Mark for further investigation	PO records fact that issue needs further investigation.	PO
Decompose issue	Team breaks issue down into smaller tasks.	Dev, QA, PO, SM
Record new task	PO records new tasks on product backlog.	PO
Make estimate	Developer makes an estimate via skype chat.	Dev
Query estimate	The Product Owner queries reason behind estimate.	PO
Explain estimate	Developer explains reason behind estimate.	Dev
Agree estimate	Team agrees on final estimate, based in part on opinion of most knowledgeable member.	Dev, QA, PO, SM
Record estimate	The Product Owner records agreed estimate.	PO

I.1.4 Prepare preliminary iteration goals (S4T266)

I.1.4.1 Purpose

Iteration goals derive from PI objectives, and so are aligned with goals of other teams. This reduces the possibility of conflicts due to teams working at cross-purposes. According to SAFe, "Iteration Goals are a high-level summary of the

business and technical goals that the Agile Team agrees to accomplish in an Iteration. They are vital to coordinating an Agile Release Train (ART) as a self-organizing, self-managing team of teams.Iteration goals provide the following benefits:

- Align the team members and the Product Owner to the mission
- Align the people to the Program Increment (PI) Objectives
- Provide context for understanding and addressing cross-team dependencies
- Whether the teams apply Scrum or Kanban, iteration goals give program stakeholders, management, and Agile teams a shared language for maintaining alignment, managing dependencies, and making necessary adjustments during the execution of the program increment (Leffingwell, 2015).

I.1.4.2 Background

According to SAFe, iteration goals support "transparency, alignment, and program execution (Leffingwell, 2015)." It is not enough to commit to complete a set of stories in an iteration. Rather, it is necessary to continually review the business value of each iteration, and then communicate this value to stakeholders such as business owners, management, etc. Iteration goals can reflect:

- Features, feature slices, or feature aspects, such as research and necessary infrastructure Business or technical;
- Milestones Architectural, infrastructure, exploration and compliance activities;
- Iteration goals are achieved by completing backlog items, even though it may not be necessary to finish every story to meet the goals. In other words, the goals for the iteration override any particular story. On occasion, it may even be necessary to add new user stories to achieve the iteration's goals (Leffingwell, 2015)."

Taking an ART view, iteration goals will help teams to focus on the bigger picture and plan for each iteration. This also feeds into the associated System Demo (for which we also have a 'To Be' practice defined).

I.1.4.3 As-is Context

The 'As is' actions are also performed in the 'To be' process; the 'To be' process also has additional tasks.

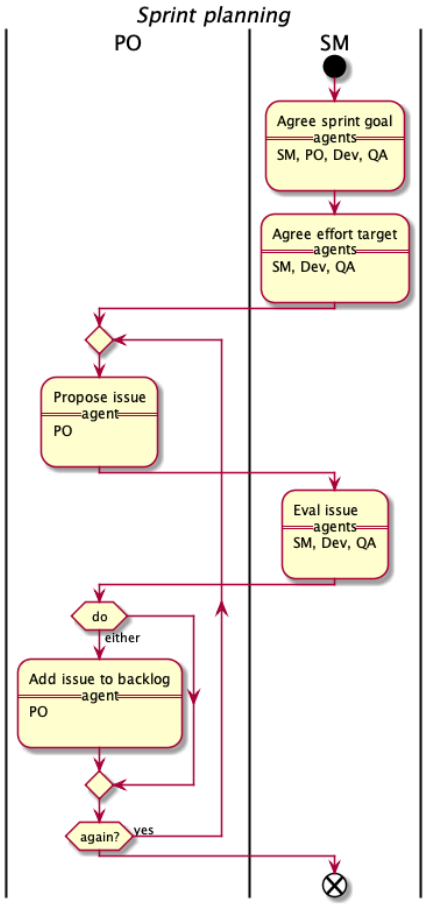

Figure I.4: S4T266 As-is context.

I.1.4.4 To-be Context

New actions are shown in green.

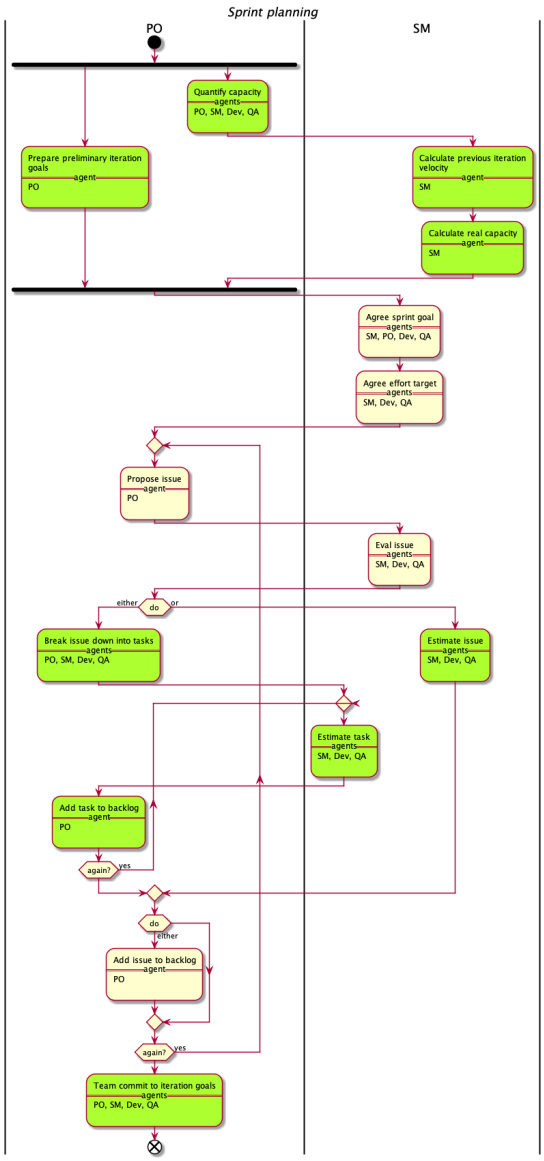

Figure I.5: S4T266 To-be context.

I.1.4.5 Stakeholders

PO, SM, Developer, and QA.

I.1.4.6 Entry and Exit Criteria

- Entry Criteria: PI Objectives, Booked holidays, Team availability

- Exit Criteria: Improvement story

I.1.4.7 Implementation details

Table I.2: Implementation details for practice S4T266 – Prepare preliminary iteration goals.

Step	Description	Role(s)
Prepare preliminary iteration goals	The Product Owner prepares preliminary iteration goals.	PO
Quantify capacity	The Team quantifies (ideal) capacity.	PO, SM, Dev, QA
Calculate previous iteration velocity	The Scrum Master calculates previous iteration velocity.	SM
Calculate real capacity	The Scrum Master calculates real capacity for upcoming iteration using ideal capacity and velocity from previous iteration.	SM
Agree sprint goal	Team agree on objectives for current sprint.	SM, PO, Dev, QA
Agree effort target	The Scrum Master sets initial target (in points) based on velocity of last three sprints, then subtracts points if people will be off on leave or training, or working on other tasks. Process adjustments from retrospective will be factored in somehow.	SM, Dev, QA
Propose issue	The Product Owner proposes development issue for sprint backlog.	PO
Eval issue	The Team decides whether issue will fit into current sprint. The Scrum Master guides this decision. If there is a large high-priority task already on the backlog, lower-priority tasks will be added to fill the backlog. Also, the backlog should contain both large and small tasks so developers can see progress, and should not load QA at the end while leaving him idle at beginning.	SM, Dev, QA
Break issue down into tasks	Team breaks story down into smaller tasks.	PO, SM, Dev, QA
Estimate task	Team estimates task resulting from breakdown of story into tasks.	SM, Dev, QA

Continued on next page

Table I.2 – *Continued from previous page*

Step	Description	Role(s)
Add task to backlog	If issue now fits into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Estimate issue	Team estimates stories recorded in the team backlog.	SM, Dev, QA
Add issue to backlog	If issue will fit into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Team commit to iteration goals	Team commits to iteration goals considering the capacity and estimates.	PO, SM, Dev, QA

I.1.5 Perform continuous integration (S4T272)

I.1.5.1 Purpose

Conflict between teams can arise when one team checks-in code to fix or enhance a specific product version, that breaks the build of other targets. Continuous integration identifies such issues as soon as code is committed, by building for all targets: any target build that fails is identified immediately.

In SAFe, the continuous integration practice includes not only the technical infrastructure and process to achieve continuous integration, but also the culture to ensure failed integrations are fixed as soon as possible.

I.1.5.2 Background

Continuous Integration (CI) has two purposes: first, to reduce risk by identifying integration problems as soon as they arise; and second, to enable fast but sustainable development, by making new functionality that might be required by other teams available as soon as it is completed.

I.1.5.3 As-is Context

Actions altered by this practice are shown in purple.

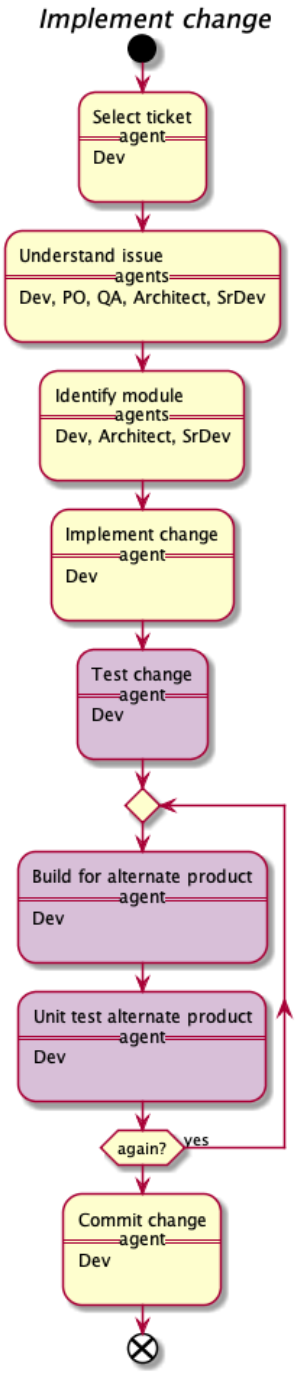

Figure I.6: S4T272 As-is context.

I.1.5.4 To-be Context

New actions are shown in green.

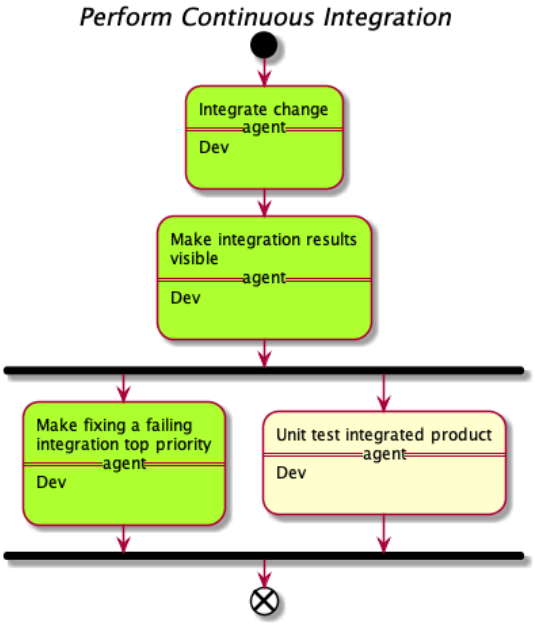

Figure I.7: S4T272 To-be context.

I.1.5.5 Stakeholders

Developer.

I.1.5.6 Entry and Exit Criteria

- Entry Criteria: PI Objective
- Exit Criteria: Improvement story

I.1.5.7 Implementation details

Table I.3: Implementation details for practice S4T272 – Perform continuous integration.

Step	Description	Role(s)
Integrate change	Commit kicks off automated build and continuous integration mechanism to build for all target platforms and configurations.	Dev
Make integration results visible	When the integration process breaks, everybody should know how and why it broke. And when it's fixed, they should know what fixed it, too.	Dev
Make fixing a failing integration top priority	When a build for any target platform fails, the team stops and fixes it immediately. Teams use flashing lights when a build is broken, and highly visible indicators of percentages of the time the system is broken.	Dev
Unit test integrated product	Developer unit-tests modified uni	Dev

I.1.6 Collaborate with large stakeholders (S4VS12)

I.1.6.1 Purpose

The purpose of this practice is to make sure that the decision-making process is decentralized.

I.1.6.2 Background

In SAFe, "the enterprise (Portfolio level) empowers product and solution management with the relevant context, knowledge, and authority to make necessary content decision at the each level of the framework (Leffingwell, 2007, 2015)". The enterprise (Portfolio level) collaborates with the "large stakeholder community to determine the best course of action (Leffingwell, 2007, 2015)."

I.1.6.3 As-is and To-be Context

I have not observed any as is process for this practice. The To-be model is shown in Fig. I.8.

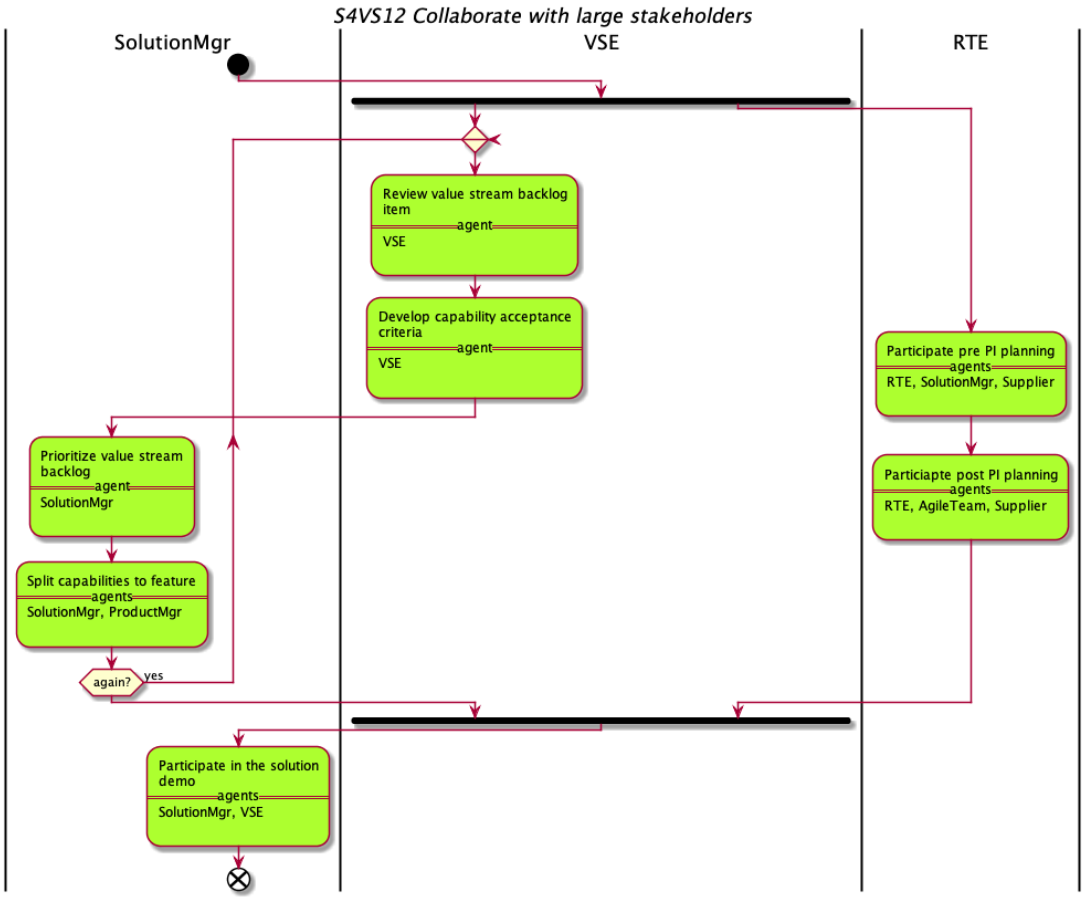

Figure I.8: S4VS12 To-be context.

I.1.6.4 Stakeholders

VSE, Solution Mgr, Product Mgr, RTE, Supplier, Agile Team.

I.1.6.5 Entry and Exit Criteria

- Entry Criteria: Capability, Previous PI Summary, Program Backlog, Roadmap available, Value Stream Backlog available, and Vision available
- Exit Criteria: Feedback

I.1.6.6 Implementation details

Table I.4: Implementation details for practice S4VS12 – Collaborate with large stakeholders.

Step	Description	Role(s)
Review value stream backlog item	At this stage, Value Stream Engineer reviews backlog item definition and updates accordingly.	VSE
Develop capability acceptance criteria	VSE looks into the backlog item definition and develops acceptance criteria.	VSE
Prioritize value stream backlog	Solution Manager uses the WSJF prioritization method for job sequencing.	Solution Mgr
Split capabilities to feature	Solution Manager analyzes ways to split backlog items into smaller chunks of incremental value. If the capability is big, then the product management split that capability into multiple features.	Solution Mgr, Product Mgr
Participate pre PI planning	The pre-planning meeting is used to build the context that allows the ARTs and Suppliers to create their plan.	RTE, Solution Mgr, Supplier
Particiapte post PI planning	The post-planning occurs after the ARTs have run their respective planning sessions, and it is used to synchronize the ARTs and create the overall solution plan and roadmap.	RTE, Agile Team, Supplier
Participate in the solution demo	Solution Manager demonstrates the capabilities to provide an understanding that Value Stream contributed and reviews the contribution of other value streams. Through this process, the solution manager and VSE reviews value stream PI objectives, demonstrate each objective and capability in an end-to-end use case and identify business value completed per objective.	Solution Mgr, VSE

I.2 Roadmap B3

Retain tasks that require frequent communication between groups within collocated teams.

I.2.1 GTM Context

Fig. I.9 provides GTM context of the recommendation B3.

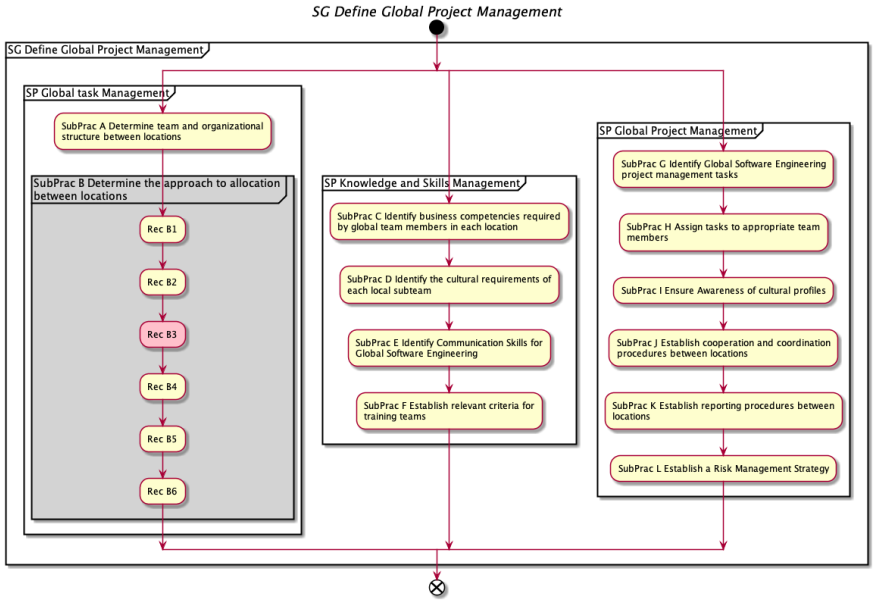

Figure I.9: B3 GTM context.

I.2.2 Why implement this recommendation?

From observations and interviews:

- Geographic distance can create communication problems and breakdown in support even if teams are located in the same timezone and country
- Dedicated team members in a different timezone
- Geographical distribution of the team members create delays

I.2.3 Decompose stories into tasks (S4T15)

I.2.3.1 Purpose

The purpose of this practice is to break each story into tasks, then estimate and identify dependencies of a specific task.

I.2.3.2 Background

Prior to the planning and iteration meeting, the Product Owner will have prepared some preliminary iteration goals, based on the team's progress in the Program Increment (PI). Typically, the Product Owner starts the meeting by reviewing the proposed iteration goals and the higher-priority stories in the team backlog. During the meeting, the Agile Team discusses implementation options, technical issues, Nonfunctional Requirements (NFRs), and dependencies, then plans the iteration. The Product Owner defines the *what*; the team defines *how* and *how much*.

Throughout the meeting, the team elaborates the acceptance criteria and estimates the effort to complete each story. Based on their velocity, the team then selects the candidate stories. The team breaks each story down into tasks and estimate them in hours to confirm that they have the capacity and skills to complete them. Once confirmed, the team commits to the work and records the iteration backlog in a visible place, such as a storyboard or tooling. This meeting is timeboxed to a maximum of four hours.

I.2.3.3 As-is Context

Existing As-Is actions do not change.

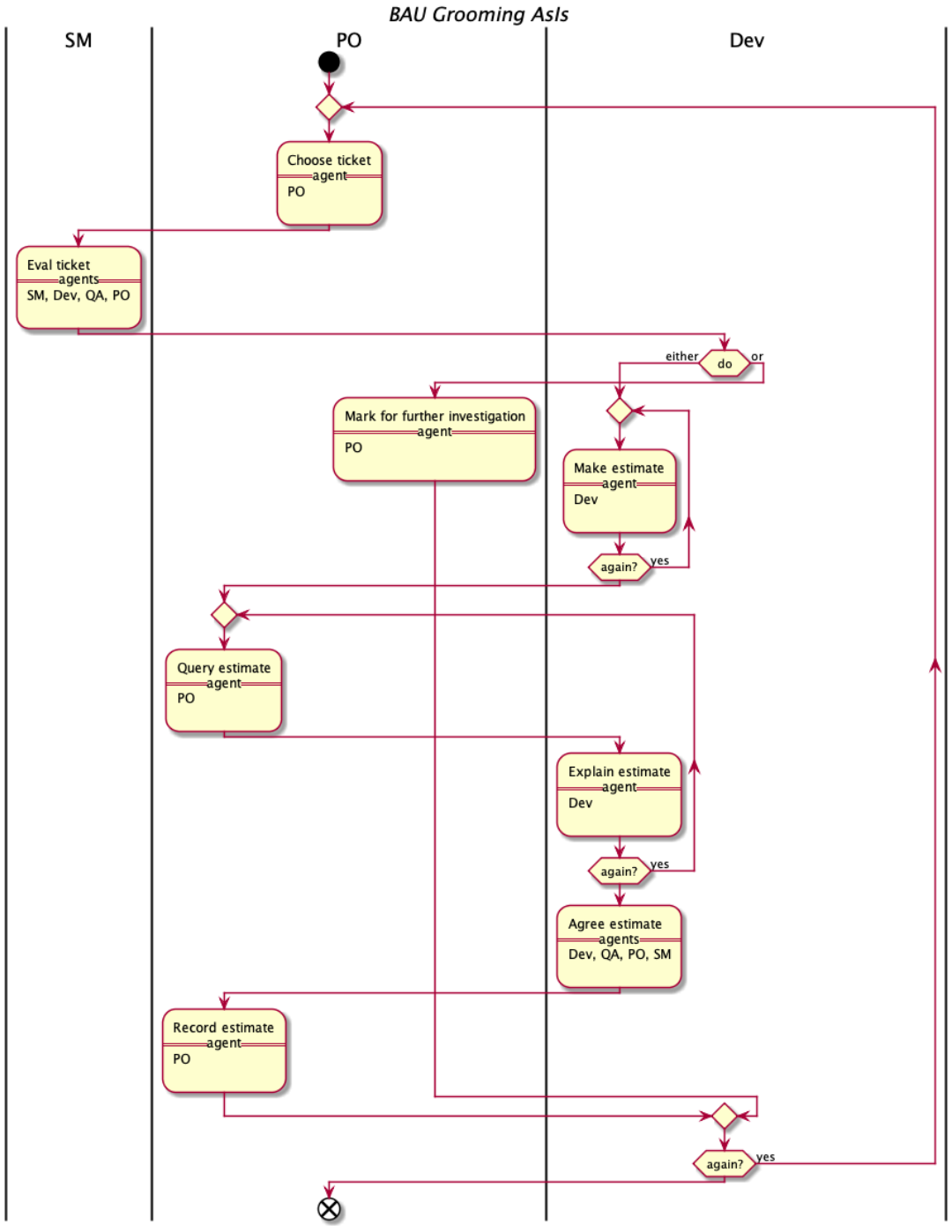

Figure I.10: S4T15 As-is context.

I.2.3.4 To-be Context

Two new actions capture possible decomposition of stories into smaller tasks.

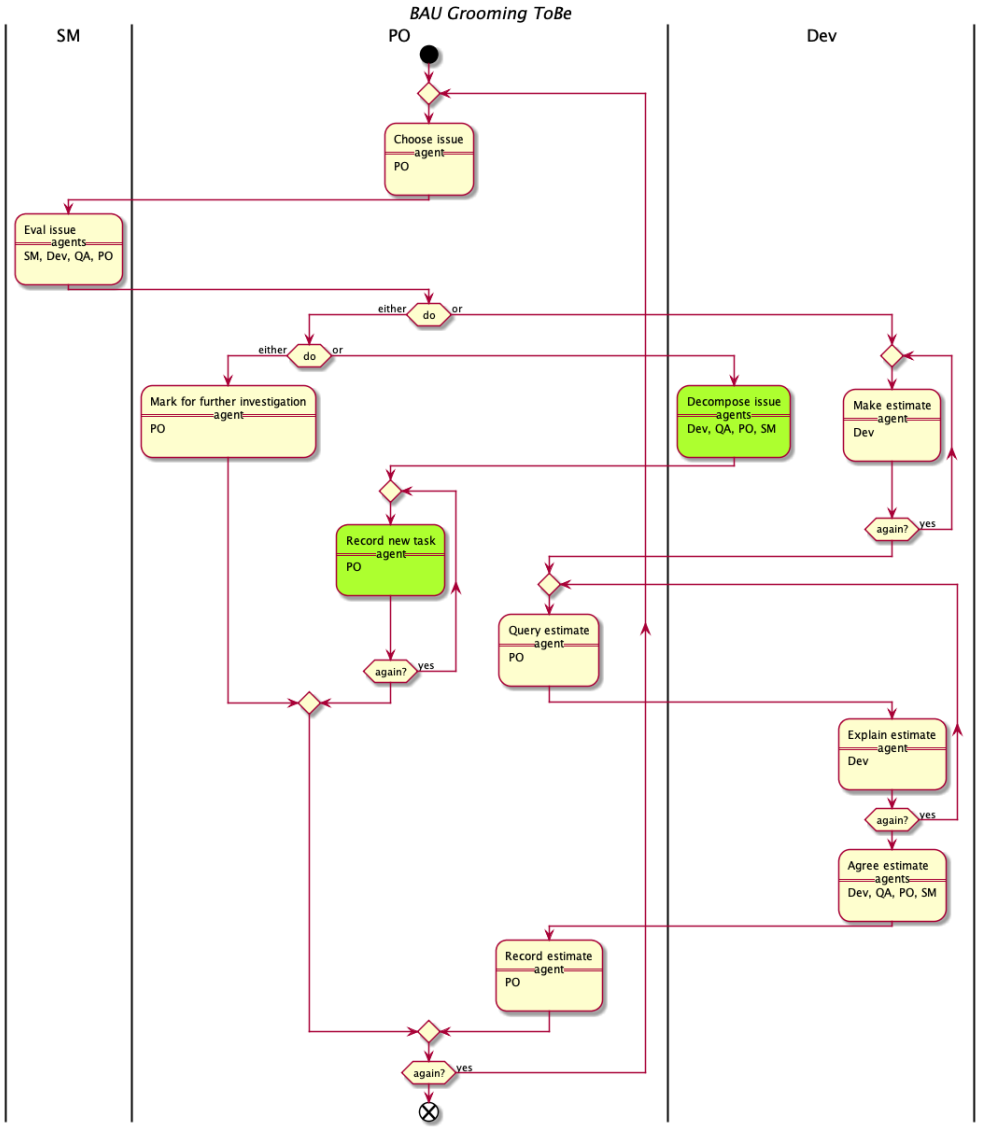

Figure I.11: S4T15 To-be context.

I.2.3.5 Stakeholders

PO, SM, Dev, QA.

I.2.3.6 Entry and Exit Criteria

- Entry Criteria: On candidate backlog ticket
- Exit Criteria: Estimated ticket, Needs further investigation ticket, On product backlog ticket

I.2.3.7 Implementation details

Table I.5: Implementation details for practice S4T15 – Decompose stories into tasks.

Step	Description	Role(s)
Choose issue	The Product Owner chooses issue from candidate backlog.	PO
Eval issue	Team discusses complexity to decide if they understand enough to implement, need further information from Architect/PO/Tech Support, need to spike to understand solution options, or if the issue can go directly to QA.	SM, Dev, QA, PO
Mark for further investigation	PO records fact that issue needs further investigation.	PO
Decompose issue	Team breaks issue down into smaller tasks.	Dev, QA, PO, SM
Record new task	PO records new tasks on product backlog.	PO
Make estimate	Developer makes an estimate via skype chat.	Dev
Query estimate	The Product Owner queries reason behind estimate.	PO
Explain estimate	Developer explains reason behind estimate.	Dev
Agree estimate	Team agrees on final estimate, based in part on opinion of most knowledgeable member.	Dev, QA, PO, SM
Record estimate	The Product Owner records agreed estimate.	PO

I.2.4 Decompose larger initiatives (S4T35)

I.2.4.1 Purpose

The purpose of this practice is to break larger initiatives into stories to ease estimation of large work items.

I.2.4.2 Background

At the portfolio and large solution levels, it is often necessary to estimate larger work items to determine their potential economic viability (Epics and Capabilities). Also, developing a program Roadmap requires:

- A knowledge of estimating (how big is the item)
- Agile Release Train (ART) velocity (how much capacity does the ART have to do it)

Larger items tend to be harder to estimate, which increases risk. Therefore, they are split into user stories, just as Scrum/XP teams do, to improve understanding, increase the accuracy of estimation and make it easier for the PO to prioritize the work. Stories are then estimated in normalized story points. This provides the ability for the Enterprise to combine estimates from various types of teams, without excessive debate.

I.2.4.3 As-is Context

Actions altered by this practice are shown in purple.

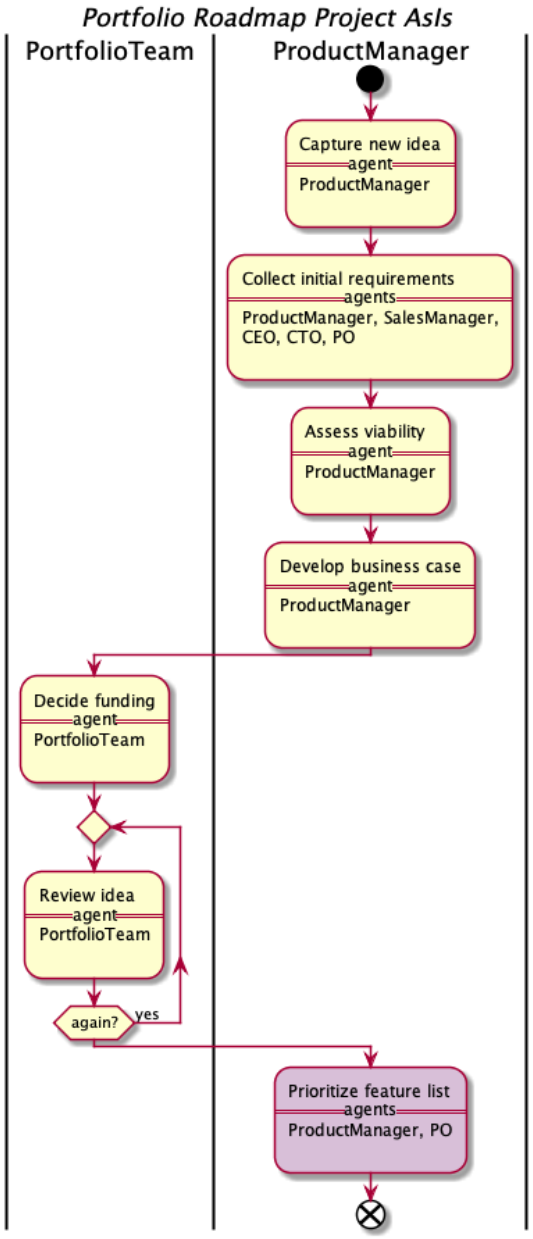

Figure I.12: S4T35 As-is context.

I.2.4.4 To-be Context

New actions are shown in green.

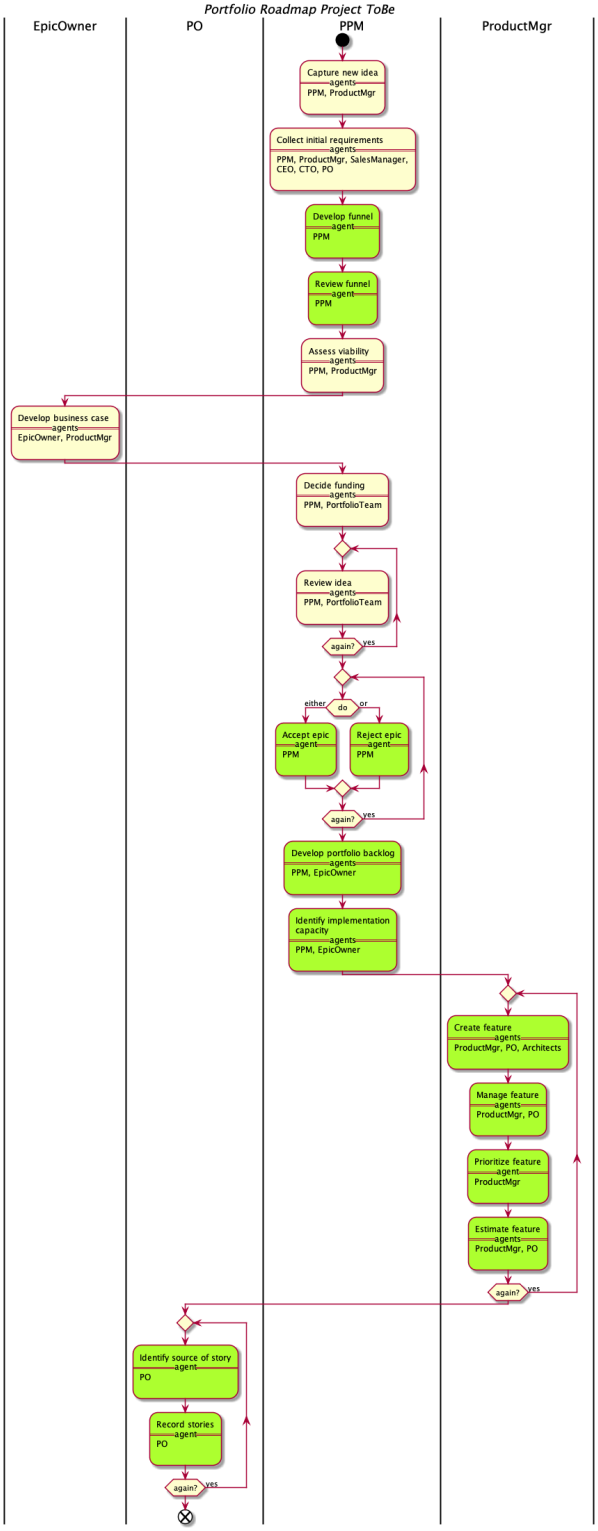

Figure I.13: S4T35 To-be context.

I.2.4.5 Stakeholders

PPM, Product Mgr, Sales Manager, CEO, CTO, PO, Epic Owner, Portfolio Team, Architects

I.2.4.6 Entry and Exit Criteria

- Entry Criteria: New idea
- Exit Criteria: Team Backlog

I.2.4.7 Implementation details

Table I.6: Implementation details for practice S4T35 – Decompose larger initiatives.

Step	Description	Role(s)
Capture new idea	The Product Manager maintains a funnel of possible new features and new integration with other products, that are submitted by sales people, management, or developers. These may be ideas they learn from customers, at trade shows, or as a result of working on the product. It's possible for anyone to add an idea to the funnel, but in practice the Product Manager is the one who does this, based on input from other parts of the organization.	PPM, Product Mgr
Collect initial requirements	The Product Manager collects and records initial requirements from the CEO, CTO, a sales person, or anyone from the company who can contribute. This starts with a very high level idea, then proceeds to initial review to see if it is viable.	PPM, Product Mgr, Sales Manager, CEO, CTO, PO
Develop funnel	This is the capture state, where all new big ideas are welcome. Funnel epics are discussed on a periodic cadence established by Program Portfolio Management. Epics that meet the decision criteria are promoted to the Review queue.	PPM

Continued on next page

Table I.6 – *Continued from previous page*

Step	Description	Role(s)
Review funnel	In this stage the preliminary estimates of opportunity, effort, and cost of delay are established. Sources of business benefit are identified and items are prioritized using WSJF. Epics that rise to the top are pulled into the Analysis queue as soon as space is available.	PPM
Assess viability	The Product Manager does an initial assessment of the viability of a new idea.	PPM, Product Mgr
Develop business case	For ideas that seem viable, the Product Manager develops a business case for review by the Portfolio Team.The business case may be as simple as an email message, or an elaborate presentation to the Portfolio Team.	Epic Owner, Product Mgr
Decide funding	Based on the feasibility study (which includes the business case), the Portfolio Team decides whether to fund development or not. The CFO and DoD support the CEO during this process.	PPM, Portfolio Team
Review idea	The Portfolio Team reviews the portfolio list once each quarter for road mapping direction and add-ons. They also gather revenue information by comparing with the previous meeting. In this stage, the CTO and DoD also decide how each portfolio item is decomposed into smaller tickets, and identify in which phase/quarter that ticket will be delivered.	PPM, Portfolio Team
Accept epic	If the business case is convincing, the epic is approved.	PPM
Reject epic	If the business case is not convincing, the epic is rejected.	PPM
Develop portfolio backlog	At this stage all epics that have made it through the portfolio Kanban with 'go' approval wait in the Portfolio Backlog until capacity is available. These epics are reviewed on a periodic basis, and this queue represents a low-cost holding pattern for upcoming implementation work. Epics from this queue are promoted to the Implementing queue when there is sufficient capacity from one or more value streams or Agile Release Trains.	PPM, Epic Owner

Continued on next page

Table I.6 – *Continued from previous page*

Step	Description	Role(s)
Identify implementation capacity	When capacity becomes available, epics are transitioned to the relevant Program and Value Stream Kanbans, where implementation begins.	PPM, Epic Owner
Create feature	In collaboration with Product Owners and other key stakeholders, features are created by Product Managers in the local context of an ART. Enabler features pave the Architectural Runway, support exploration, or may provide the infrastructure needed to develop, test, and integrate the initiative. Enabler features are generally created by Architects or Engineers and maintained in the program backlog alongside business features.	Product Mgr, PO, Architects
Manage feature	Through a program backlog Product Manager manages all features.	Product Mgr, PO
Prioritize feature	SAFe applies WSJF for continuous prioritization of features in the program backlog.	Product Mgr
Estimate feature	Feature estimation usually occurs in the 'refinement' state of the Program Kanban and relies on normalized estimation techniques, equivalent to the approach used by Agile Teams for estimating stories. Feature estimation at this point, however, does not require full breakdown into stories or involve all the teams that possibly will be involved in feature development. Instead, select subject matter experts may be involved in basic exploration and sizing.	Product Mgr, PO
Identify source of story	Stories are generally driven by splitting business features and enabler features from program backlog	PO
Record stories	The PO records all user and enabler stories into team backlog.	PO

I.3 Roadmap C1

Document and define customer base and functions relative to the application being developed.

I.3.1 GTM Context

Fig. I.14 provides GTM context of the recommendation C1.

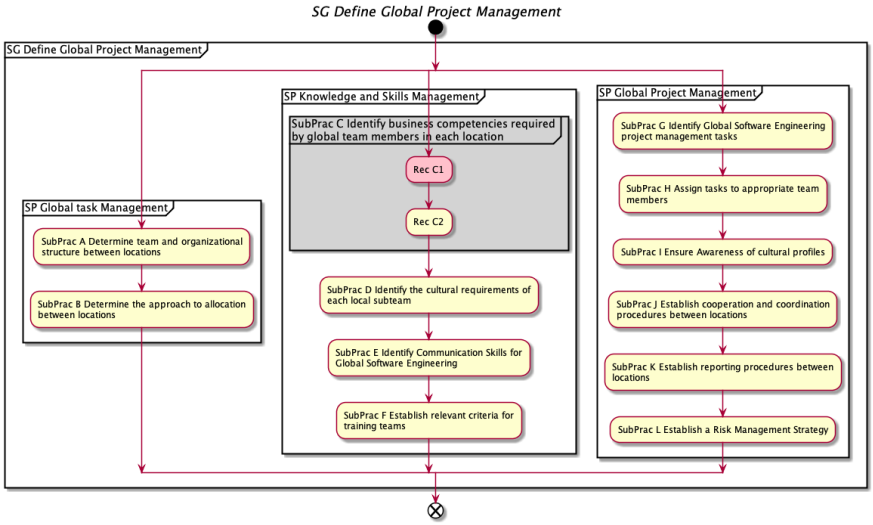

Figure I.14: C1 GTM context.

I.3.2 Why implement this recommendation?

From observations and interviews:

- Poor coordination of resources and artifacts means that testing is difficult (e.g. need to have both the file and the sample data).
- Pressure from the customer can drive project managers to give incorrect, rushed estimates about the application that is being developed.
- Regional differences impose additional requirements, which in turn delays the development process.

- Lack of local market knowledge causes delays. For example how insurance is handled in the US varies from region to region.
- Lack of synchronisation in the product line.
- Risk identification is crucial especially when we are dealing with third parties.
- Third party integration.

I.3.3 Accept stories continuously to improve flow (S4T128)

I.3.3.1 Purpose

This practice ensures functionality that is incorrectly implemented is detected as soon as possible, so corrections can be made (Leffingwell, 2015)". The flow is improved by continuously accepting stories. This allows for quick identification of problems and creates a more secure base on which to build new functionality. This focus prevents the occurrence of context switching that is common occurrence with rework. SAFe requires teams to manage work throughout each iteration timebox (as individual teams or members of the Agile Release Train), stating that empowering the teams to "focus on rapid value delivery fuels the team with energy, motivation, purpose". In fact "optimizing program execution" is one of SAFe's four core values.

I.3.3.2 Background

The ideal is that all teams follow the same iteration cadence and duration. This allows you to synchronize your work and prepares you for the Team Demo and System Demo (that requires integration and evaluation as well as the demonstration). We have provided different practice outlines for both the Team Demo and System Demo – which are two very different ceremonies. According to SAFe v4, a successful iteration execution:

1. Tracks iteration progress (where stories or Kanban boards track iteration progress)

2. Builds stories serially and incrementally
3. Allows for constant communication
4. Improves flow
5. Executes the program.

I.3.3.3 As-is Context

Currently, implemented issues are demonstrated during the sprint retrospective. In the 'To Be' model, the actions shown in purple will move to the middle of the sprint execution.

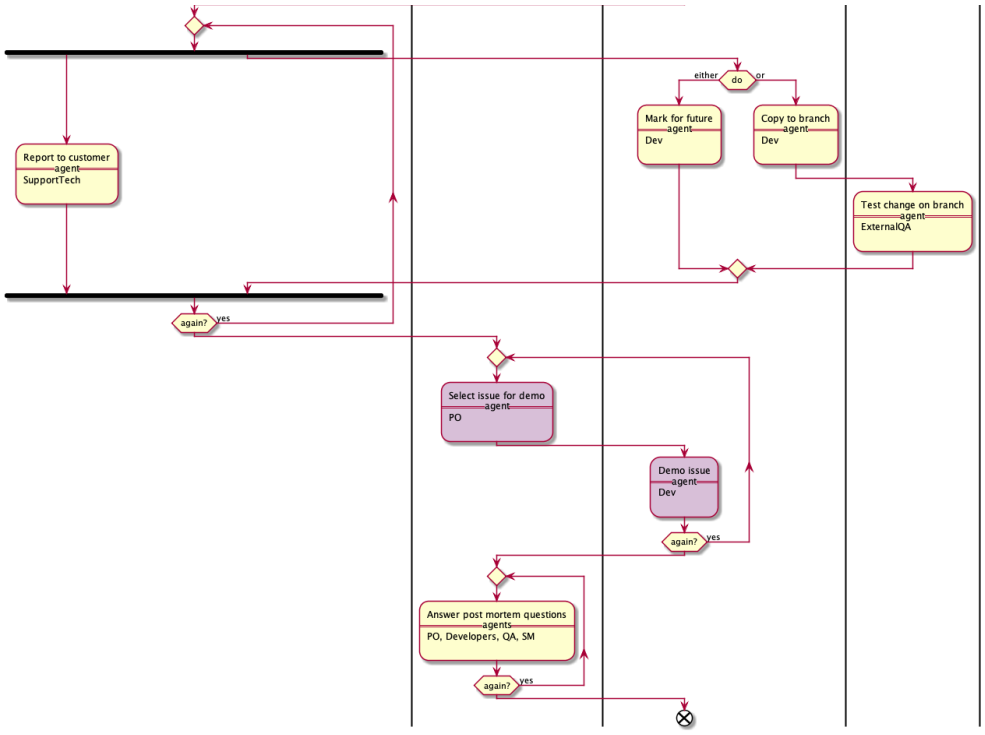

Figure I.15: S4T128 As-is context.

I.3.3.4 To-be Context

Using this practice, the implementation of an issue is demonstrated during the sprint, as soon as it is complete and tested. This spreads the load over the sprint,

and so reduces the time required for the sprint retrospective, which can be focused on process improvements.

Actions shown in green are moved from the end of the sprint to the middle.

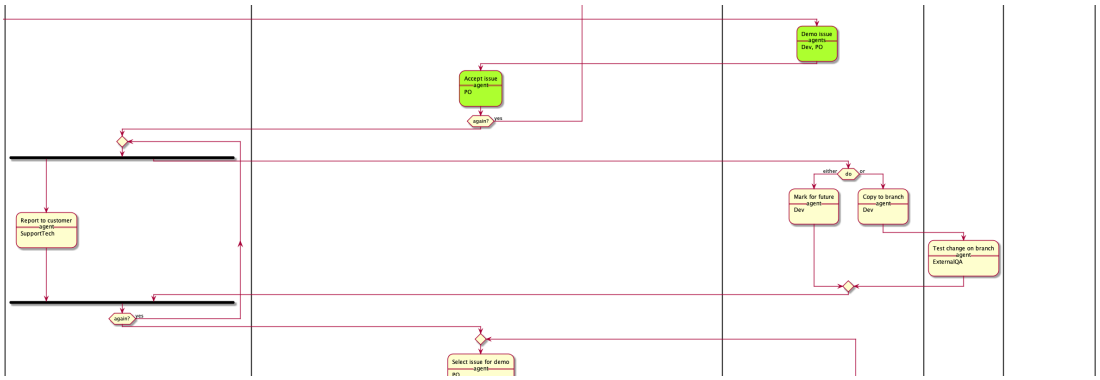

Figure I.16: S4T128 To-be context.

I.3.3.5 Stakeholders

PO, SM, Dev, QA, Architect, Sr Dev, Support Tech, ExternalQA, Developers

I.3.3.6 Entry and Exit Criteria

- Entry Criteria: PI objectives, Booked holidays, Team availability
- Exit Criteria: Improvement Story

I.3.3.7 Implementation details

Table I.7: Implementation details for practice S4T128 – Accept stories continuously to improve flow.

Step	Description	Role(s)
Prepare preliminary iteration goals	The Product Owner prepares preliminary iteration goals.	PO
Quantify capacity	The Team quantifies (ideal) capacity.	PO, SM, Dev, QA

Continued on next page

Table I.7 – *Continued from previous page*

Step	Description	Role(s)
Calculate previous iteration velocity	The Scrum Master calculates previous iteration velocity.	SM
Calculate real capacity	The Scrum Master calculates real capacity for upcoming iteration using ideal capacity and velocity from previous iteration.	SM
Agree sprint goal	Team agree on objectives for current sprint.	SM, PO, Dev, QA
Agree effort target	The Scrum Master sets initial target (in points) based on velocity of last three sprints, then subtracts points if people will be off on leave or training, or working on other tasks. Process adjustments from retrospective will be factored in somehow.	SM, Dev, QA
Propose issue	The Product Owner proposes *dev issue* for sprint backlog.	PO
Eval issue	The Team decides whether issue will fit into current sprint. The Scrum Master guides this decision. If there is a large high-priority task already on the backlog, lower-priority tasks will be added to fill the backlog. Also, the backlog should contain both large and small tasks so developers can see progress, and should not load QA at the end while leaving him idle at beginning.	SM, Dev, QA
Break issue down into tasks	Team breaks story down into smaller tasks.	PO, SM, Dev, QA
Estimate task	Team estimates task resulting from breakdown of story into tasks.	SM, Dev, QA
Add task to backlog	If issue now fits into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Estimate issue	Team estimates stories recorded in the team backlog.	SM, Dev, QA
Add issue to backlog	If issue will fit into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Team commit to iteration goals	Team commits to iteration goals considering the capacity and estimates.	PO, SM, Dev, QA
Begin change	Developer chooses a ticket from the sprint backlog.	Dev
Understand issue	Developer investigates issue and gathers background to design implementation of change.	Dev, PO, QA, Architect

Continued on next page

Table I.7 – *Continued from previous page*

Step	Description	Role(s)
Identify unit	Developer identifies code unit to be modified. May involve asking QA	Dev, Architect, Sr Dev
Implement change	Developer modifies unit to implement change.	Dev
Unit test change	Developer manually unit tests modified unit.	Dev
Provide example	The Product Owner provides examples of desired behaviour, for example a sketch of new screen layout.	PO
Write acceptance test	QA writes functional acceptance test specification in business language, as black-box tests, using an acceptance test template or checklist.	QA, PO
Automate acceptance test	QA or Developer turns test specification into an automated acceptance test using a framework like Cucumber.	QA, Dev
Add to acceptance test suite	QA or Developer incorporates test into growing acceptance test suite.	QA, Dev
Run acceptance tests	Developer runs automated acceptance tests.	Dev
Commit change	Developer commits change to trunk.	Dev
Integrate change	Commit kicks off automated build and continuous integration mechanism to build for all target platforms and configurations.	Dev
Make integration results visible	When the integration process breaks, everybody should know how and why it broke. And when it's fixed, they should know what fixed it, too.	Dev
Make fixing a failing integration top priority	When a build for any target platform fails, the team stops and fixes it immediately. Teams use flashing lights when a build is broken, and highly visible indicators of percentages of the time the system is broken.	Dev
Unit test integrated product	Developer unit-tests modified unit.	Dev
Perform system test	QA lead performs system test to verify correct behavior.	QA
Mark resolved	QA marks issue as tested successfully.	QA
Add to PMQ	QA lead adds new issue to Product Management Queue	QA
Return to dev		
Demo issue	Developer demonstrates the implementation to the Product Owner.	Dev, PO

Continued on next page

Table I.7 – *Continued from previous page*

Step	Description	Role(s)
Accept issue	The Product Owner accepts implementation, based on satisfactory demonstration of desired functionality.	PO
Report to customer	Tech support notifies customer that issue is changed and will be part of a future release.	Support Tech
Mark for future	Developer marks issue as needing to be copied to branch but blocked because branch is protected.	Dev
Copy to branch	If change should be included in branch, Developer must copy code to unit on branch.	Dev
Test change on branch	QA tests change on branch.	ExternalQA
Select issue for demo	The Product Owner selects issue for demonstration in retrospective.	PO
Demo issue	Developers demonstrate issue.	Dev
Assess iteration goal	The Team assesses whether they met the iteration goal.	PO, Developers, QA, SM
Compute velocity	The Scrum Master computes velocity as a function of story points completed and ideal effort available.	SM
Compute other metrics	The Team collects any other metrics they have agreed to analyze, such as unit test coverage, defect density, new test cases created or automated, fraction of tests automated, number of refactoring, etc. Agile Teams collect and apply other Iteration Metrics for visibility and to help with process improvement.	SM
Review improvement story	The Team reviews the improvement stories from the previous iteration. Improvement stories that were accomplished are marked as such; for those not accomplished, the team must decide what to do about them.	PO, Developers, QA, SM
Propose improvement ideas	Individual Team Members write ideas on post-its, which are then analyzed by the team for patterns.	PO, Developers, QA, SM
Acknowledge help	Team Members acknowledge who has helped them in the previous iteration.	PO, Developers, QA, SM

Continued on next page

Table I.7 – *Continued from previous page*

Step	Description	Role(s)
Describe iteration in one word	Each Team Member states one word to describe the iteration.	PO, Developers, QA, SM
Rate iteration	Each Team Member rates the iteration on a five-point scale. The team then brainstorms ways to make the next iteration score at the top of the scale.	PO, Developers, QA, SM
Answer post-mortem questions	Each Team Member provides answers to three questions: what went well, what did not go so well, and what can be done better next time.	PO, Developers, QA, SM
Create improvement stories	The Team develops improvement stories by assessing the iteration. Using the outputs from the above actions, the team develops one or two improvement stories to add to the team backlog.	PO, Developers, QA, SM

I.3.4 Use acceptance test-driven development (ATDD) (S4T292)

I.3.4.1 Purpose

This practice ensures functionality is well-defined using automated acceptance tests, so developers know exactly what is required.

I.3.4.2 Background

Automated acceptance tests, developed by the PO and QA lead, provide an unambiguous specification of the required functionality. According to SAFe “code is developed and tested in small increments, often with the development of the test itself preceding the development of the code (Leffingwell, 2015).” The focus is on the system’s intended behavior, and is said to reduce the need for long requirements specs. Where it isn’t possible to automate the tests, using TDD means you at least have a clear statement of "what the system actually does" as opposed to early thoughts on was it is “supposed to do.”

I.3.4.3 As-is Context

At present, developers manually "unit" test their changes before handing over to QA for “system” testing.

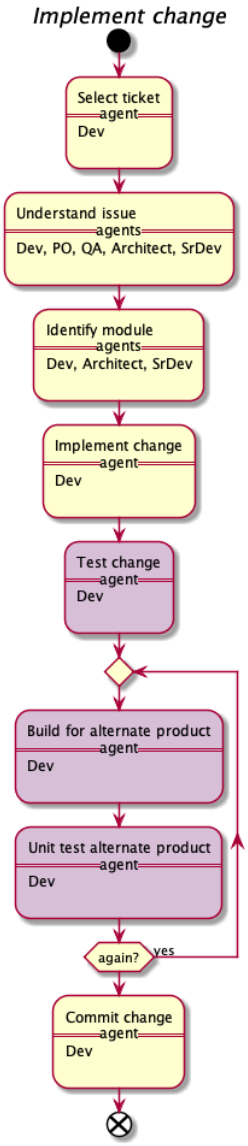

Figure I.17: S4T292 As-is context.

I.3.4.4 To-be Context

Using acceptance Test-Driven Development (ATDD), the PO and QA personnel develop automated acceptance tests that developers can use to validate their implementation before handing over to QA for system testing. This has two benefits:

1. The workload on QA is spread more evenly over the entire sprint, as QA are involved in developing tests as well as testing.

2. Tests are run automatically, so failures are detected before handing over to QA; as such, testers can focus on "deep" system testing that is hard to automate.

New actions are shown in green.

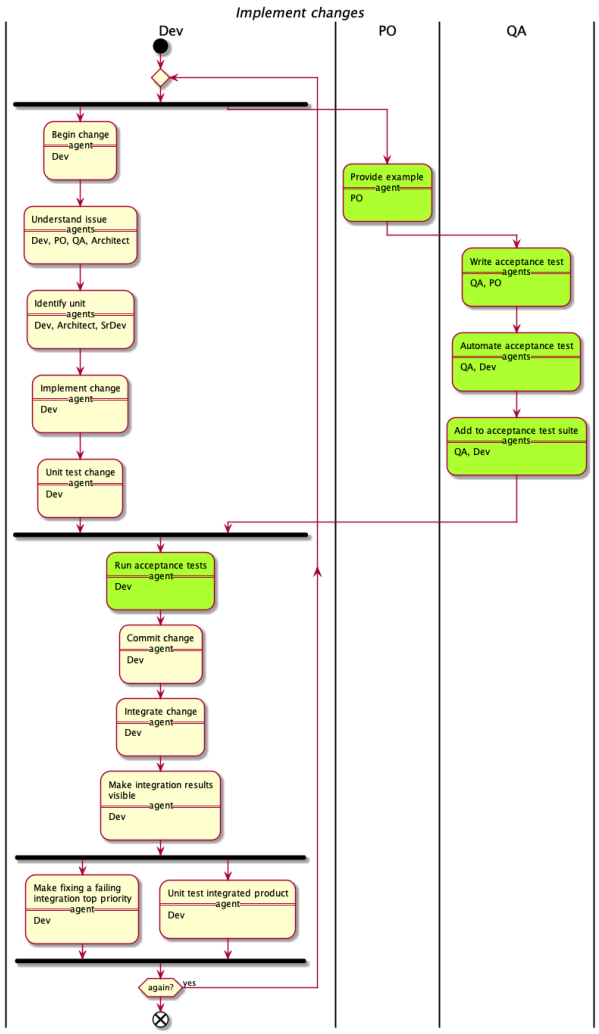

Figure I.18: S4T292 To-be context.

I.3.4.5 Stakeholders

PO, SM, Dev, QA, Architect, Sr Dev, Support Tech, ExternalQA, Developers

I.3.4.6 Entry and Exit Criteria

- Entry Criteria: PI objectives, Booked holidays. Team availability
- Exit Criteria: Improvement story created

I.3.4.7 Implementation details

Table I.8: Implementation details for practice S4T292 – Use acceptance test-driven development (ATDD)

Step	Description	Role(s)
Prepare preliminary iteration goals	The Product Owner prepares preliminary iteration goals.	PO
Quantify capacity	The Team quantifies (ideal) capacity.	PO, SM, Dev, QA
Calculate previous iteration velocity	The Scrum Master calculates previous iteration velocity.	SM
Calculate real capacity	The Scrum Master calculates real capacity for upcoming iteration using ideal capacity and velocity from previous iteration.	SM
Agree sprint goal	Team agree on objectives for current sprint.	SM, PO, Dev, QA
Agree effort target	The Scrum Master sets initial target (in points) based on velocity of last three sprints, then subtracts points if people will be off on leave or training, or working on other tasks. Process adjustments from retrospective will be factored in somehow.	SM, Dev, QA
Propose issue	The Product Owner proposes *dev issue* for sprint backlog.	PO

Continued on next page

Table I.8 – *Continued from previous page*

Step	Description	Role(s)
Eval issue	The Team decides whether issue will fit into current sprint. The Scrum Master guides this decision. If there is a large high-priority task already on the backlog, lower-priority tasks will be added to fill the backlog. Also, the backlog should contain both large and small tasks so developers can see progress, and should not load QA at the end while leaving him idle at beginning.	SM, Dev, QA
Break issue down into tasks	Team breaks story down into smaller tasks.	PO, SM, Dev, QA
Estimate task	Team estimates task resulting from breakdown of story into tasks.	SM, Dev, QA
Add task to backlog	If issue now fits into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Estimate issue	Team estimates stories recorded in the team backlog.	SM, Dev, QA
Add issue to backlog	If issue will fit into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Team commit to iteration goals	Team commits to iteration goals considering the capacity and estimates.	PO, SM, Dev, QA
Begin change	Developer chooses a ticket from the sprint backlog.	Dev
Understand issue	Developer investigates issue and gathers background to design implementation of change.	Dev, PO, QA, Architect
Identify unit	Developer identifies code unit to be modified. May involve asking QA.	Dev, Architect, Sr Dev
Implement change	Developer modifies unit to implement change.	Dev
Unit test change	Developer manually unit tests modified unit.	Dev
Provide example	The Product Owner provides examples of desired behaviour, for example a sketch of new screen layout.	PO
Write acceptance test	QA writes functional acceptance test specification in business language, as black-box tests, using an acceptance test template or checklist.	QA, PO
Automate acceptance test	QA or Developer turns test specification into an automated acceptance test using a framework like Cucumber.	QA, Dev
Add to acceptance test suite	QA or Developer incorporates test into growing acceptance test suite.	QA, Dev

Continued on next page

Table I.8 – *Continued from previous page*

Step	Description	Role(s)
Run acceptance tests	Developer runs automated acceptance tests.	Dev
Commit change	Developer commits change to trunk.	Dev
Integrate change	Commit kicks off automated build and continuous integration mechanism to build for all target platforms and configurations.	Dev
Make integration results visible	When the integration process breaks, everybody should know how and why it broke. And when it's fixed, they should know what fixed it, too.	Dev
Make fixing a failing integration top priority	When a build for any target platform fails, the team stops and fixes it immediately. Teams use flashing lights when a build is broken, and highly visible indicators of percentages of the time the system is broken.	Dev
Unit test integrated product	Developer unit-tests modified unit.	Dev
Perform system test	QA lead performs system test to verify correct behavior.	QA
Mark resolved	QA marks issue as tested successfully.	QA
Add to PMQ	QA lead adds new issue to Product Management Queue	QA
Return to dev		
Demo issue	Developer demonstrates the implementation to the Product Owner.	Dev, PO
Accept issue	The Product Owner accepts implementation, based on satisfactory demonstration of desired functionality.	PO
Report to customer	Tech support notifies customer that issue is changed and will be part of a future release.	Support Tech
Mark for future	Developer marks issue as needing to be copied to branch but blocked because branch is protected.	Dev
Copy to branch	If change should be included in branch, Developer must copy code to unit on branch.	Dev
Test change on branch	QA tests change on branch.	ExternalQA
Select issue for demo	The Product Owner selects issue for demonstration in retrospective.	PO
Demo issue	Developers demonstrate issue.	Dev

Continued on next page

Table I.8 – *Continued from previous page*

Step	Description	Role(s)
Assess iteration goal	The Team assesses whether they met the iteration goal.	PO, Developers, QA, SM
Compute velocity	The Scrum Master computes velocity as a function of story points completed and ideal effort available.	SM
Compute other metrics	The Team collects any other metrics they have agreed to analyze, such as unit test coverage, defect density, new test cases created or automated, fraction of tests automated, number of refactoring, etc. Agile Teams collect and apply other Iteration Metrics for visibility and to help with process improvement.	SM
Review improvement story	The Team reviews the improvement stories from the previous iteration. Improvement stories that were accomplished are marked as such; for those not accomplished, the team must decide what to do about them.	PO, Developers, QA, SM
Propose improvement ideas	Individual Team Members write ideas on post-its, which are then analyzed by the team for patterns.	PO, Developers, QA, SM
Acknowledge help	Team Members acknowledge who has helped them in the previous iteration.	PO, Developers, QA, SM
Describe iteration in one word	Each Team Member states one word to describe the iteration.	PO, Developers, QA, SM
Rate iteration	Each Team Member rates the iteration on a five-point scale. The team then brainstorms ways to make the next iteration score at the top of the scale.	PO, Developers, QA, SM
Answer post-mortem questions	Each Team Member provides answers to three questions: what went well, what did not go so well, and what can be done better next time.	PO, Developers, QA, SM
Create improvement stories	The Team develops improvement stories by assessing the iteration. Using the outputs from the above actions, the team develops one or two improvement stories to add to the team backlog.	PO, Developers, QA, SM

I.3.5 Perform system demo (S4PR137)

I.3.5.1 Purpose

The purpose of the System Demo is to test and evaluate the full system that the Agile Release Train (ART) is working on, and to get feedback from the primary stakeholders, including business owners, executive sponsors, other agile teams, development management, customers, and customer proxies.

I.3.5.2 Background

In addition to serving as a test of the full system that the Agile Release Train (ART) is working on, it also provides a way for the primary stakeholders – including business owners, executive sponsors, other agile teams, development management, customers, and customer proxies – to provide feedback on the evolving product.

The System Demo occurs at the end of every Iteration. It provides an integrated, aggregated view of the new Features that have been delivered by all the teams on the train in the most recent iteration (Leffingwell, 2007, 2015).

The system demo is the integrated demo of the work of all teams on the train. The system demo does not replace each team's local Team Demo, which also occurs at the end of every iteration (Leffingwell, 2007, 2015).

I.3.5.3 As-is Context

I have not observed any as-is.

I.3.5.4 To-be Context

New actions are shown in green.

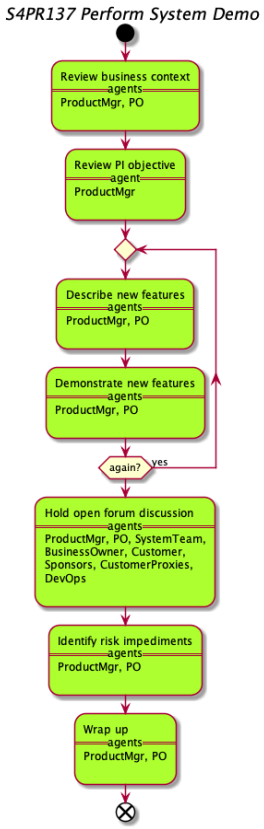

Figure I.19: S4PR137 To-be context.

I.3.5.5 Stakeholders

Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps

I.3.5.6 Entry and Exit Criteria

- Entry Criteria: Requirement
- Exit Criteria: Action items, Feedback, Progress report

I.3.5.7 Implementation details

Table I.9: Implementation details for practice S4PR137 – Perform system demo.

Step	Description	Role(s)
Review business context	The Product Manager and Product Owner review initial business context by going through each requirement of each integrated feature.	Product Mgr, PO
Review PI objective	During the system demo, the Product Manager briefly reviews the PI objectives.	Product Mgr
Describe new features	The Product Manager and PO describe integrated and aggregated view of the new features that have been delivered by the all teams on the train in the most recent iteration.	Product Mgr, PO
Demonstrate new features	The Product Manager and PO demonstrate each new feature in an end-to-end use case.	Product Mgr, PO
Hold open forum discussion	At the end of the system demo, the Product Manager opens the forum for questions and comments.	Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps
Identify risk impediments	Based on the feedback from the open forum, the Product Manager and PO note the current risks and impediments.	Product Mgr, PO
Wrap up	The Product Manager and PO wrap up by summarizing progress, feedback, and action items.	Product Mgr, PO

I.3.6 Release value (S4PR50)

I.3.6.1 Purpose

Every release of the product delivers value to the customer. Moreover, a solution cannot be fully validated until it is released to the customer environment.

I.3.6.2 Background

One of the goals of the Lean-Agile enterprise is to release the product more frequently. This provides meaningful feedback about efficacy, deployability, and usability in the development environment. In SAFe, the Release Train Engineer (RTE) identifies a possible release (e.g., a batch) to be tested in the end-user environment before deployment. Then, the QA tests the batch or integrated solution provided by the Developer.

I.3.6.3 As-is Context

Actions altered by this practice are shown in purple.

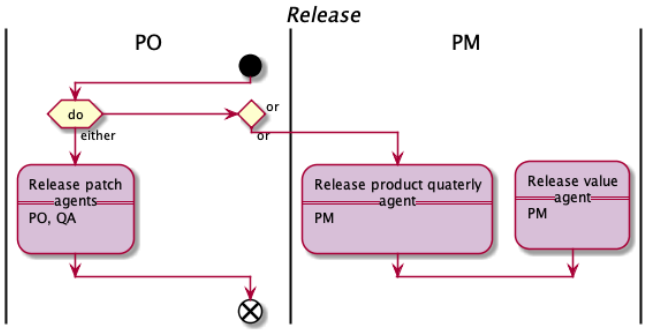

Figure I.20: S4PR50 As-is context.

I.3.6.4 To-be Context

Identify release and release value (from AsIs) is combined into one action.

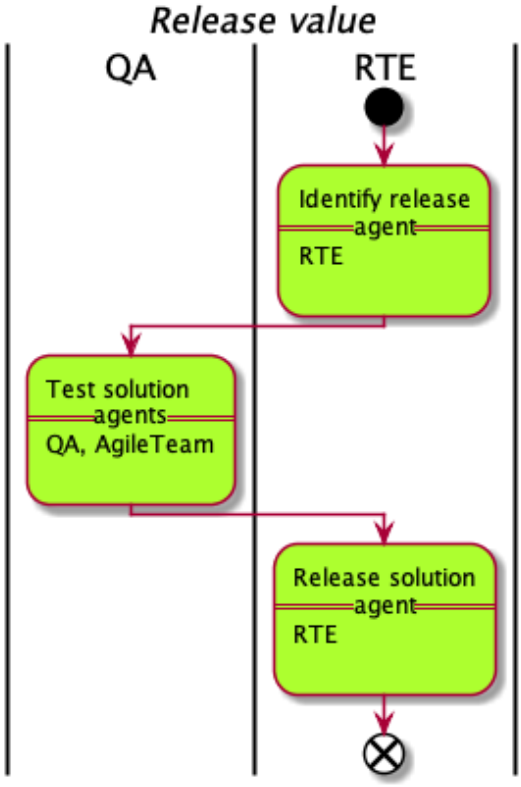

Figure I.21: S4PR50 To-be context.

I.3.6.5 Stakeholders

RTE, QA, Agile Team.

I.3.6.6 Entry and Exit Criteria

- Entry Criteria: Existing System
- Exit Criteria: Feedback

I.3.6.7 Implementation details

Table I.10: Implementation details for practice S4PR50 – Release value.

Step	Description	Role(s)
Identify release	RTE identifies possible release (e.g., batch) to be tested in the end-user environment before deployment. A release is a collection of files that can be source code, data base, or an environment. It's a collection of files that goes to production and deliver typically value -new value for customers. Then there is a release for customers. Specific customers and a release for geographical areas.	RTE
Test solution	QA tests the batch or integrated solution provided by the developer.	QA, Agile Team
Release solution	Releasing a tested solution to the end-user environment provides meaningful feedback about efficacy, deployment, and usability.	RTE

I.3.7 Provide fast intimate feedback (S4PR3)

I.3.7.1 Purpose

The purpose of this practice is to shape the solution intent where the customer frequently and closely works with solution and product management. As such the customer needs to identify the ceremonies they need to participate in that are essential to continuous execution.

I.3.7.2 Background

The customer participates continuously throughout agile solution development either in person or by proxy to fulfil the following general responsibilities (Leffingwell, 2015)]:

- Participate as Business Owner in PI planning.
- Attend solution and possibly system demos to evaluate the solution increment.
- Participate in Inspect and Adapt workshops to assist in removing some systemic impediments.

- Interact with analysts and subject matter experts during specification workshops.
- Collaboratively manage scope, time, and other constraints with Product and Solution Management.
- Help to define the Roadmap, Milestones, and Releases.
- Communicate the economic logic behind the solution and help validate assumptions in the Economic Framework.
- Review the technical and financial status of the solution.
- Participate in beta testing, User acceptance testing (UAT), other forms of solution validation.

I.3.7.3 As-is & To-be Context

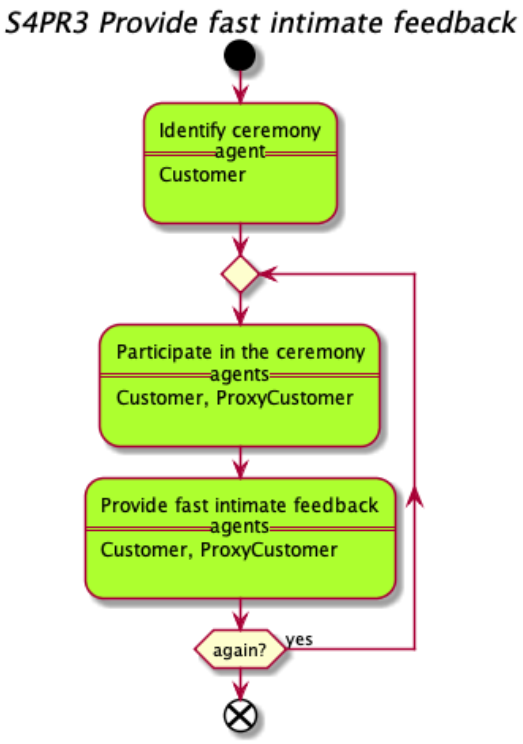

Figure I.22: S4PR3 To-be context.

I.3.7.4 Stakeholders

Customer, Proxy Customer.

I.3.7.5 Entry and Exit Criteria

- Entry Criteria: Ceremony
- Exit Criteria: Feedback

I.3.7.6 Implementation details

Table I.11: Implementation details for practice S4PR3 – Provide fast intimate feedback.

Step	Description	Role(s)
Identify ceremony	To shape the solution intent the customer must work frequently and closely with solution and product management. As such, the customer needs to identify the ceremonies they need to participate in that are essential to continuous execution.	Customer
Participate in the ceremony	Customer participates in person or by proxy throughout agile solution development.	Customer, Proxy Customer

Continued on next page

Table I.11 – *Continued from previous page*

Step	Description	Role(s)
Provide fast intimate feedback Table 5: Implementation details for practice S4PR3 – Provide fast intimate feedback	Customers engage continuously throughout agile solution development. Customer participates either in person or by proxy to fulfill the following general responsibilities: 1) Participate as Business Owner in PI planning. 2) Attend solution and possibly system Demo to evaluate the solution increment. 3) Participate in Inspect and Adapt workshops to assist in removing some systemic impediments. 4) Interact with analysts and subject matter experts during specification workshops. 5) Collaboratively manage scope, time, and other constraints with Product and Solution Management. 6) Help to define the Roadmap, Milestones, and Releases. 7) Communicate the economic logic behind the solution and help validate assumptions in the Economic Framework. 8) Review technical and financial status of the solution. 9) Participate in beta testing, UAT, other forms of solution validation.	Customer, Proxy Customer

I.3.8 Understand solution feasibility (S4PR42)

I.3.8.1 Purpose

This practice provides the necessary means to understand the feasibility of the solution.

I.3.8.2 Background

"Product management team develops learning milestones via testing a concept of a new capability with a focus group, building and releasing a minimum viable product (MVP), or validating User Experience assumptions for the new functionality. Such milestones do not necessarily occur on PI boundaries and may require significant effort, not only on behalf of the product development organization but also on the part of other business functions in the Enterprise, such as sales, marketing, operations, finance, etc." (Leffingwell, 2007)

I.3.8.3 As-is Context

Actions altered by this practice are shown in purple.

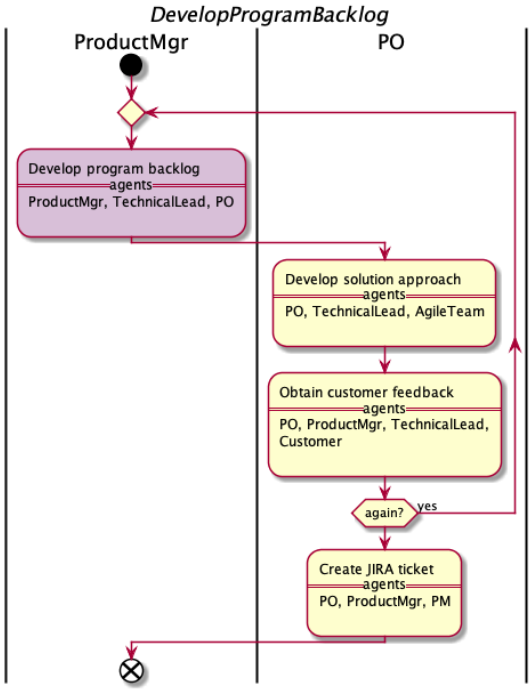

Figure I.23: S4PR42 As-is context.

I.3.8.4 To-be Context

New actions are shown in green.

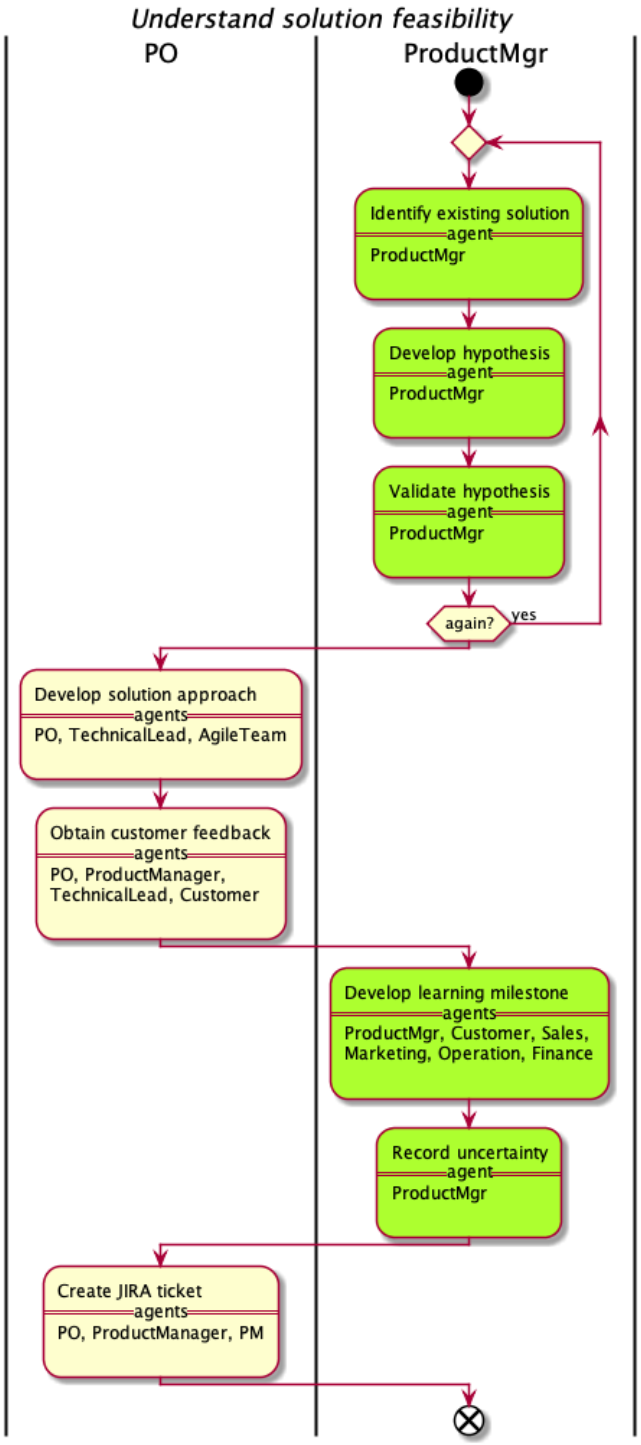

Figure I.24: S4PR42 To-be context.

I.3.8.5 Stakeholders

Product Mgr, PO, Technical Lead, Agile Team, Customer, Sales, Marketing, Operation, Finance, PM.

I.3.8.6 Entry and Exit Criteria

- Entry Criteria: Existing System
- Exit Criteria: Feedback

I.3.8.7 Implementation details

Table I.12: Implementation details for practice S4PR42 – Understand solution feasibility.

Step	Description	Role(s)
Identify existing solution	The Product Manager identifies an existing solution from the product line.	Product Mgr
Develop hypothesis	The Product Manager develops the hypothesis to conceptualize and validate the existing solution against the actual market condition.	Product Mgr
Validate hypothesis	The Product Manager asks the following questions to validate hypothesis against actual market condition: (1) Do the new product capabilities have a market that is ready to pay for them? (Not relevant to customer driven project.) (2) Do they solve the user problem for the users being targeted? (3) Are the necessary non-financial accounting measures available to demonstrate real progress? (4) What revenue can the organization expect? (5) Is there a viable business model to support the new product or capability?	Product Mgr
Develop solution approach	Product Owner goes through each of the development needs with the Technical Lead to create a specification.	PO, Technical Lead, Agile Team

Continued on next page

Table I.12 – *Continued from previous page*

Step	Description	Role(s)
Obtain customer feedback	Product Manager and Technical Lead send the solution approach to the Customer. If they agree, then the specification is logged into JIRA.	PO, Product Manager, Technical Lead, Customer
Develop learning milestone	The Product Management team develops learning milestones via testing a concept of a new capability with a focus group, building and releasing a minimum viable product (MVP), or validating User Experience assumptions for the new functionality. Such milestones do not necessarily occur on PI boundaries and may require significant effort, not only on behalf of the product development organization but also on the part of other business functions in the Enterprise, such as sales, marketing, operations, finance, etc.	Product Mgr, Customer, Sales, Marketing, Operation, Finance
Record uncertainty	The Product Manager records the degree of uncertainty that needs to be translated into knowledge and, ultimately, into business benefits for the organization. This requires set-based thinking and the ability to pivot, if necessary, to a different concept of the solution. The set-based design is particularly effective in programs that require a high degree of fixed schedule commitments. After all, since the schedule is unmovable, it makes sense to keep multiple design options present, even if some of the more reliable design options do not necessarily provide the degree of innovation or enhanced performance that the systems developers would otherwise prefer. But when the deadline is sacrosanct, teams must do what they can within the schedule.	Product Mgr
Create JIRA ticket	Product Manager, PO, or Project Manager creates tickets in JIRA for each agreed requirement. Each ticket contains an initial estimate, time, and release plan (in which phase specific ticket will be released).	PO, Product Manager, PM

I.3.9 Plan milestones incrementally (S4PR43)

I.3.9.1 Purpose

This practice helps to evaluate progress toward the goal.

I.3.9.2 Background

Milestones are planned incrementally since the outcome of any learning milestones impacts the understanding of intent. However, learning does not stop even when the product capabilities hit the market and start generating business benefits. In a Lean enterprise environment, learning is an integral part of the development and meaningful learning milestones can help.

I.3.9.3 As-is Context

I have not observed any as-is context.

I.3.9.4 To-be Context

To be model shown in Fig. I.25.

I.3.9.5 Stakeholders

Product Mgt, Customer.

I.3.9.6 Entry and Exit Criteria

- Entry Criteria: Product, User
- Exit Criteria: Milestone backlog

I.3.9.7 Implementation details

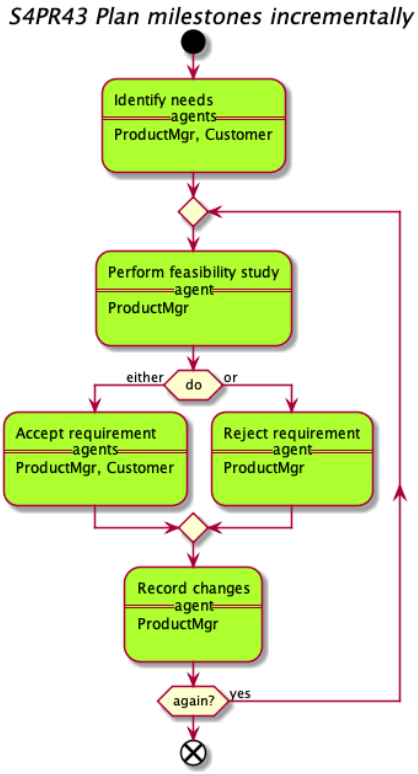

Figure I.25: S4PR43 To-be context.

Table I.13: Implementation details for practice S4PR43 – Plan milestones incrementally.

Step	Description	Role(s)
Identify needs	Product manager identifies the need for new infrastructure. For example, do the users need cloud capabilities? Or do the users need social network functionality? Or work-group subscription model?	Product Mgr, Customer
Perform feasibility study	Product manager performs a feasibility study to evaluate the need for new infrastructure or new functionality.	Product Mgr
Accept requirement	The Product manager identifies a minimum viable product (MVP) based on the feasibility study report.	Product Mgr, Customer

Continued on next page

Table I.13 – *Continued from previous page*

Step	Description	Role(s)
Reject requirement	Product manager provides a detail report of why the request is rejected and what needs to be done. For example, users do not need social network functionality, but users need capability for work-groups.	Product Mgr
Record changes	Product manager records all changes to milestone backlog.	Product Mgr

I.3.10 Release product (S4PR53)

I.3.10.1 Purpose

The purpose of releasing the product anytime is to meet development and business needs.

I.3.10.2 Background

Delivery can happen based on the business needs, however, enterprises can align delivery of some or all aspects of the solution to a program increment (PI) boundary. In these cases, the development and delivery rhythm can be the same.

A large system is not homogeneous and may contain different types of components and subsystems, each of which may have its own release model. In this case, it is most probably easiest to consider releasing whatever you want, whenever it makes sense within the governance and business model.

I.3.10.3 As-is Context

Actions altered by this practice are shown in purple.

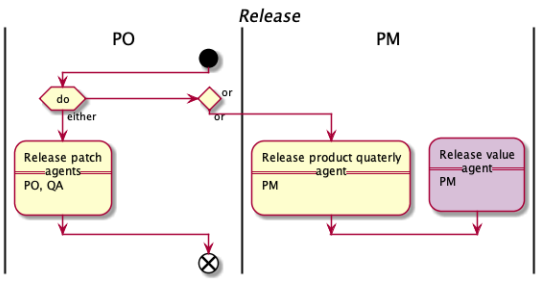

Figure I.26: S4PR53 As-is context.

I.3.10.4 To-be Context

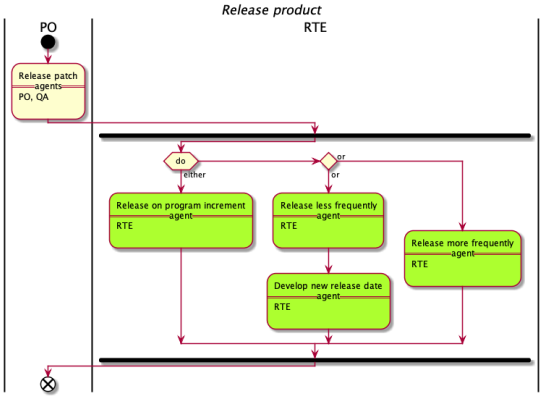

Figure I.27: S4PR53 To-be context.

I.3.10.5 Stakeholders

PO, QA, RTE.

I.3.10.6 Entry and Exit Criteria

- Entry Criteria: Existing System
- Exit Criteria: Feedback

I.3.10.7 Implementation details

Table I.14: Implementation details for practice S4PR53 – Release product.

Step	Description	Role(s)
Release patch	The agile team release small functions to the client especially to a client in a particular environment. The reason of this type of release is to validate the product at the customer end.	PO, QA

Continued on next page

Table I.14 – *Continued from previous page*

Step	Description	Role(s)
Release on program increment	The RTE identifies the PI boundaries (also known as PI cadence) that provides PI planning, releasing, and inspect and adapt's calendar date (these three activities are performed in the same cadence). In addition to these activities, innovation and planning iteration also help to identify extensive release activities that include final verification and validation, UAT, release documentation, and release dates. Based on the release dates, agile release train release the product on program increment cadence. The major release usually takes place each quarter. The integration of multiple patches could also be a quarterly release.	RTE
Release less frequently	The RTE identifies the infrastructure criticality, license agreements, and availability of new hardware before releasing the product because deployed systems can constitute critical infrastructure for a customer's operating system. If the software is new, then license agreements may be prohibitive. If the timeline of the enterprise that is building systems contains both software and hardware then the new hardware must be available first. In that case, releasing early and incrementally on PI cadence is not an option, and the planning and releasing activities need to be completely decoupled.	RTE
Develop new release date	The RTE develops a new release date based on the criticality of the product.	RTE
Release more frequently	If the enterprise builds a complex solution then various components of the system may have to be released more frequently. In that case, the enterprise can adopt a periodic planning function technique to release more frequently that provides the cadence, synchronization, and alignment to the enterprise to manage variability and deviations from expectation. However, forcing the releases to the same cadence is unnecessary and constrains the system too much.	RTE

I.3.11 Align development to business (S4PR104)

I.3.11.1 Background

PI planning is a routine, face-to-face event with a standardized agenda that includes presentation of business context and vision, followed by team planning breakouts wherein the teams create the plans for the upcoming Program Increment (PI) (Leffingwell, 2007). During the PI planning meeting the team aligns the development to the business via business context, vision, and PI objectives (Leffingwell, 2007).

I.3.11.2 As-is & To-be Context

I have not observed any as-is process model for this practice. New actions are shown in green.

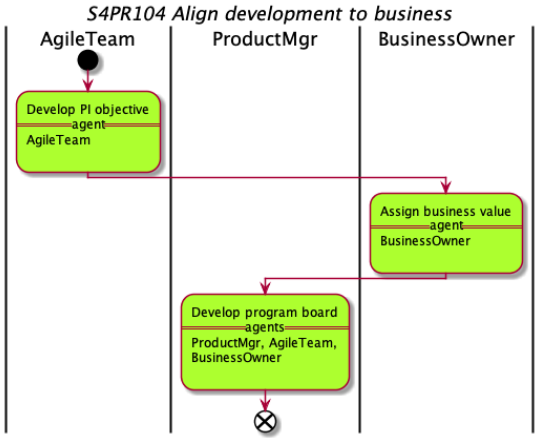

Figure I.28: S4PR104 To-be context.

I.3.11.3 Stakeholders

Agile Team, Business Owner, and Product Mgr.

I.3.11.4 Entry and Exit Criteria

- Entry Criteria: Roadmap, Vision
- Exit Criteria: Program backlog

I.3.11.5 Implementation details

Table I.15: Implementation details for practice S4PR104 – Align development to business.

Step	Description	Role(s)
Develop PI objective	The Agile Team develops a set of 'SMART' team PI objectives for themselves. The term SMART means: (1) Specific: States the intended outcome as simply, concisely, and explicitly as possible (Start with an action verb.) (2) Measurable: It should be clear what a team needs to do to achieve the objectives. The measures can be descriptive, yes/no, qualitative, or provide a range. (3) Achievable: The objective should be written with the Team's control and influence. (4) Realistic: Avoid making happy path assumptions and factors need to recognize those that cannot be controlled. (5) Time-bound: The objective must be scoped appropriately to achieve within the PI.	Agile Team
Assign business value	The Business Owner assigns business value to each PI objective and summaries a set of program PI objectives.	Business Owner
Develop program board	The Agile Team, Product Manager, and Business Owner together with develop a program backlog by highlighting new features, delivery date, and milestones that are aggregated from the team objectives.	Product Mgr, Agile Team, Business Owner

I.3.12 Empower people (S4VS11)

I.3.12.1 Purpose

The purpose of this practice is to empower people which helps to decentralized the decision making process.

I.3.12.2 Background

In SAFe, when budget elements are in place then the enterprise empowers the solution management with the relevant context, knowledge, and authority necessary to make content decisions at each levels of the framework (Leffingwell, 2007). However, the solution management does not work alone; rather, they coordinate with the larger stakeholder community to determine the best course of action.

I.3.12.3 As-is & To-be Context

I have not observed any as-is model of this practice. New actions are shown in green.

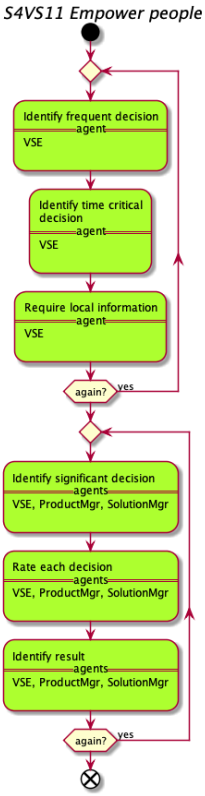

Figure I.29: S4VS11 To-be context.

I.3.12.4 Stakeholders

VSE, Product Mgr, and Solution Mgr.

I.3.12.5 Entry and Exit Criteria

- Entry Criteria: Team backlog, Program backlog, ART
- Exit Criteria: Centralize decision, Decentralize decision

I.3.12.6 Implementation details

Table I.16: Implementation details for practice S4VS11 – Empower people.

Step	Description	Role(s)
Identify frequent decision	Team or product management decentralizes decision-making process for decisions made frequently. For example, team and program backlog prioritization, real-time ART scoping, response to defects and emerging issues.	VSE
Identify time critical decision	The Team decentralizes time critical decisions. For example, point releases, customer emergencies, or dependencies with other teams. Because, a delay in these types of decisions comes at a high cost of delay.	VSE
Require local information	The Team keeps the decision-making process decentralized to identify the specific local context, whether it be technology, organization, or specific customer or market impact. For example, ship/no ship release to a specific customer, resolve a significant design problem, self-organization of individuals and teams to an emerging challenge.	VSE
Identify significant decision	The first step is to identify the pending decisions.	VSE, Product Mgr, Solution Mgr
Rate each decision	Rate each item using 0 to 2 (low to high). Frequent? Time Critical? Economies of scale?	VSE, Product Mgr, Solution Mgr
Identify result	Team calculates the total value provided in the previous step. If the total value is in between 0 to 3 then the decision is centralized and if the total value is in between 4 to 6 then the decision is decentralized. Understanding how decisions are made is a key factor in empowering knowledge workers.	VSE, Product Mgr, Solution Mgr

I.3.13 Meet contractual obligation (S4VS21)

I.3.13.1 Purpose

The purpose of this practice is to make sure that the decision-making process is decentralized.

I.3.13.2 Background

In SAFe, "the enterprise (Portfolio level) empowers product and solution management with the relevant context, knowledge, and authority to make necessary content decision at the each level of the framework (Leffingwell, 2007)." The enterprise (Portfolio level) collaborates with the "large stakeholder community to determine the best course of action (Leffingwell, 2007)."

I.3.13.3 As-is Context

Actions altered by this practice are shown in purple (see Fig. I.30).

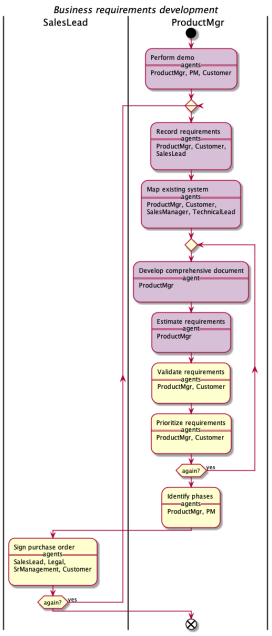

Figure I.30: S4VS21 As-is context.

I.3.13.4 To-be Context

New actions are shown in green.

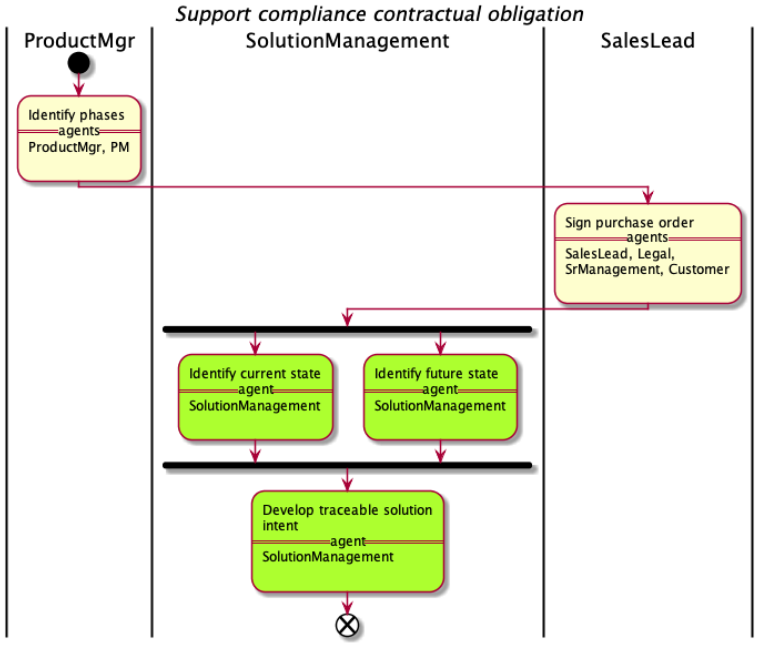

Figure I.31: S4VS21 To-be context.

I.3.13.5 Stakeholders

Product Mgr, PM, Sales Lead, Legal, Sr Management, Customer, and Solution Management.

I.3.13.6 Entry and Exit Criteria

- Entry Criteria: Customer To-be context, Prioritize List of Development Requirements.
- Exit Criteria: Traceability

I.3.13.7 Implementation details

Table I.17: Implementation details for practice S4VS21 – Meet contractual obligation.

Step	Description	Role(s)
Identify phases	PM or Product Manager divides the requirements into two phases. Phase 1 contains the requirements that are out of the box; phase 2 contains the bespoke development requirements.	Product Mgr, PM
Sign purchase order	If the Customer agrees on Phase 1 and Phase 2 requirements, then a Purchase Order is signed.	Sales Lead, Legal, Sr Management, Customer
Identify current state	Solution management identifies what exactly the current system does. That includes Specification (Capabilities, Features, Stories, NFRs, Standards), Design (Models, Design Decision, Trade Studies), and Tests (Functional and unit tests, System and NFR tests).	Solution Management
Identify future state	Solution management identifies what changes are intended for a future state. That includes Specification (Capabilities, Features, Stories, NFRs, Standards), Design (Models, Design Decision, Trade Studies), and Tests (Functional and unit tests, System and NFR tests).	Solution Management
Develop traceable solution intent	Solution management develops a traceable solution by including life-critical systems, mission-critical systems, and other systems governed by regulatory standards to ensure that the system behaves as intended (difference of current and future states provides what needs to be done). By developing a traceable solution, solution management can support to identify the compliance and contractual obligation.	Solution Management

I.3.14 Perform solution demo (S4VS113)

I.3.14.1 Purpose

The purpose of performing solution demo is the visualize the development efforts from multiple ARTs to the customers and stakeholders.

I.3.14.2 Background

In SAFe, the "solution demo is the apex event of the Program Increment cycle for a value stream (Leffingwell, 2007, 2015)." During the solution demo, the value stream team evaluates the objectives and try to get customers feedback via an open forum discussion.

Accordingly, in SAFe a typical system demo even agenda includes (Leffingwell, 2007, 2015).:

- Review the value stream PI objectives
- Demonstrate each capability
- Identify business value completed per objective
- Open forum for stakeholders feedback
- Finally, wrap up by summarising progress, feedback, and action items

I.3.14.3 As-is & To-be Context

I have not observed any as-is context for this practice. New actions are shown in green.

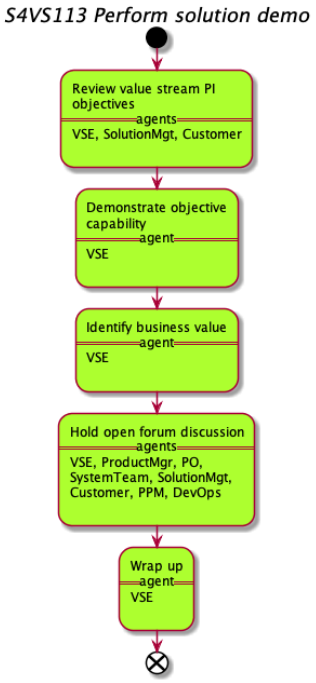

Figure I.32: S4VS113 To-be context.

I.3.14.4 Stakeholders

VSE, Product Mgr, PO, System Team, Solution Mgt, Customer, PPM, and DevOps.

I.3.14.5 Entry and Exit Criteria

- Entry Criteria: Value stream PI objectives
- Exit Criteria: Action items, Feedback, Progress report.

I.3.14.6 Implementation details

Table I.18: Implementation details for practice S4VS113 – Perform solution demo.

Step	Description	Role(s)
Review value stream PI objectives	In collaboration with other stakeholders, VSE reviews the value stream PI objectives that were agreed to at the beginning of the PI.	VSE, Solution Mgt, Customer
Demonstrate objective capability	VSE demonstrates each PI objective and capability in an end-to-end use case.	VSE
Identify business value	VSE identifies the business value completed per objective.	VSE
Hold open forum discussion	At the end of the solution demo, the VSE opens the forum for questions and comments.	VSE, Product Mgr, PO, System Team, Solution Mgt, Customer, PPM, Dev Ops
Wrap up	VSE wraps up by summarizing progress, feedback, and action items.	VSE

I.3.15 Evaluate customer deployment environment (S4VS171)

I.3.15.1 Purpose

The purpose of this practice is to make sure that, both the system and deployment environment is in the same state.

I.3.15.2 Background

"Some solution contexts are established customer environments that the solution must simply 'fit into' (Leffingwell, 2007)." However, many cases a new solution may require evaluation of the customer's deployment environment. In that case, the system builder uses the solution context to understand how their system will be packaged and deployed in its ultimate operating environment. The solution supplier to customer relationship in large system-of-systems context is a unique and cascading thing where each organization in the supply chain delivers its solution to the customer's context that specifies how the solution is

packaged, deployed, and integrated. For example, a vehicle navigation system supplier operates first, in the infotainment supplier's context, then in the vehicle manufacturer's context, and finally in the customer's context. All of these contexts can impact the viability of the solution. So the system builder must be aware of the full end-to-end value chain (Leffingwell, 2007, 2015)."

I.3.15.3 As-is & To-be Context

I have not observed any as-is context for this practice. New actions are shown in green.

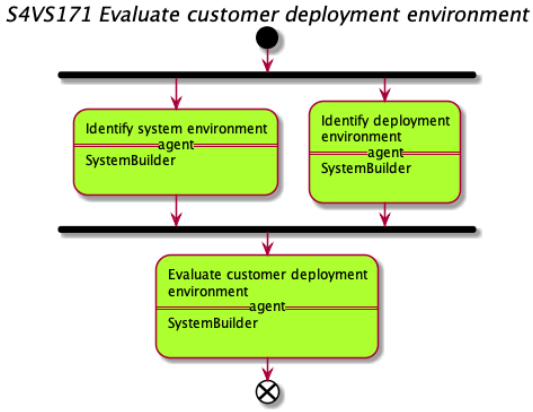

Figure I.33: S4VS171 To-be context.

I.3.15.4 Stakeholders

System Builder.

I.3.15.5 Entry and Exit Criteria

- Entry Criteria: New solution
- Exit Criteria: Evaluated Deployment environment

I.3.15.6 Implementation details

Table I.19: Implementation details for practice S4VS171 – Evaluate customer deployment environment.

Step	Description	Role(s)
Identify system environment	System Builder uses the solution context to understand how their system will be packaged and deployed in its ultimate operating environment. The solution supplier to customer relationship in large system-of-systems context is a unique and cascading thing where each organization in the supply chain delivers its solution to the customer's context that specifies how the solution is packaged, deployed, and integrated. For example, a vehicle navigation system supplier operates first, in the infotainment supplier's context, then in the vehicle manufacturer's context, and finally in the customer's context. All of these contexts can impact the viability of the solution. So the system builder must be aware of the full end-to-end value chain.	System Builder
Identify deployment environment	System Builder identifies the specific interfaces, deployed OSs, firewalls, APIs to other application hosted or cloud infrastructure.	System Builder
Evaluate customer deployment environment	The System Builder track changes as both the system and deployment environment have to evolve into a common state.	System Builder

I.3.16 Understand business objectives (S4RR138)

I.3.16.1 Purpose

The purpose of this practice is to ensure that business objectives are understood and agreed to by key stakeholders of the train.

I.3.16.2 Background

Business Owner understands and ensures that business objectives are understood and agreed to by key stakeholders on the train.

I.3.16.3 As-is & To-be Context

I have not observed any as-is model for this practices. New actions are shown in green.

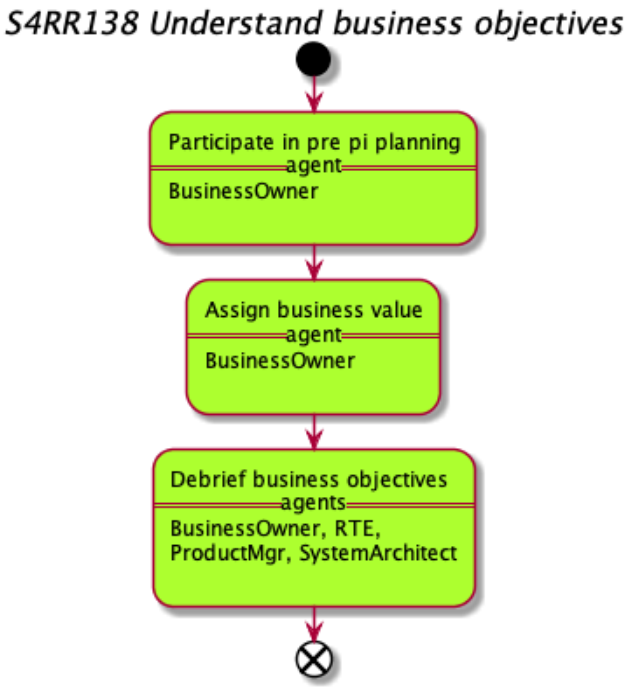

Figure I.34: S4RR138 To-be context.

I.3.16.4 Stakeholders

Business Owner, RTE, Product Mgr, and System Architect.

I.3.16.5 Entry and Exit Criteria

- Entry Criteria: PI planning
- Exit Criteria: Feedback

I.3.16.6 Implementation details

Table I.20: Implementation details for practice S4RR138 – Understand business objectives.

Step	Description	Role(s)
Participate in pre pi planning	Business owner participates in the Pre-PI planning meeting to identify the PI objectives.	Business Owner
Assign business value	Business owner assigns business value to each PI objective and summarizes a set of program PI objectives.	Business Owner
Debrief business objectives	Business owner debriefs business objectives to the key stakeholders of the train to ensure that business objectives are understood and agreed.	Business Owner, RTE, Product Mgr, System Architect

I.3.17 Develop portfolio kanban (S4PO80)

I.3.17.1 Purpose

The purpose of the Portfolio Kanban system is to capture, analyze, approve, and track epics.

I.3.17.2 Background

Overall, there are a number of stages that an epic passes through on the way to implementation or rejection. The typical collaborators of the portfolio kanban system are: Business Owner, Product and Solution Management, Epic Owner, Development Team, and Solution/System Architect. There are in total six phases in the Portfolio Kanban system (Leffingwell, 2015).

- **Funnel** – The funnel queue is the capture queue where all ideas come from different sources i.e. portfolio strategic theme. At this stage the epic does not need to have any business case or estimates; it can be stated as a short keyword or phrase. All ideas are captured for consideration and can be represented in different formats, i.e. a document, spreadsheet, or visual system on the wall. Program Portfolio Management (PPM) typically

discusses funnel epics on a periodic cadence, and epics that meet the decision criteria are moved to the review queue.

- **Review** – In this state, ideas get preliminary estimates of opportunity, effort, and cost of delay.

- **Analysis** – At this stage, more work is done to establish viability, measurable benefits, development and deployment impact, and potential availability of resources. A lightweight business case is developed and the epic is either approved or rejected at the end of this state.

- **Portfolio Backlog** – Epics with a "go" decision are kept in the portfolio backlog and reviewed on a periodic basis. The queue in the portfolio backlog represents a low-cost holding pattern for upcoming implementation work. Epics are moved to the implementing queue when there is sufficient capacity on one or more value streams or Agile Release Trains.

- **Implementation** – When the capacity becomes available then epics are transitioned to either the Program or Value Stream Kanban, where implementation begins.

- **Done** – The Epic Owner checks the implemented epic against epic success criteria. The epic is considered done when it has met all its success criteria. However, due to the scope of epics, completion to the original intent is not always the desired case. At this stage, some identified capabilities and features might be discarded. Regardless, the epic eventually reaches a done state.

I.3.17.3 As-is Context

Actions altered by this practice are shown in purple.

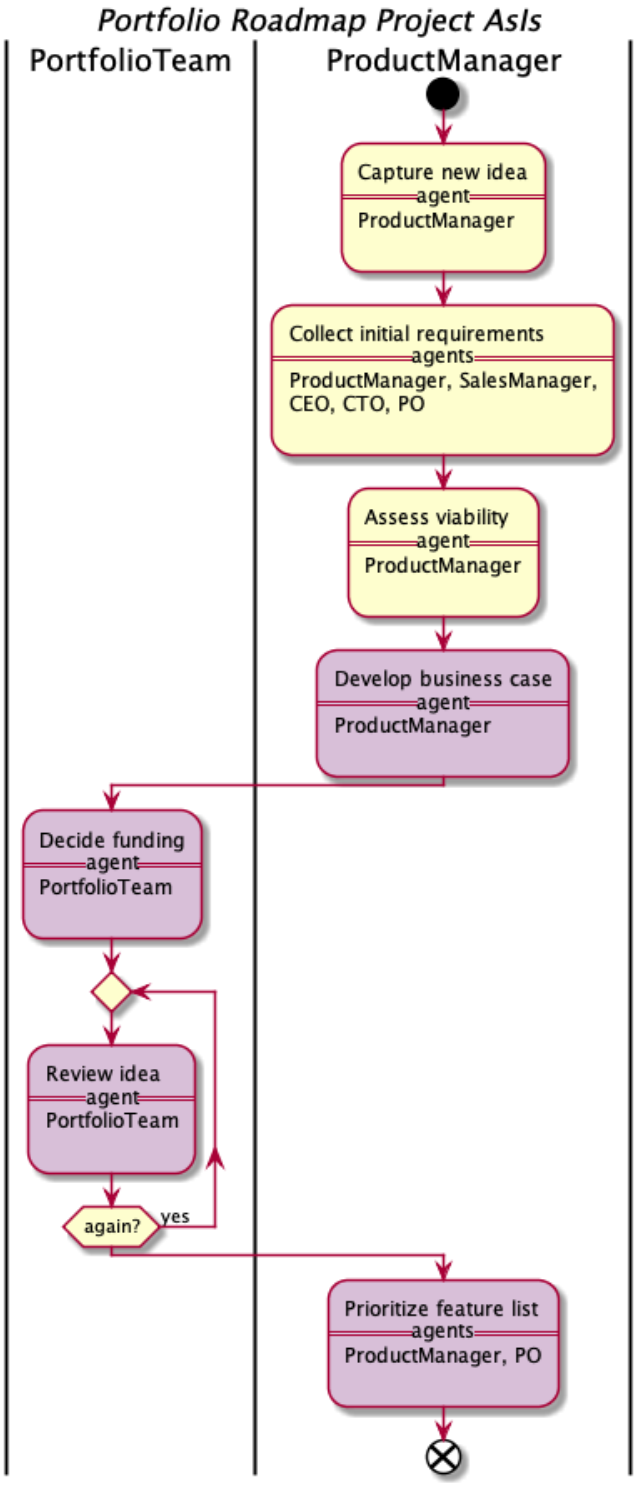

Figure I.35: S4PO80 As-is context.

I.3.17.4 To-be Context

New actions are shown in green.

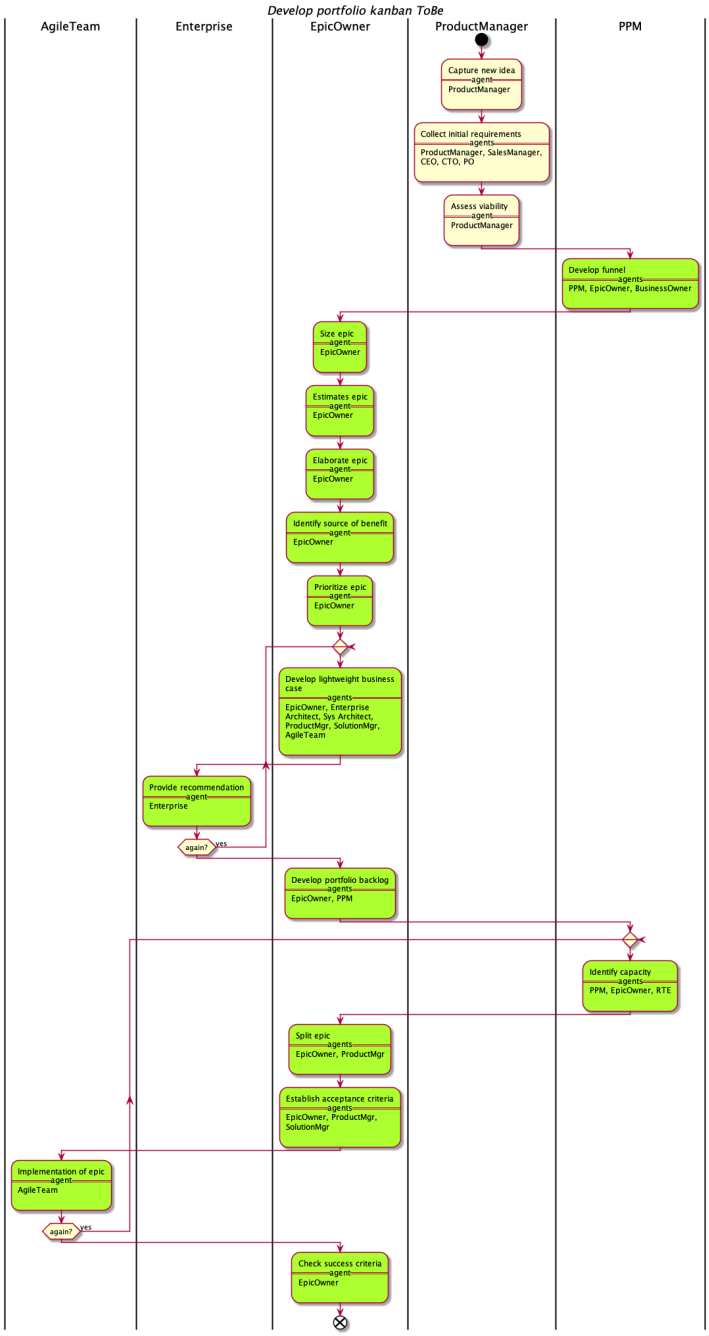

Figure I.36: S4PO80 To-be context.

I.3.17.5 Stakeholders

Sales Manager, CEO, CTO, PO, PPM, Epic Owner, Business Owner, Enterprise Architect, Sys Architect, Product Mgr, Solution Mgr, Agile Team, Enterprise, and RTE.

I.3.17.6 Entry and Exit Criteria

- Entry Criteria: New idea
- Exit Criteria: Done Epic

I.3.17.7 Implementation details

Table I.21: Implementation details for practice S4PO80 – Develop portfolio kanban.

Step	Description	Role(s)
Capture new idea	The Product Manager maintains a funnel of possible new features and new integrations with other products, that are submitted by sales people, management, or developers. These may be ideas learned from customers, at trade shows, or as a result of working on the product. It's possible for anyone to add an idea to the funnel, but in practice the Product Manager is the one who does this, based on input from other parts of the organization.	Product Mgr
Collect initial requirements	The Product Manager collects and records initial requirements from the CEO, CTO, a sales person, or anyone from the company who can contribute. This starts with a very high level idea, then proceeds to initial review to see if it is viable.	Product Mgr, Sales Manager, CEO, CTO, PO
Assess viability	The Product Manager does an initial assessment of the viability of a new idea.	Product Mgr

Continued on next page

Table I.21 – *Continued from previous page*

Step	Description	Role(s)
Develop funnel	The funnel queue is the capture queue where all ideas come from different sources, such as the portfolio strategic themes. At this stage an epic does not need to have any business case or estimates. It can be stated as a short keyword or phrase. All ideas are captured for consideration and can be represented in different formats, i.e. a document, spreadsheet, or visual system on the wall. Program Portfolio Management (PPM) typically discusses funnel epics on a periodic cadence, and epics that meet the decision criteria are moved to the review queue.	PPM, Epic Owner, Business Owner
Size epic	The Epic Owner roughly sizes the epic.	Epic Owner
Estimates epic	The Epic Owner estimates size of the epic in terms of epic value.	Epic Owner
Elaborate epic	The Epic Owner elaborates the epic using the epic value statement format. This value stream statement can be used to capture, organize, and communicate key information about an epic. There are two parts to the value statement template: 1) Forward-Looking position statement: for (customer), who (do something), the (solution), is a (something–the how), that (provides this value), unlike (competitor, current solution, or non-existing solution), our solution (does something better–the why) 2) Scope: Success criteria, In scope, Out of scope, and NFRs.	Epic Owner
Identify source of benefit	The Epic Owner identifies the business benefits from epic value statement.	Epic Owner
Prioritize epic	The Epic Owner prioritizes the epic based on the needs or business benefits.	Epic Owner

Continued on next page

Table I.21 – *Continued from previous page*

Step	Description	Role(s)
Develop lightweight business case	For ideas that seem viable, the Product Manager develop a business case for review by the Portfolio Team. The business case may be as simple as an email message, or an elaborate presentation to the Portfolio Team. An Epic Owner takes responsibility for the ongoing work. Then an action collaboration is initiated among Enterprise Architects, System Architects, Agile Teams, Product and Solution Management, and key stakeholders on the agile release train. Through this collaboration the key stakeholders also explore solution, design, and implementation alternatives. The options for internal and (or) outsourcing development are considered.	Epic Owner, Enterprise Architect, Sys Architect, Product Mgr, Solution Mgr, Agile Team
Provide recommendation	The Enterprise provides a go or no-go recommendation based on the business case and resource availability (i.e. business analyst, development team, and enterprise architect.)	Enterprise
Develop portfolio backlog	Epic with a 'go' decision is kept in the portfolio backlog and reviewed on a periodic basis. The queue in the portfolio backlog represents a low-cost holding pattern for upcoming implementation work. Epics are moved to the implementing queue when there is sufficient capacity from one or more value stream or Agile Release Trains.	Epic Owner, PPM
Identify capacity	PPM or Epic Owner identifies the capacity in the value stream. As capacity becomes available, epics are pulled into the relevant value stream or program kanban.	PPM, Epic Owner, RTE
Split epic	The epics are split into capabilities and features.	Epic Owner, Product Mgr
Establish acceptance criteria	Epic Owner, Solution Manager, and Product Manager develop acceptance criteria for capabilities and features.	Epic Owner, Product Mgr, Solution Mgr
Implementation of epic	Finally, the Team performs the actual development.	Agile Team

Continued on next page

Table I.21 – *Continued from previous page*

Step	Description	Role(s)
Check success criteria	The Epic Owner checks the implemented epic against epic success criteria. The epic is considered done when it has met all its success criteria. However, due to scope of epics, completion to the original intent is not always the desired case. At this stage, some identified capabilities and features might be discarded. Regardless, the epic reaches a done state.	Epic Owner

I.4 Roadmap F2

Undertake training onsite and face-to-face so team members can be directly assessed and training provision tailored to their specific requirements.

I.4.1 GTM Context

Fig. I.37 provides GTM context of the recommendation F2.

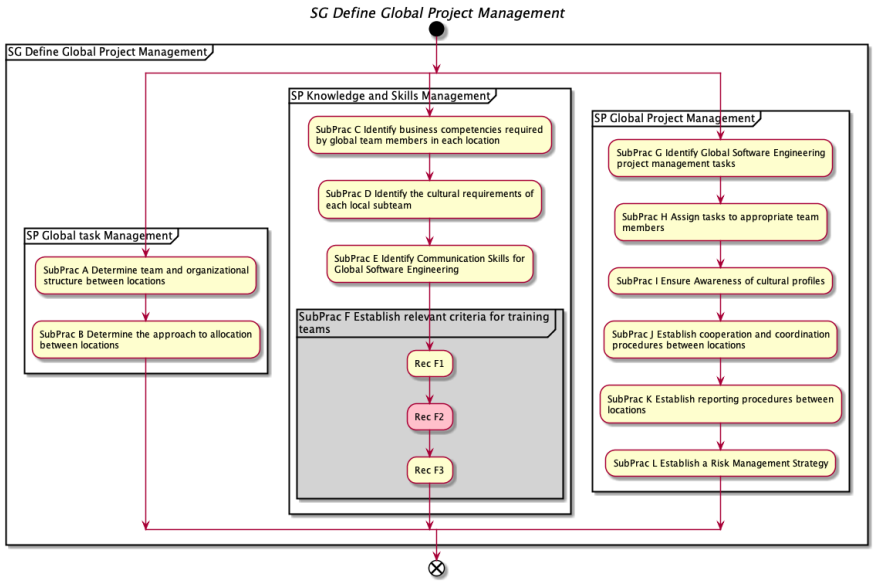

Figure I.37: F2 GTM context.

I.4.2 Why implement this recommendation?

From observations and interviews:

- Lack of training could demotivate team members
- Proper training required to get to know about the role and responsibilities
- Lack of technical knowledge could cause of wrong requirements.

I.4.3 Understand Pre and Post PI planning (S4VS141)

I.4.3.1 Purpose

Pre- and Post-PI planning meetings "allow ART and Suppliers in large value stream to build an aligned plan for the next Program Increment (PI) (Leffingwell, 2015)".

I.4.3.2 Background

In SAFe, PI planning is a significant event that requires preparation, coordination, and communication; in preparation for this event, each stakeholder is notified in advance. The success of this event depends on organization readiness, content readiness, and facility readiness. Therefore, before planning, programs must have a strategy alignment among participants. It should provide a clear vision and context, so that the correct stakeholder can participate. And finally, there is a need to provide physical space and technical infrastructure to support a large number of attendees, especially if there are remote participants (Razzak et al., 2018).

I.4.3.3 As-is & To-be Context

There is no as-is context for this practice. In the to-be model, new actions are shown in green.

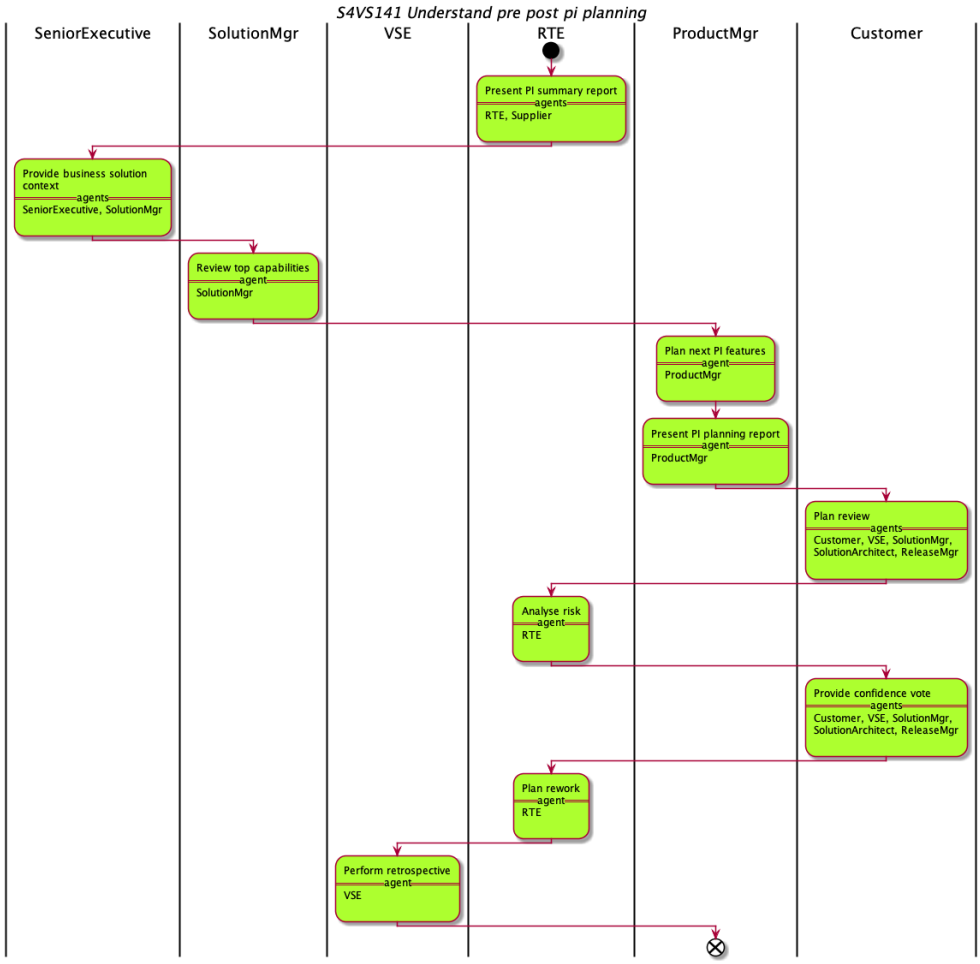

Figure I.38: S4VS141 To-be context.

I.4.3.4 Stakeholders

RTE, Supplier, Senior Executive, Solution Mgr, Product Mgr, Customer, VSE, Solution Architect, Release Mgr.

I.4.3.5 Entry and Exit Criteria

- Entry Criteria: PI report
- Exit Criteria: Post PI report

I.4.3.6 Implementation details

Table I.22: Implementation details for practice S4VS141 – Understand Pre and Post PI planning.

Step	Description	Role(s)
Present PI summary report	Each agile release train (ART) and supplier presents a brief report of accomplishments of the previous program increment (PI). This report provides the context of what has been achieved for the planning process.	RTE, Supplier
Provide business solution context	A senior executive presents a briefing about the current state of the value stream and program portfolio. Solution management presents the current value stream vision and highlights changes from the previous PI. Solution management also presents the roadmap for the upcoming three PIs, as well as milestones that fall during that period to ensure that they are known and addressed.	Senior Executive, Solution Mgr
Review top capabilities	Solution management reviews the top capabilities for the upcoming PI.	Solution Mgr
Plan next PI features	ART's product management presents the program backlog that they prepared for the upcoming PI and discuss dependencies with the other trains.	Product Mgr
Present PI planning report	ART's product management present the plans devised by their individual PI planning meetings, explaining the PI objectives and when each is anticipated to be available. RTEs fill out their ART's row of the value stream board and discuss dependencies with other ARTs or with suppliers.	Product Mgr
Plan review	All participants review the complete plan.	Customer, VSE, Solution Mgr, Solution Architect, Release Mgr

Continued on next page

Table I.22 – *Continued from previous page*

Step	Description	Role(s)
Analyse risk	During PI planning, RTE identifies critical risks and impediments that could affect the ability to meet PI objectives. Relevant risks are addressed in a broader, value stream context in front of the full group. RTE categorized risks into one of the following groups and addressed in a clear, honest, and viable manner: (1) Resolved- The group agrees that the issue is no longer a concern (2) Owned- The item cannot be resolved in the meeting, so someone takes ownership (3) Accepted- Some risks are facts or potential occurrences that simply must be understood and accepted (4) Mitigated- The group can identify a plan to mitigate the impact of an item.	RTE
Provide confidence vote	Once all risks have been addressed, the group votes on its confidence in meeting the value stream PI objectives. The Team conducts a 'five-of-five vote.' If the average is three or four fingers, then management should accept the commitment. If the average is fewer than three fingers, then planning adjustments are made, and plans are reworked. Any person voting two fingers or fewer should be given time to voice his or her concern, which might add to the list of risks.	Customer, VSE, Solution Mgr, Solution Architect, Release Mgr
Plan rework	If necessary, the group reworks its plans as long as it takes for commitment to be reached. This could cascade into the follow-up meeting in the ARTs, as teams will need to be involved in any change to the plans.	RTE
Perform retrospective	Finally, the value stream engineer leads a brief meeting retrospective to capture what went well, what did not, and what could be done better next time. Following this, the next steps are discussed, including capturing objectives, use of project management tooling, and finalizing the schedule of upcoming key activities and events.	VSE

I.5 Roadmap G5

Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global change management agent.

I.5.1 GTM Context

Fig. I.39 provides GTM context of the recommendation G5.

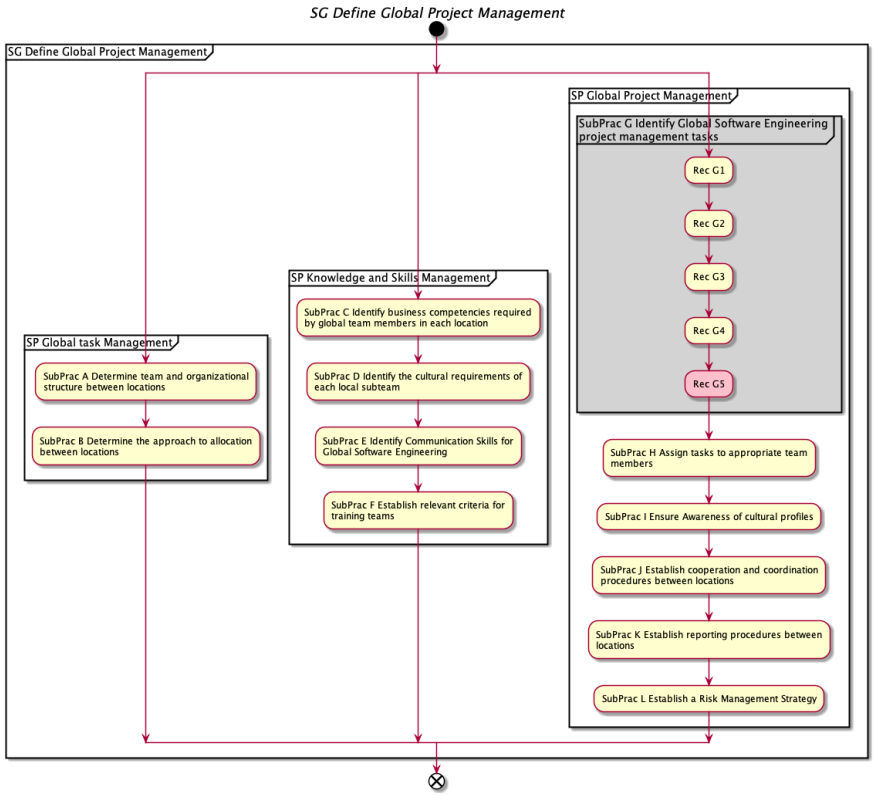

Figure I.39: G5 GTM context.

I.5.2 Why implement this recommendation?

From observations and interviews:

- Joint roles, responsibilities, and dependencies, mean that information flow is delayed when working across geographic distance.

- Geographic distance can create communication problems and breakdown in support even if teams are located in the same timezone and country.
- Geographic separation can leave the customer without support.
- Poor streamlining of process leads to delays; processes that worked in a collocated setting do not always work when distributed, as in the case of locking the development unit, and unlocking when developer has finished.
- Multiple versions of the build machines can lead to confusion and delays, as developers are uncertain which machine to use and which is available.

I.5.3 Perform retrospective at the end of each iteration (S4T3.4)

I.5.3.1 Purpose

The purpose of the sprint retrospective is to "discuss team practices and identify ways to improve (Leffingwell, 2015)."

I.5.3.2 Background

The Scrum team performs a sprint retrospective at the end of each sprint, which is time-boxed to an hour or less. During the sprint retrospective, the team reviews what worked well, what did not work well, and what can be done better in the next Sprint.

The team also reviews quantitative metrics (such as velocity) that they are using to measure performance, as well as qualitative aspects by identifying challenges from the last iteration, to perform root cause analysis. They then discuss corrective actions and document these as improvement stories which are placed on the team backlog.

I.5.3.3 As-is Context

Existing AsIs actions do not change.

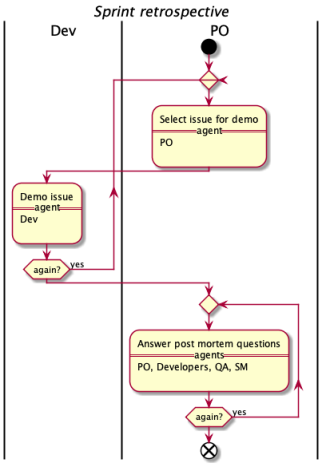

Figure I.40: S4T3.4 As-is context.

I.5.3.4 To-be Context

New actions are shown in green.

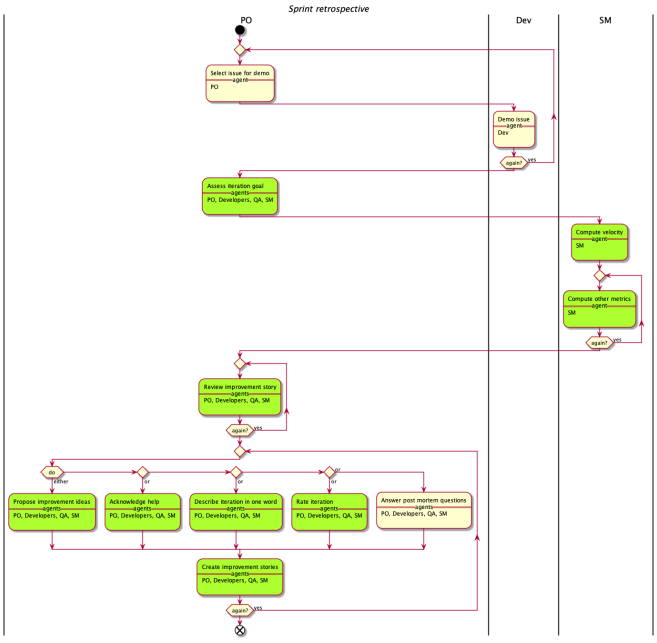

Figure I.41: S4T3.4 To-be context.

I.5.3.5 Stakeholders

PO, SM, Dev, QA, Architect, Sr Dev, Support Tech, ExternalQA, and Developers.

I.5.3.6 Entry and Exit Criteria

- Entry Criteria: PI Objectives, Booked holidays, Team availability
- Exit Criteria: Improvement Story

I.5.3.7 Implementation details

Table I.23: Implementation details for practice S4T3.4 – Perform retrospective at the end of each iteration.

Step	Description	Role(s)
Select issue for demo	The Product Owner selects issue for demonstration in retrospective.	PO
Demo issue	Developers demonstrate issue.	Dev
Assess iteration goal	The Team assesses whether they met the iteration goal.	PO, Developers, QA, SM
Compute velocity	The Scrum Master computes velocity as a function of story points completed and ideal effort available.	SM
Compute other metrics	The Team collects any other metrics they have agreed to analyze, such as unit test coverage, defect density, new test cases created or automated, fraction of tests automated, number of refactoring, etc. Agile Teams collect and apply other Iteration Metrics for visibility and to help with process improvement.	SM
Review improvement story	The Team reviews the improvement stories from the previous iteration. Improvement stories that were accomplished are marked as such; for those not accomplished, the team must decide what to do about them.	PO, Developers, QA, SM

Continued on next page

Table I.23 – *Continued from previous page*

Step	Description	Role(s)
Propose improvement ideas	Individual Team Members write ideas on post-its, which are then analyzed by the team for patterns.	PO, Developers, QA, SM
Acknowledge help	Team Members acknowledge who has helped them in the previous iteration.	PO, Developers, QA, SM
Describe iteration in one word	Each Team Member states one word to describe the iteration.	PO, Developers, QA, SM
Rate iteration	Each Team Member rates the iteration on a five-point scale. The team then brainstorms ways to make the next iteration score at the top of the scale.	PO, Developers, QA, SM
Answer post mortem questions	Each Team Member provides answers to three questions: what went well, what did not go so well, and what can be done better next time.	PO, Developers, QA, SM
Create improvement stories	The Team develops improvement stories by assessing the iteration. Using the outputs from the above actions, the team develops one or two improvement stories to add to the team backlog.	PO, Developers, QA, SM

I.5.4 Perform team demo (S4T140)

I.5.4.1 Purpose

The purpose of the team demo is to measure the team's progress by showing working stories to the Product Owner and other stakeholders, and get their feedback.

I.5.4.2 Background

The Team Demo is the traditional Scrum-prescribed ceremony whereby the team reviews the increment that results from the iteration. Planning and presenting an effective team demo requires some work on the part of the teams, but without it they will not have the fast feedback they need to build the right thing. This demo is a one- to two-hour demonstration of new functionality.

I.5.4.3 As-is Context

This practice is already performed as shown.

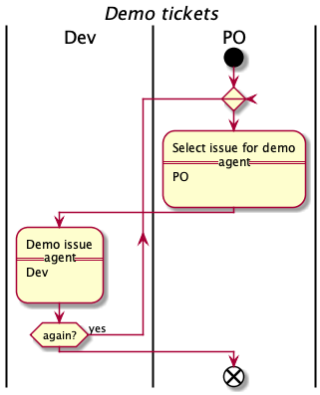

Figure I.42: S4T140 As-is context

I.5.4.4 To-be Context

This practice is already performed as shown.

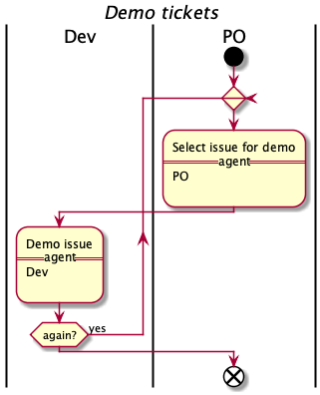

Figure I.43: S4T140 To-be context

I.5.4.5 Stakeholders

PO and Dev.

Entry and Exit Criteria

- Entry Criteria: Ticket status (Ticket to demo)

- Exit Criteria: ticket demoed

I.5.4.6 Implementation details

Table I.24: Implementation details for practice S4T140 – Perform team demo.

Step	Description	Role(s)
Select issue for demo	The Product Owner selects issue for demonstration in retrospective.	PO
Demo issue	Developers demonstrate issue.	Dev

I.5.5 Participate in community of practice (S4T302)

I.5.5.1 Purpose

The purpose of a CoP is to provide a way for practitioners to share tips and best practices, ask questions of their colleagues, and provide support for each other (Leffingwell, 2007, 2015).

I.5.5.2 Background

Communities of Practice (CoPs) are organized groups of people who have a common interest in a specific technical or business domain. They collaborate regularly to share information, improve their skills, and actively work on advancing the general knowledge of the domain (Leffingwell, 2007, 2015).

Healthy CoPs have a culture built on professional networking, personal relationships, shared knowledge, and common skills. Combined with voluntary participation, CoPs provide knowledge workers with opportunities to experience autonomy, mastery, and purpose beyond their daily tasks on an Agile Release Train (ART).

CoPs enable practitioners to exchange knowledge and skills with people across the entire organization. This open membership offers access to a wide range of expertise to help with technical challenges, fuel continuous improvement and allows more meaningful contributions to the larger goals of the Enterprise. The result is

that organizations benefit from rapid problem-solving, improved quality, cooperation across multiple domains, and increased retention of top talent (Leffingwell, 2007, 2015).

I.5.5.3 As-is and To-be Context

I have not observed any as is process for this practice.

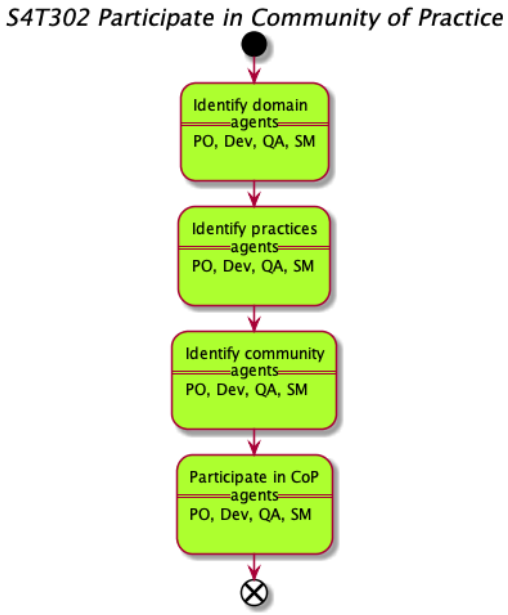

Figure I.44: S4T302 To-be context.

I.5.5.4 Stakeholders

Product Owner, Developer, Quality Assurance, and Scrum Master.

I.5.5.5 Entry and Exit Criteria

- Entry Criteria: CoP list available
- Exit Criteria: Shared Explicit Knowledge created or updated

I.5.5.6 Implementation details

Table I.25: Implementation details for practice S4T302 – Participate in community of practice.

Step	Description	Role(s)
Identify domain	Identify a domain that cares about your interest.	PO, Dev, QA, SM
Identify practices	Identify a shared body of knowledge, experiences, and techniques.	PO, Dev, QA, SM
Identify community	Identify a self-selected group of individuals who care enough about the topic to participate in regular interactions.	PO, Dev, QA, SM
Participate in CoP	Access and share knowledge to promote learning in a particular area.	PO, Dev, QA, SM

I.5.6 Develop program performance metrics(S4PR31)

I.5.6.1 Purpose

The purpose of this practice is to aggregate the team's metrics at the end of the Program Increment (PI), i.e., functionality (velocity, predictability, features, enablers, stories, etc) and quality (test, automation, coverage, defects, performance, etc).

I.5.6.2 Background

Program performance metrics aggregate the team performance in the Agile Release Train (ART). The aggregated visualization contains team productivity, quality, and story points completed. Also includes velocity, story points, and product quality.

I.5.6.3 As-is Context

I.5.6.4 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

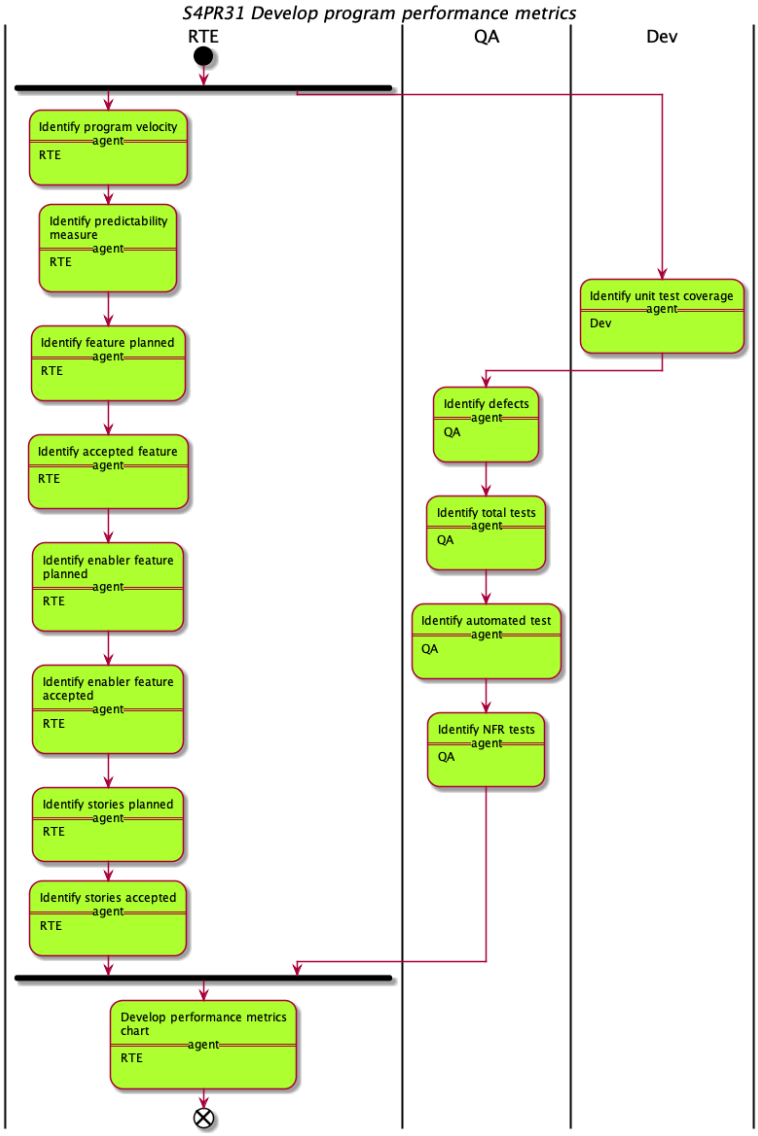

Figure I.45: S4PR31 To-be context.

I.5.6.5 Stakeholders

RTE, Dev, QA

I.5.6.6 Entry and Exit Criteria

- Entry Criteria: ART, Code, Previous program increment, Resource availability.

- Exit Criteria: Performance metrics chart

I.5.6.7 Implementation details

Table I.26: Implementation details for practice S4PR31 – Develop program performance metrics.

Step	Description	Role(s)
Identify program velocity	Release train engineer (RTE) identifies program velocity based on previous program increment and resource availability.	RTE
Identify predictability measure	To assess the overall predictability of the release train, the 'Team PI Performance Report' is aggregated. All teams on the train calculate the program predictability measure by comparing actual business value achieved to planned business value.	RTE
Identify feature planned	RTE identifies a number of features planned for upcoming program increment from program kanban.	RTE
Identify accepted feature	RTE identifies a number of features accepted within a program increment from program kanban.	RTE
Identify enabler feature planned	RTE identifies a number of enablers planned for upcoming program increment from program kanban.	RTE
Identify enabler feature accepted	RTE identifies a number of enablers accepted within a program increment from program kanban.	RTE
Identify stories planned	RTE identifies a number of stories planned for upcoming program increment from program kanban.	RTE
Identify stories accepted	RTE identifies a number of stories accepted within a program increment from program kanban.	RTE
Identify unit test coverage	Developer provides code coverage report. The results table shows the percentage of the code that was run in each assembly, class, and method. Besides, the source editor shows which code has been tested.	Dev
Identify defects	QA provides a list of defects identified within a program increment.	QA
Identify total tests	QA identifies the number of tests performed within a program increment.	QA
Identify automated test	QA provides a report by compiling percentage of automated test coverage.	QA
Identify NFR tests	QA provides a test report of NFR.	QA

Continued on next page

Table I.26 – *Continued from previous page*

Step	Description	Role(s)
Develop performance metrics chart	RTE develop agile release train's performance metrics chart based on the calculated functionality and quality at the end of each PI.	RTE

I.5.7 Release value (S4PR50)

I.5.7.1 Purpose

Every release of the product delivers value to the customer. Moreover, a solution cannot be fully validated until it is released to the customer environment.

I.5.7.2 Background

One of the goals of the Lean-Agile enterprise is to release the product more frequently. This provides meaningful feedback about efficacy, deployability, and usability in the development environment. In SAFe, the Release Train Engineer (RTE) identifies a possible release (e.g., a batch) to be tested in the end-user environment before deployment. Then, the QA tests the batch or integrated solution provided by the Developer.

I.5.7.3 As-is Context

Actions altered by this practice are shown in purple.

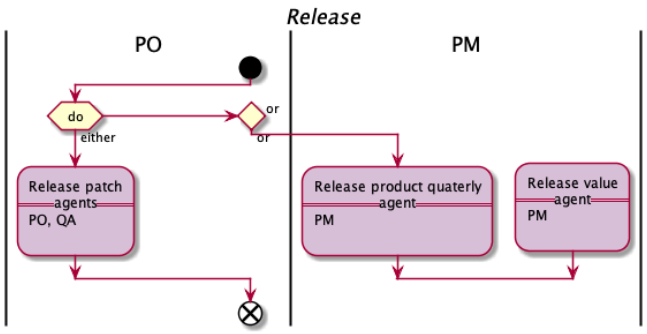

Figure I.46: S4PR50 As-is context.

I.5.7.4 To-be Context

Identify release and release value (from AsIs) is combined into one action.

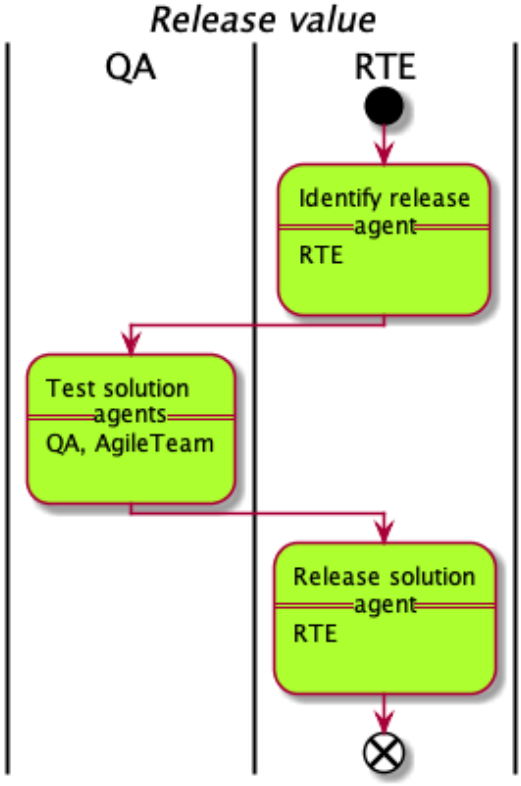

Figure I.47: S4PR50 To-be context.

I.5.7.5 Stakeholders

RTE, QA, Agile Team.

I.5.7.6 Entry and Exit Criteria

- Entry Criteria: Existing System
- Exit Criteria: Feedback

I.5.7.7 Implementation details

Table I.27: Implementation details for practice S4PR50 – Release value.

Step	Description	Role(s)
Identify release	RTE identifies possible release (e.g., batch) to be tested in the end-user environment before deployment. A release is a collection of files that can be source code, data base, or an environment. It's a collection of files that goes to production and deliver typically value -new value for customers. Then there is a release for customers. Specific customers and a release for geographical areas.	RTE
Test solution	QA tests the batch or integrated solution provided by the developer.	QA, Agile Team
Release solution	Releasing a tested solution to the end-user environment provides meaningful feedback about efficacy, deployment, and usability.	RTE

I.5.8 Refine program backlog (S4PR77)

I.5.8.1 Purpose

The purpose of the backlog refinement or backlog grooming is to keep the backlog clean and orderly.

I.5.8.2 Background

In SAFe, there are four types of backlogs: Team backlog, Program backlog, Value stream backlog, and Portfolio backlog. In the program backlog refinement meeting, the Product Manager and Product Owner discuss the top items on the project or program backlog. At this level, as a preparation for the PI planning meeting the Product Management team also refines the backlog (Leffingwell, 2007, 2015).

I.5.8.3 As-is Context

Actions altered by this practice are shown in purple.

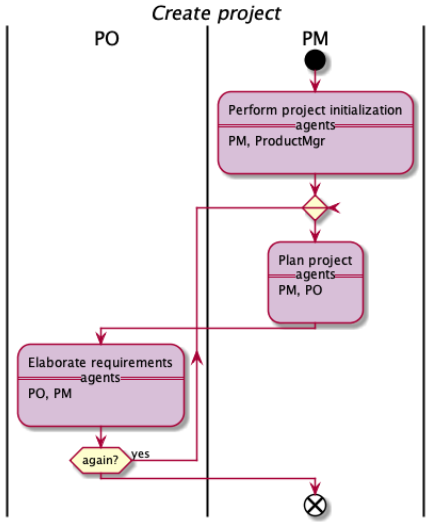

Figure I.48: S4PR77 As-is context.

I.5.8.4 To-be Context

New actions are shown in green.

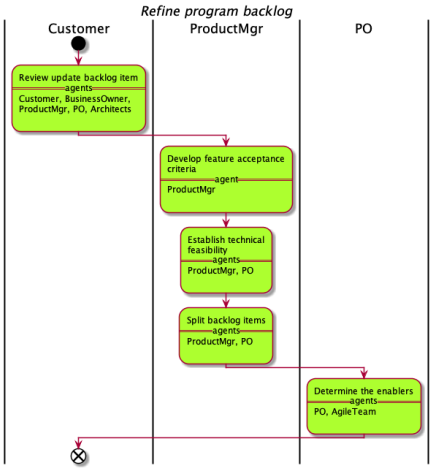

Figure I.49: S4PR77 To-be context.

I.5.8.5 Stakeholders

Customer, Business Owner, Product Mgr, PO, Architects, and Agile Team.

I.5.8.6 Entry and Exit Criteria

- Entry Criteria: Epic, Portfolio backlog.
- Exit Criteria: Product backlog.

I.5.8.7 Implementation details

Table I.28: Implementation details for practice S4PR77 – Refine program backlog.

Step	Description	Role(s)
Review update backlog item	The Product Manager reviews and updates backlog items in preparation for the next PI planning.	Customer, Business Owner, Product Mgr, PO, Architects
Develop feature acceptance criteria	In preparation of the PI planning event, the Product Manager also develops acceptance criteria for each feature in the program backlog.	Product Mgr
Establish technical feasibility	Product Manager and PO work with the team to establish technical feasibility and scope estimates.	Product Mgr, PO
Split backlog items	The Product Management team analyzes ways to split backlog items into smaller chunks of incremental value. If the feature is big, then the team split that feature into multiple features otherwise a story.	Product Mgr, PO
Determine the enablers	PO and Agile Team Determine the Enablers spawned by new features and capabilities, and establish their capacity allocation.	PO, Agile Team

I.5.9 Create and manage features (S4PR114)

I.5.9.1 Purpose

The purpose of this practice is to create the features which address the needs of the user.

I.5.9.2 Background

There are two types of features: Business and Enabler. Features are developed through the program Kanban in-collaboration with the Product Manager, Product Owner, and other stakeholders. However, there are some features that arise as a result of decomposition of epics (from the Portfolio level). Each feature includes a statement of benefits and acceptance criteria. The enabler feature provides the infrastructure needed to develop, test, and integrate the initiative. These types of features are generally created by the architects and maintained in the program backlog alongside business features (Leffingwell, 2007, 2015).

I.5.9.3 As-is Context

Actions altered by this practice are shown in purple.

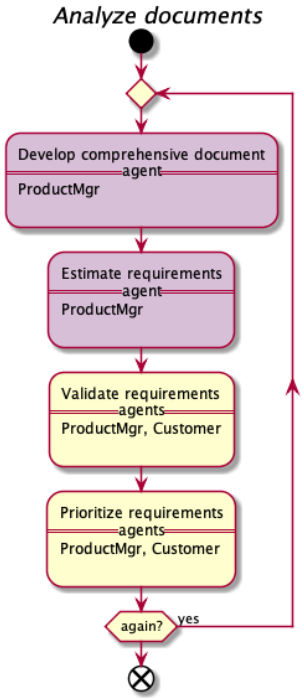

Figure I.50: S4PR114 As-is context.

I.5.9.4 To-be Context

New actions are shown in green and two as-is actions merged with to-be actions.

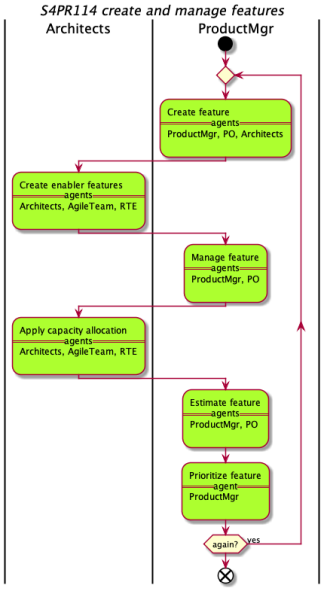

Figure I.51: S4PR114 To-be context.

I.5.9.5 Stakeholders

Product Mgr, PO, Architects, Agile Team, and RTE.

I.5.9.6 Entry and Exit Criteria

- Entry Criteria: Epic, Portfolio backlog
- Exit Criteria: Feature prioritized

I.5.9.7 Implementation details

Table I.29: Implementation details for practice S4PR114 – Create and manage features.

Step	Description	Role(s)
Create feature	In collaboration with Product Owners and other key stakeholders, features are created by Product Managers in the local context of an ART. Enabler features pave the Architectural Runway, support exploration, or may provide the infrastructure needed to develop, test, and integrate the initiative. Enabler features are generally created by Architects or Engineers and maintained in the program backlog alongside business features.	Product Mgr, PO, Architects
Create enabler features	Architects or Engineers create enabler features to pave the architectural runway, support exploration, or to provide the infrastructure needed to develop, test, and integrate the initiative. Enabler features are maintained in the program backlog alongside business features.	Architects, Agile Team, RTE
Manage feature	Through a program backlog Product Manager manages all features.	Product Mgr, PO
Apply capacity allocation	Apply capacity allocation for the enabler to work as a whole or to differentiate between various types of enablers.	Architects, Agile Team, RTE
Estimate feature	Feature estimation usually occurs in the 'refinement' state of the Program Kanban and relies on normalized estimation techniques, equivalent to the approach used by Agile Teams for estimating stories. Feature estimation at this point, however, does not require full breakdown into stories or involve all the teams that possibly will be included in feature development. Instead, select subject matter experts may be engaged in basic exploration and sizing.	Product Mgr, PO
Prioritize feature	Prioritize features in the program backlog using weighted shortest job first (WSJF).	Product Mgr

I.5.10 Perform system demo (S4PR137)

I.5.10.1 Purpose

The purpose of the System Demo is to test and evaluate the full system that the Agile Release Train (ART) is working on, and to get feedback from the primary stakeholders, including business owners, executive sponsors, other agile teams, development management, customers, and customer proxies.

I.5.10.2 Background

In addition to serving as a test of the full system that the Agile Release Train (ART) is working on, it also provides a way for the primary stakeholders – including business owners, executive sponsors, other agile teams, development management, customers, and customer proxies – to provide feedback on the evolving product.

The System Demo occurs at the end of every Iteration. It provides an integrated, aggregated view of the new Features that have been delivered by all the teams on the train in the most recent iteration (Leffingwell, 2007, 2015).

The system demo is the integrated demo of the work of all teams on the train. The system demo does not replace each team's local Team Demo, which also occurs at the end of every iteration (Leffingwell, 2007, 2015).

I.5.10.3 As-is & To-be Context

I have not observed any as-is for this practice. New actions are shown in green.

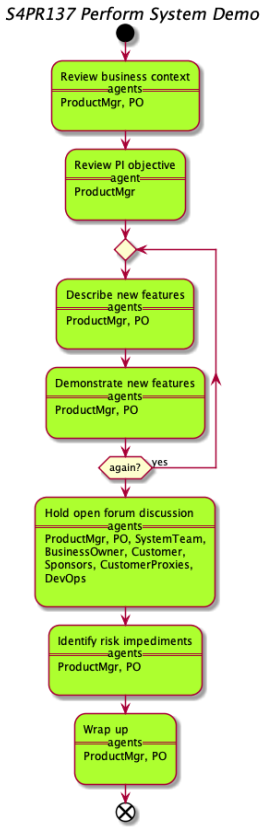

Figure I.52: S4PR137 To-be context.

I.5.10.4 Stakeholders

Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps

I.5.10.5 Entry and Exit Criteria

- Entry Criteria: Requirement
- Exit Criteria: Action items, Feedback, Progress report

I.5.10.6 Implementation details

Table I.30: Implementation details for practice S4PR137 – Perform system demo.

Step	Description	Role(s)
Review business context	The Product Manager and Product Owner review initial business context by going through each requirement of each integrated feature.	Product Mgr, PO
Review PI objective	During the system demo, the Product Manager briefly reviews the PI objectives.	Product Mgr
Describe new features	The Product Manager and PO describe integrated and aggregated view of the new features that have been delivered by the all teams on the train in the most recent iteration.	Product Mgr, PO
Demonstrate new features	The Product Manager and PO demonstrate each new feature in an end-to-end use case.	Product Mgr, PO
Hold open forum discussion	At the end of the system demo, the Product Manager opens the forum for questions and comments.	Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps
Identify risk impediments	Based on the feedback from the open forum, the Product Manager and PO note the current risks and impediments.	Product Mgr, PO
Wrap up	The Product Manager and PO wrap up by summarizing progress, feedback, and action items.	Product Mgr, PO

I.5.11 Perform inspect and adapt (I&A) workshop (S4PR152)

I.5.11.1 Purpose

The purpose of the Inspect and Adapt workshop is to create a mindset of continuous improvement requires you to spend time thinking about "what could be done better (Leffingwell, 2015)." The Inspect & Adapt workshop provides this opportunity.

I.5.11.2 Background

The Inspect and Adapt workshop is held at the end of each Program Increment. It is important that this workshop is held as it allows time to reflect on the execution and results of the previous PI, and build improvement backlog items for the next PI. This workshop can be held at both the Program Level and Value Stream Level (Leffingwell, 2007).

The I&A workshop has three steps:

1. PI System Demo (we have covered this in practice S4PR137).
2. Quantitative measurement.
3. Retrospective and problem-solving workshop.

I.5.11.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

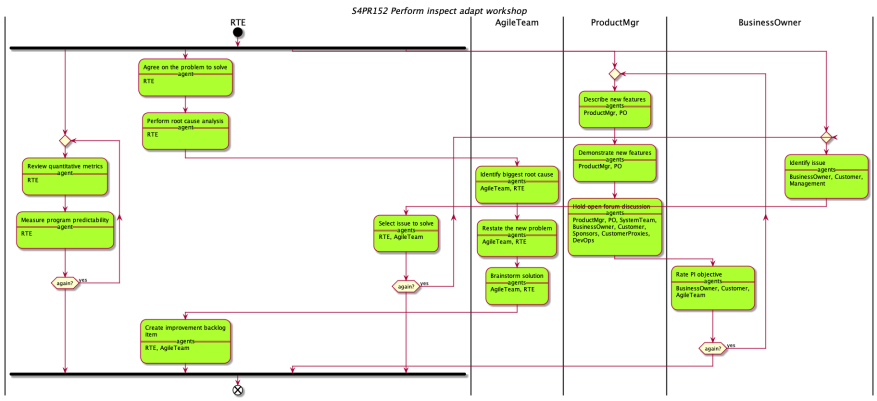

Figure I.53: S4PR152 To-be context.

I.5.11.4 Stakeholders

RTE, Agile Team, Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps, and Management.

I.5.11.5 Entry and Exit Criteria

- Entry Criteria: Candidate issue, Issues, Iteration report, Metric agreed, Program predictability, and Team PI performance report.
- Exit Criteria: Business value, Candidate issue, Program predictability, Team backlog, Program backlog.

I.5.11.6 Implementation details

Table I.31: Implementation details for practice S4PR152 – Perform inspect and adapt (I&A) workshop.

Step	Description	Role(s)
Review quantitative metrics	In preparation for this, the Release Train Engineer (RTE) and the Solution Train Engineer are often responsible for gathering the information, analyzing it to identify potential issues, and facilitate the presentation of the findings to the ART.	RTE
Measure program predictability	The RTE also measures the program predictability of each team using planned vs actual business value (in the form of PI performance report.)	RTE
Agree on the problem to solve	At this point, the teams have a self-selected problem they want to work on. The team spend a few minutes stating the problem, thinking about the what, where, when, and the impact of succinctly as they can.	RTE
Perform root cause analysis	SAFe recommend using Fishbone Diagrams tools to perform root cause analysis. Candidate issues are identified and then grouped into major categories as boned off the main bone. Team preload the main bones with the categories: People, Process, Tools, Program, and Environment. Team members then brainstorm factors that they think contribute to the problem to be solved. Once an issue/cause is identified, its root cause is identified with the five whys technique.	RTE

Continued on next page

Table I.31 – *Continued from previous page*

Step	Description	Role(s)
Identify biggest root cause	Once all possible issues/causes have been identified, Team members then vote on the item they think is the biggest factor causing the end problem. The team creates a Pareto Chart which is a collective consensus on the largest root causes.	Agile Team, RTE
Restate the new problem	In this stage, the Team picks the largest cause from the list and restates it clearly as a problem.	Agile Team, RTE
Brainstorm solution	At this stage, the root cause will start to imply some potential solutions. During the brainstorming session, the Team applies the following rules: generate as many ideas as possible, do not criticize or debate, let the imagination soar, mutate and combine ideas.	Agile Team, RTE
Create improvement backlog item	The Team then votes on up to three most likely solutions. These will serve as improvement stories and features to be fed directly into the PI Planning session that follows. During that session, the Release Train Engineer helps to ensure that the relevant improvement stories are loaded onto the Iteration plans, thus ensuring that action will be taken and resources allocated, as with any other backlog item.	RTE, Agile Team
Describe new features	The Product Manager and PO describe integrated and aggregated view of the new features that have been delivered by the all teams on the train in the most recent iteration.	Product Mgr, PO
Demonstrate new features	The Product Manager and PO demonstrate each new feature in an end-to-end use case.	Product Mgr, PO
Hold open forum discussion	At the end of the system demo, the Product Manager opens the forum for questions and comments.	Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, Dev Ops

Continued on next page

Table I.31 – *Continued from previous page*

Step	Description	Role(s)
Rate PI objective	During the PI system demo, the business owners, customers, and the other vital stakeholders collaborate with each Agile Team to rate their actual business value achieved.	Business Owner, Customer, Agile Team
Identify issue	Initially, the Team identifies issues they would like to address, and team identify few significant problems that the team can potentially address.	Business Owner, Customer, Management
Select issue to solve	Based on the issues identified, the facilitator helps the group decide which issues they want to tackle. Each Team has a choice of resolving Team level or Program level problems. Then the candidate issues use in the problem-solving workshop.	RTE, Agile Team

I.5.12 Perform retrospective on the Agile Release Train (ART) (S4PR160)

I.5.12.1 Purpose

A retrospective at the Program level is a special meeting where the team gathers after completing an increment of work to inspect and adapt their methods and teamwork. This ceremony enables whole-team learning, acts as a catalyst for change, and generates action. Moreover, the team identifies improvements during their retrospective, and applies new practices in the next iteration (where applicable) (Derby et al., 2006).

I.5.12.2 Background

Retrospectives are a natural fit in an Agile work environment. Scrum and Crystal explicitly include an "inspect and adapt" cycle for the methods and teamwork along with mechanisms to examine and improve the product (Derby et al., 2006). While continuous builds, automated unit tests, and frequent demonstrations of working code are all ways to focus attention on product and allow the team to make adjustments, retrospectives focus attention on how the team does their work and interacts.

Iteration retrospectives focus on real problems that affect teams. During retrospectives, the team discovers real solutions that they can implement without waiting for management's permission.

Benefits from retrospectives, adopted from (Derby et al., 2006):

- Improved Productivity
- Improved Capability
- Improved Quality
- Increased Capacity

Beneficiary groups:

Software Development Team Member Suppose you are a team member of a software development team. If you are starting to see signs of interpersonal friction on the team, and some people you would like to retain on the team are dusting off their resumes. Then, you need to adapt your practices and ease the interpersonal tension before things get worse.

Team Lead If you are a team lead and heard about retrospectives but have never tried one. Then, this practice will helpful for you to revitalize your retrospectives.

I.5.12.3 As-is Context

Actions altered by this practice are shown in purple.

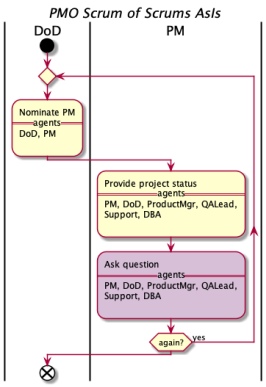

Figure I.54: S4PR160 As-is context.

I.5.12.4 To-be Context

New actions are shown in green.

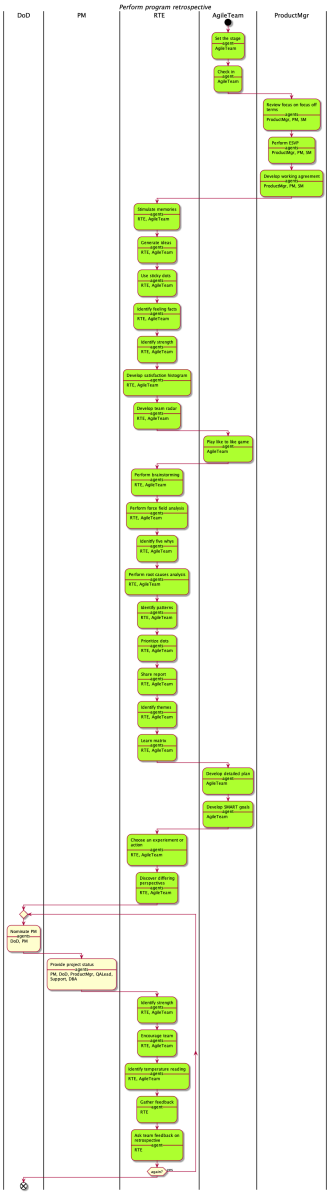

Figure I.55: S4PR160 To-be context.

I.5.12.5 Stakeholders

Agile Team, Product Mgr, PM, SM, RTE, DoD, QA Lead, Support, and DBA.

I.5.12.6 Entry and Exit Criteria

- Entry Criteria: Review goal, Schedule, Working agreement.
- Exit Criteria: Feedback

I.5.12.7 Implementation details

Table I.32: Implementation details for practice S4PR160 – Perform retrospective on the Agile Release Train (ART).

Step	Description	Role(s)
Set the stage	After the opening (review goal, schedule, and working agreements), this activity will help establish a mind-set of looking at the issues without assigning blame.	Agile Team
Check in	After welcoming the participants and reviewing the goal and agenda, the retrospective leader asks one brief question. For example, a) Ask one question that each person can answer with a word or short phrase, and b) Go around the room listening to each person's answer. You may thank each person (be sure to thank every person if you do). Refrain from offering evaluative comments such as 'good' or 'wonderful'. Each person answers in round-robin fashion.	Agile Team

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Review focus on focus off terms	After welcoming the participants and reviewing the goal and agenda, the retrospective leader describes productive and unproductive communication patterns. After describing those patterns, the participants discuss what they mean for the retrospective. Follow following steps: (1) Draw attention to the Focus On/Focus Off poster (2) Form small groups, with no more than four people per group. Ask each group to take one pair of words to define and describe. If there are more than four pairs or groups, it's OK if more than one group has the same pair of words. (3) Ask each group to discuss what their two words mean and what behaviors they represent. Have them describe the impact each would have on the Team and the retrospective. (4) Each group reports on their discussion to the whole team. (5) Ask people whether they are willing to stay in the left column	Product Mgr, PM, SM

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Perform ESVP	Each participant reports (anonymously) his or her attitude toward the retrospective as an Explorer, Shopper, Vacationer, or Prisoner (ESVP). The retrospective leader collects the results and creates a histogram to show the data, and then guides a discussion about what the results mean for the group. Follow following steps: 1) Explain that you are taking a poll to learn about how people view their participation in the retrospective. 2) Show the flip chart and define the terms: 2.1) Explorers are eager to discover new ideas and insights. They want to learn everything they can about the iteration–release– project. 2.2) Shoppers will look over all the available information, and will be happy to go home with one useful new idea. 2.3) Vacationers aren't interested in the work of the retrospective, but are happy to be away from the daily grind. They may pay attention some of the time, but they are mostly glad to be out of the office. 2.4) Prisoners feel that they've been forced to attend and would rather be doing something else. 3) Distribute slips of paper or small index cards for people to record their attitude toward learning in the retrospective today. Instruct people to fold their paper in half for privacy. 4) As people finish writing and folding, collect the slips and shuffle them. 5) Ask one of the participants to make tick marks on the histogram as you read the slips. After you read each slip, put them in your pocket. When you've read all the slips, tear them up and throw them away. Be conspicuous about this so people know that no one will try to identify who responded with what from the handwriting. 6) Ask the group, What do you make of this data? Then lead a brief discussion about how the attitudes in the room will affect the retrospective. 7) Debrief by asking How are these categories like our attitudes toward daily work?	Product Mgr, PM, SM

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Develop working agreement	Team members work together to generate ideas for effective behaviors at work then choose five to seven agreements to guide team interactions or processes. Detail steps: 1) Explain the activity 2) Form pairs or small groups, no more than four per group. 3) Ask each group to develop three to five working agreements that, if followed, would help the team have productive discussions during the retrospective. Remind the group that these aren't for business as usual-they should be new behaviors or ones that aren't yet normal behavior for the group. 4) In round-robin fashion, ask each group to report their most important agreement, and write it on a flip chart page. Write down the exact words used by the team member. Continue until you've captured all the unique proposed agreements. 5) Explain that for the retrospective, the group should choose three to seven agreements. Having more than seven is too hard to remember and follow. 6) If there are fewer than three proposed agreements, ask for clarifying questions for each agreement. When everyone understands, use a consensus thumb vote for each agreement. Thumb up = I agree. Thumb sideways = I will support the will of the group. Thumb down = I veto. 7) If there are more than seven proposed agreements, use dot voting to prioritize. Give each team member three color dots to vote with. Each person can put one dot on three separate items, or all dots on one. Use a consensus thumb vote to ratify the top five to seven vote getters.	Product Mgr, PM, SM

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Stimulate memories	Group members write cards to represent memorable, personally meaningful, or otherwise significant events during the iteration, release, or project and then post them in (roughly) chronological order. The retrospective leader supports the Team to discuss the events to understand facts and feelings during the iteration, release, or project. Steps: 1) Set up the activity by saying We'll fill in a timeline to create a fuller picture of this iteration/release/project. We want to see it from many perspectives. 2) Divide the team into small groups, with no more than five in a group. Keep people who worked closely with each other together (affinity groups). It's better to have two small groups representing one affinity than one big group. 3) Describe the process. Ask people to think back over the iteration/release/project and remember all the memorable, personally meaningful, or significant events and write them down, one per card or sticky note. 4) Monitor the level of activity as people start talking about events and writing cards. 5) When all the cards are posted, invite the team to walk by the timeline and see what others have posted. 6) Allow a break or take lunch before analyzing the timeline.	RTE, Agile Team

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Generate ideas	Generate ideas for actions or recommendations that uncover important topics about the project history. In the groups, each person has five minutes to brainstorm and write down ideas individually. At the end of five minutes, each person passes the paper to the person on his or her right. That person has five minutes to write down ideas that build on the ideas already written on the paper. Repeat until the paper returns to the original writer. Steps: 1) Set up the activity by saying In this activity, our goal is to generate as many ideas as we can about (topic). 2) Divide the Team into small groups, with no more than five in a group. Hand out paper for people to write on. Make sure each person has a pen or pencil. Remind people to write legibly so the next person can read the ideas. 3) Describe the process: In the first round, each person will have five minutes to write down ideas related to the topic. Aim for at least five ideas. In subsequent rounds, each person writes ideas that are sparked by the already written ideas or builds on them in some way. 4) Time the group. After five minutes, ring a chime and tell the group to pass the paper to the right. 5) Ask each person to read the ideas listed on the paper. 6) Debrief using these questions: 6.1) What did you notice while you wrote ideas? 6.2) What surprised you? What met your expectations? How? 6.3) What is missing from these lists? 6.4) What ideas and topics should we examine further?	RTE, Agile Team

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Use sticky dots	Show how people experienced events on the timeline. Team members use sticky dots to show events on the timeline where emotions ran high or low. Steps: 1) Set up the activity by saying Now that we've seen the facts, let's see how it was to be doing this work. 2) Provide each individual with dots in two colors. Start with seven to ten dots per person but have more available. Explain which color indicates high energy and which indicates low energy. 3) Ask each person to use the dots to show where energy was high and where energy was stalled, flagging, or at low point.	RTE, Agile Team
Identify feeling facts	Individuals use colored cards or sticky notes to describe times during the project where they were mad, sad, or glad. Steps: 1) Draw attention to three posters labeled Mad, Sad, Glad and sample color-coded cards. Put out colored cards/sticky notes where everyone can reach them. 2) Describe the process, and give the time limit. 3) Give notice when time is up, and invite people to post their cards on the appropriate poster. 4) Cluster the cards on each poster. Go to the first poster, pick one card, and read it. 5) Ask the group to name each cluster. Use another card to write the title. Differentiate the title by drawing a box around the outer edge of the card or using a different color card. 6) Debrief using these questions: 6.1) What stands out for you as you look at these cards? 6.2) What is unexpected about these cards? What was difficult about this task? What parts felt positive? 6.3) What patterns do you see in the clusters? What do those patterns mean for us as a team? 6.4) What does this suggest for us as next steps?	RTE, Agile Team

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Identify strength	Identify strengths so the Team can build on them in the next iteration. Provide balance when an iteration, release, or project hasn't gone well. Team members interview each other about high points on the project. The goal is to understand the sources and circumstances that created those high point.	RTE, Agile Team
Develop satisfaction histogram	Highlight how satisfied Team members are with a focus area. Provide a visual picture of current status in a particular area to help the team have deeper discussions and analysis. Acknowledge differences in perspective among team members. Team members use a histogram to gauge individual and group satisfaction with practices and process.	RTE, Agile Team
Develop team radar	Help the Team gauge how well they are doing on a variety of measures, such as, engineering practices, team values, or other processes. Team members track individual and group ratings for specific factors about process or development practices they want to examine.	RTE, Agile Team
Play like to like game	The Agile Team uses this game to gather data during an iteration, release, or project retrospective. This also helps Team members recall their experiences during the iteration (release or project), and hear that others may have perceived it differently. Team members take turns judging which events or factors about their iteration are the best fits for quality cards. As the cards are evaluated, team members learn about each other's perspective on the same events or conditions.	Agile Team
Perform brainstorming	The Agile Team performs brainstorming to generate insight in an iteration, release, or project retrospective. Through brainstorming, the Team generates a large number of ideas and filters them against a defined set of criteria. Team members generate ideas using traditional brainstorming, then test whether each idea is applicable to the current situation.	RTE, Agile Team

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Perform force field analysis	To examine what factors in the organization will support a proposed change and which will inhibit the change. The Team defines a desired state they want to achieve. Small groups work to identify the factors that could either restrain or drive the change they want. The factors are listed on a poster, then the group assesses the strength of each supporting factor relative to the other supporting factors and repeats the process for inhibiting factors. The Team discusses which factors they can influence-either by increasing the strength of a supporting factor or by reducing the strength of an inhibiting factor.	RTE, Agile Team
Identify five whys	Discover underlying conditions that contribute to an issue. Team members work in pairs or small groups to look at issues. They ask Why five times to get beyond habitual thinking. Steps: 1) Review the issues and themes that the Team has already identified. 2) Divide the Team into pairs or small groups (no more than four to a group). And explain the process. 3) Monitor the time, and ring a chime or otherwise announce when the time is up. 4) Have the groups report what they discovered 5) Use this information as input into the next phase, Decide What to Do.	RTE, Agile Team

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Perform root causes analysis	Look past symptoms to identify root causes related to an issue. Look for reasons behind problems and breakdowns. The Team identifies factors that are causing or affecting a problem situation and then looks for the most likely causes. After they've identified the most likely causes, they look for ways they can make changes or influence those factors. Steps: 1) Draw a fishbone diagram and write the problem or issue at the fish's head. Include the five W's-What, Who, When, Where, and Why. Label the bones of the fish with categories. 2) Brainstorm factors within each category. 3) Continue asking Why is this happening? 4) Look for items that appear in more than one category. These may be the most likely causes. Engage the group in looking for areas where they can make a difference.	RTE, Agile Team
Identify patterns	Look for links and connection between facts and feelings. Analyze the data about the iteration/release/project. Guide the group in recognizing and naming patterns that contribute to current issues. After gathering data, facilitate a discussion to analyze the data, looking for patterns of events, behaviors, or feelings. Also look for times when there has been a shift; for example, everything was going smoothly, and then the energy dropped. Capture insights on flip charts or, if you're using a timeline, right on the timeline	RTE, Agile Team
Prioritize dots	Team members prioritize the top issues, ideas, or proposals.	RTE, Agile Team
Share report	Each small group shares the result of their work with the whole group. The retrospective leader keeps a progress bar to help the reporter stay within time. After the final report, the group looks for common threads and themes and identifies those they want to work on.	RTE, Agile Team

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Identify themes	Find common threads from Locate Strengths interviews. Discern compelling ideas for experiments, changes, and recommendations. After Locate Strengths interviews, the interview pairs form groups and report what each learned as they interviewed the other person. As they report the high points, Team members listen for common themes and compelling ideas. After the identifying themes, the group clusters all the cards. Small groups self-select to further define the ideas contained in the cluster.	RTE, Agile Team
Learn matrix	Team members look at four perspectives on their data to brainstorm a list of issues quickly.	RTE, Agile Team
Develop detailed plan	Team members work individually or in pairs to brainstorm all the tasks necessary to complete an experiment, improvement, or recommendation. After brainstorming, Team members eliminate redundant tasks and fill in gaps. The task are arranged in order, and team members sign up for tasks they will complete.	Agile Team
Develop SMART goals	Focus the Team's attention on developing goals that are Specific, Measurable, Attainable, Relevant, and Timely. Goals that have these characteristics are more likely to reach fruition.	Agile Team
Choose an experiment or action	Team members engage in a question asking and answering process to reach consensus on next steps.	RTE, Agile Team
Discover differing perspectives	Help to discover differing perspectives on how the Team is doing and provide variety in very short retrospectives. The team brainstorms lists of ideas for action, in response to prompts on the 2-3 flip charts.	RTE, Agile Team
Nominate PM	DoD nominates a PM to provide the status of each project or team, she or he has been assigned. When finished, that PM nominates the next person or PM to provide his/her team update.	DoD, PM
Provide project status	Each PM provides the current status of each project they are responsible for, and the progress of the delivery team. During this session, the PM also discusses obstacles they are facing, and DoD takes notes for further action or follow-up.	PM, DoD, Product Mgr, QA Lead, Support, DBA

Continued on next page

Table I.32 – *Continued from previous page*

Step	Description	Role(s)
Identify strength	Team identify the strengths and improvements to retrospect on the retrospective. The Team also identifies strengths (do more of) and changes for the next retrospective.	RTE, Agile Team
Encourage team	To allow Team members to notice and appreciate each other. End the retrospective on a positive note. Team members appreciate other team members for helping them, contributing, to the team, solving a problem, etc. However, offering an appreciation is optional.	RTE, Agile Team
Identify temperature reading	Team members report on what's happening to them and what they want.	RTE, Agile Team
Gather feedback	Retrospective leader gathers feedback from Team members to discover what helped team members work and learn together during the session, find out what hindered them, and get ideas about what else to try in future retrospectives.	RTE
Ask team feedback on retrospective	At the end of retrospective, ask Team members to give feedback on whether they spent their time well. This session help to generate feedback on the retrospective process and gauge the effectiveness of the session from the team members' perspectives.	RTE

I.5.13 Conduct problem solving workshop (S4PR161)

I.5.13.1 Purpose

The purpose of this problem solving workshop is to solve a problem in a systematic and structured way.

I.5.13.2 Background

There are different types a problem solving techniques available, however, SAFe prescribes a root cause analysis problem solving workshop. Root cause analysis helps to identify the root causes of a problem rather simply addressing the symptoms (Leffingwell, 2007, 2015). On the agile release train, a RTE usually facilitates this problem-solving session, in a timebox of two hours or less.

I.5.13.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

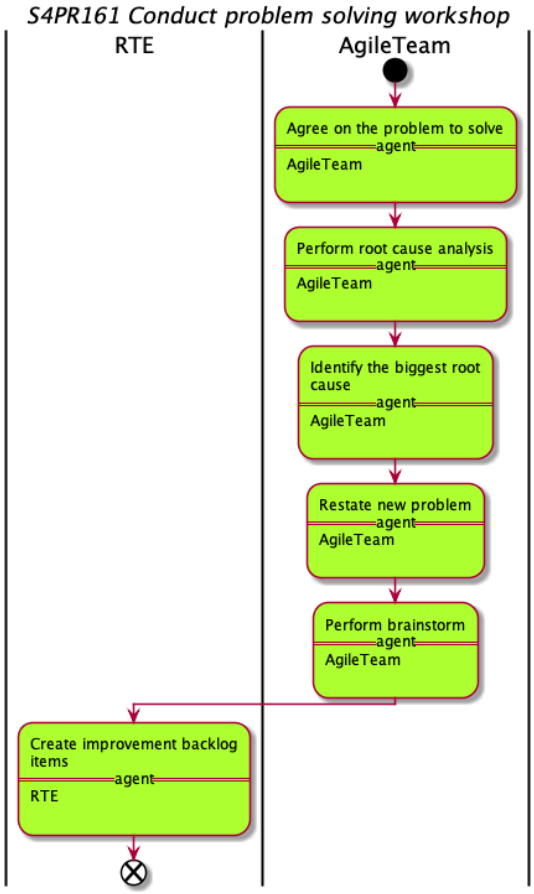

Figure I.56: S4PR161 To-be context.

I.5.13.4 Stakeholders

Agile Team and RTE.

I.5.13.5 Entry and Exit Criteria

- Entry Criteria: Problem
- Exit Criteria: Improvement feature, Improvement stories

I.5.13.6 Implementation details

Table I.33: Implementation details for practice S4PR161 – Conduct problem solving workshop.

Step	Description	Role(s)
Agree on the problem to solve	At this point, the Team has a self-selected problem they want to work on. But do they really agree on what the problem is, or is it more likely that they have differing perspectives? To this end, teams should spend a few minutes stating the problem, thinking about the what, where, when, and impact as succinctly as they can.	Agile Team
Perform root cause analysis	Fish-bone diagram is a visual tool used to explore the causes of specific events or sources of variation in a process. Causes are identified and then grouped into major categories as bones off the main bone namely 'People', 'Process', 'Tools', 'Program', and 'Environment'. Once a cause is identified, its root cause is identified with the 5 Whys technique. By simply asking Why, as many as five times, each cause of a cause is easier to discover and is added to the diagram.	Agile Team
Identify the biggest root cause	Once all the possible cause of causes have been identified, Team members then cumulatively vote on the item they think is the biggest factor causing the end problem. They can do this by placing stars (five stars are allocated to each group member, which can be spread among one or more items as they see fits on the causes they think are most problematic. The team then creates a Pareto chart, which illustrates their collective consensus on the largest root causes.	Agile Team
Restate new problem	The next step is to pick the largest cause from the list and restate it clearly as a problem.	Agile Team

Continued on next page

Table I.33 – *Continued from previous page*

Step	Description	Role(s)
Perform brainstorm	At this point, the root cause will start to imply some potential solutions. The working group brainstorms as many possible corrective actions as they can think of in a 15-30 minute session. The rules of brainstorming apply here: 1) Generate as many ideas as possible 2) Do not allow criticism or debate 3) Let the imagination soar 4) Mutate and combine ideas.	Agile Team
Create improvement backlog items	The Team then cumulatively votes on up to three most likely solutions. These will serve as improvement stories and features to be fed directly into the PI Planning session that follows. During that session, the RTE helps ensure that the relevant improvement Stories are loaded onto the Iteration plans, thus ensuring that action will be taken and resources allocated, as with any other backlog item. This closes the loop on the retrospective and ensures that people and resources are dedicated as necessary to improve the current state.	RTE

I.5.14 Perform verification and validation (S4PR212)

I.5.14.1 Background

Verification is used to illustrate how each phase of the process (PRS to SRS, SRS to code) meets the requirements imposed by the prior phase. During the verification process, we have to:

- Make sure that it works correctly, and
- be able to prove that it does via documentation and audit trail (change control).

Moreover, the verification process is done via change control mechanisms, tracking and traceability matrices, and regression test automation.

The code developer delivers creates values to the end user which evolve incrementally. However, before any software released for use by the end user- be it alpha, beta or production code. The results of all this work will have to be

validated and shall be performed under defined operating conditions i.e. *testing under actual or simulated usage scenarios* to ensure that the devices conform to define user needs and intended uses.

Moreover, validation is the final "end run" on quality, where we evaluate the system increment against its product or system level functional and nonfunctional requirements to make sure it does what we intended

I.5.14.2 As-is Context

ctions altered by this practice are shown in purple.

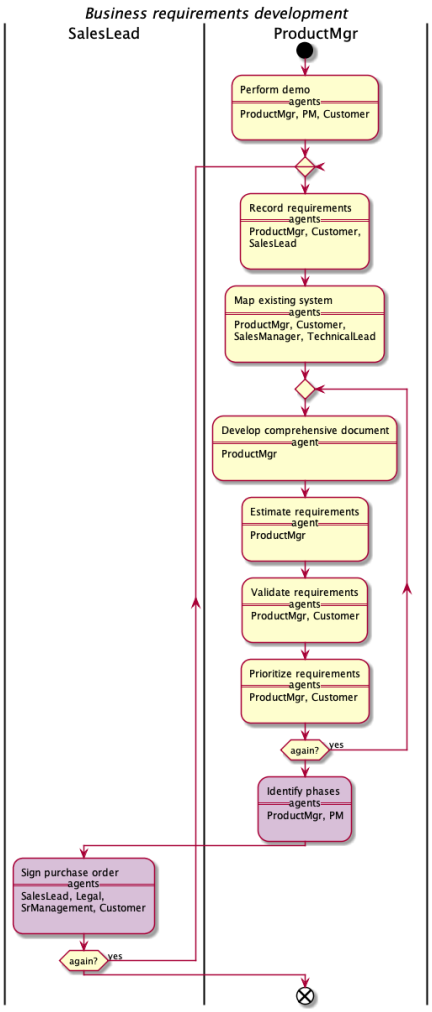

Figure I.57: S4PR212 As-is context.

I.5.14.3 To-be Context

New actions are shown in green.

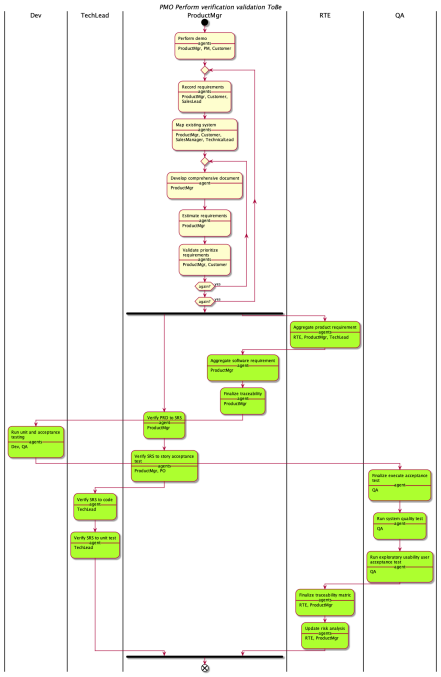

Figure I.58: S4PR212 To-be context.

I.5.14.4 Stakeholders

Product Mgr, PM, Customer, Sales Lead, Sales Manager, Technical Lead, PO, Tech Lead, RTE, Dev, and QA.

I.5.14.5 Entry and Exit Criteria

- Entry Criteria: Existing system
- Exit Criteria: Mitigation, Verified Unit test

I.5.14.6 Implementation details

Table I.34: Implementation details for practice S4PR212 – Perform verification and validation.

Step	Description	Role(s)
Perform demo	PM or Product Manager demonstrates the existing product to the (future) Customer and identifies the initial Customer needs and requirements.	Product Mgr, PM, Customer
Record requirements	Product Manager or PM collects requirements from various sources, such as the Customer and Sales Lead, resulting in a list of requirements. A team member also spends time with the Customer to extract existing software flow.	Product Mgr, Customer, Sales Lead
Map existing system	Product Manager maps the Customer's current as-is system to Ocuco's existing system to identify how the requirements will fit with the existing system. This approximately two week analysis fills the gap between what the Customer would like to have and what the Ocuco team can provide. Then, the Ocuco team provides a bespoke development document to the Customer to check whether the needs and cost are aligned.	Product Mgr, Customer, Sales Manager, Technical Lead
Develop comprehensive document	Product Manager develops a comprehensive document of the list of requirements based on the Customer's to-be context. The Product Manager provides a detailed functional requirement document to the Customer, showing what is implementable and the gap between the existing product and the to-be product.	Product Mgr
Estimate requirements	Product Manager estimates the list of development requirements.	Product Mgr
Validate prioritize requirements	Customer validates and prioritizes the list of development requirements.	Product Mgr, Customer
Verify PRD to SRS	Product Manager verifies that the software requirements specification (SRS) document is correctly translated from product requirements document (PRD).	Product Mgr
Verify SRS to story acceptance test	Product Manager and PO verify that agile "user stories" are developed based on SRS.	Product Mgr, PO

Continued on next page

Table I.34 – *Continued from previous page*

Step	Description	Role(s)
Verify SRS to code	Technical Lead verifies the code using SRS via Source Control Management (SCM) change item. Jankins supports SCM options and instructs Jenkins to obtain your Pipeline from Source control management.	Tech Lead
Verify SRS to unit test	Technical Lead verifies the code unit tests via SCM change item.	Tech Lead
Aggregate product requirement	Release Train Engineer, Product Manager, and Technical Lead help to aggregate program increments of product requirements into a version control system.	RTE, Product Mgr, Tech Lead
Aggregate software requirement	Product Manager aggregates increments of software requirements (user stories) into an SRS document (traditional document, repository, data base, etc.)	Product Mgr
Finalize traceability Run unit and acceptance testing	Product Manager finalizes traceability from PRD to SRS.	Product Mgr
Run unit and acceptance testing	Developer runs unit testing to assure that they still pass. Then, QA also runs story level acceptance testing, regression tests to assure that they still pass.	Dev, QA
Finalize execute acceptance test	QA runs system level test to make sure that the features are working perfectly as documented in PRD.	QA
Run system quality test	QA runs all system quality tests for nonfunctional requirements such as reliability, accuracy, security.	QA
Run exploratory usability user acceptance test	QA performs exploratory, usability, and user acceptance tests until it pass.	QA
Finalize traceability metric	RTE and Product Manager finalize and update traceability matrices to reflect the current state.	RTE, Product Mgr
Update risk analysis	RTE and Product Manager identify mitigation technique for each risk, then update accordingly.	RTE, Product Mgr

I.5.15 Perform value stream backlog refinement (S4VS95)

I.5.15.1 Purpose

The purpose of this practice is to build the backlog of capabilities.

I.5.15.2 Background

Technical and content authority are involved in the value stream level to build the backlog of capabilities. During the value stream backlog refinement, the Value Stream Engineer (VSE) sizes and elaborates the business benefits of each capability, then develop acceptance criteria (Leffingwell, 2015).

I.5.15.3 As-is & To-be Context

I have not observed any as-is process model for this practice. New actions are shown in green.

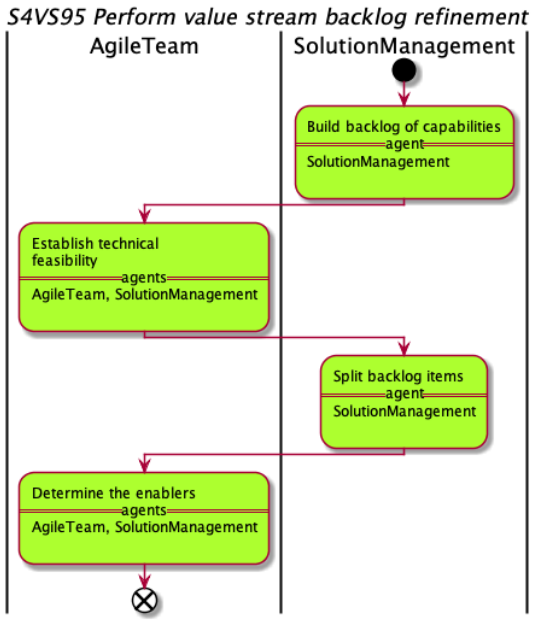

Figure I.59: S4VS95 To-be context.

I.5.15.4 Stakeholders

Solution Management and Agile Team.

I.5.15.5 Entry and Exit Criteria

- Entry Criteria: Capabilities
- Exit Criteria: Allocated Capacity

I.5.15.6 Implementation details

Table I.35: Implementation details for practice S4VS95 – Perform value stream backlog refinement.

Step	Description	Role(s)
Build backlog of capabilities	Solution Management reviews and updates backlog item definition then develop acceptance criteria.	Solution Management
Establish technical feasibility	Solution Management works with the Team to establish technical feasibility and scope estimate.	Agile Team, Solution Management
Split backlog items	Solution Management analyzes ways to split backlog items into smaller chunks of incremental value.	Solution Management
Determine the enablers	Determine the Enablers spawned by new features and capabilities, and establish their capacity allocation.	Agile Team, Solution Management

I.5.16 Develop value stream capability (S4VS105)

I.5.16.1 Purpose

A capability describes the higher-level behaviors of a solution at the Value Stream Level.

I.5.16.2 Background

"Capabilities are similar to features; however, they describe higher-level solution behaviors and often take multiple ARTs to implement (Leffingwell, 2007, 2015)."

These are the actions that the stakeholders of your product can take to get value from the product. This is more in line with the use cases you want to solve for each stakeholder. Examples are:

- Facebook users can post to their 'friends' or 'friends of friends'
- Facebook users can message their friends using Instant Message (IM) within Facebook web application

I.5.16.3 As-is & To-be Context

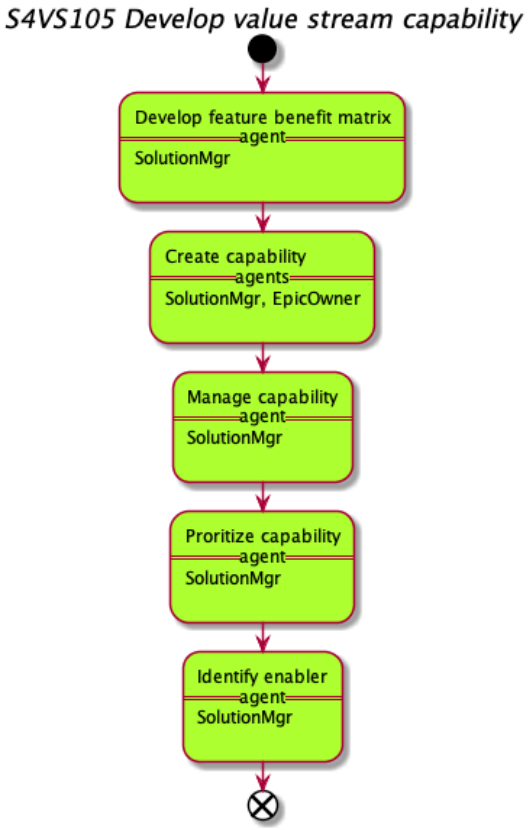

Figure I.60: S4VS105 To-be context.

I.5.16.4 Stakeholders

Solution Mgr and Epic Owner.

I.5.16.5 Entry and Exit Criteria

- Entry Criteria: Epic
- Exit Criteria: Enabler

I.5.16.6 Implementation details

Table I.36: Implementation details for practice S4VS105 – Develop value stream capability.

Step	Description	Role(s)
Develop feature benefit matrix	Solution Manager describes each capability using a feature (A short phrase giving a name and some implied context) and benefit (A short description of the benefit to the user and the business; there may be multiple benefits per feature) matrix. At this stage, the solution manager also develops acceptance criteria for each capability.	Solution Mgr
Create capability	Solution manager in collaboration with Epic owner creates capability (each capability is derived from an epic).	Solution Mgr, Epic Owner
Manage capability	Solution manager manages capabilities using a value stream backlog.	Solution Mgr
Prioritize capability	Solution manager applies WSJF strategy to prioritize capability in the value stream backlog continuously.	Solution Mgr
Identify enabler	Solution manager identifies enabler for each capability. Finally, solution manager uses associated enablers to describe and bring visibility to all the technical work necessary to support efficient development and delivery of business capabilities.	Solution Mgr

I.5.17 Facilitate continuous improvement (S4PO73)

I.5.17.1 Purpose

The purpose of this practice is to facilitate continuous improvement by quantitative metrics, customer feedback, and the inspect and adapt (I&A) retrospective cycle.

I.5.17.2 Background

SAFe prescribes *cadence based development* which is an essential method for managing the inherent variability of systems development in a flow-based system. Cadence based development makes sure that important events and activities occur on a regular basis (Leffingwell, 2007).

I.5.17.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

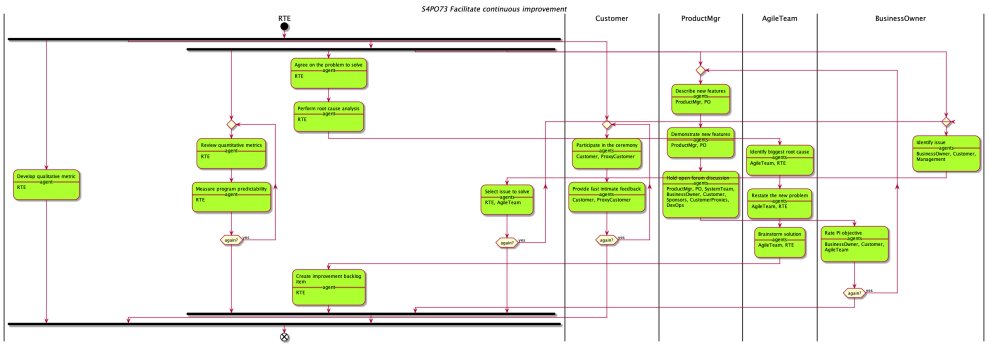

Figure I.61: S4PO73 To-be context.

I.5.17.4 Stakeholders

RTE, Customer, Proxy Customer, Agile Team, Product Mgr, PO, System Team, Business Owner, Sponsors, Customer Proxies, Dev Ops, and Management.

I.5.17.5 Entry and Exit Criteria

- Entry Criteria: Candidate issue, Issues, Iteration report, Metric agreed, PI planning or Solution demo or System demo or Inspect and adapt or Specification workshop or Beta testing or UAT or Solution validation, Program predictability, Release, Team PI performance.
- Exit Criteria: Business value, Candidate issue, Feedback, Program predictability, Quantitative metric, Team backlog or Program backlog.

I.5.17.6 Implementation details

Table I.37: Implementation details for practice S4PO73 – Facilitate continuous improvement.

Step	Description	Role(s)
Develop qualitative metric	RTE identifies quantitative metrics based on the release or incremental releases.	RTE
Participate in the ceremony	Customer participates in person or by proxy throughout agile solution development to fulfill the responsibilities.	Customer, Proxy Customer
Provide fast intimate feedback	Customers engage continuously throughout agile solution development. Customer participates either in person or by proxy to fulfill the following general responsibilities: 1) Participate as Business Owner in PI planning. 2) Attend solution and possibly system Demo to evaluate the solution increment. 3) Participate in Inspect and Adapt workshops to assist in removing some systemic impediments. 4) Interact with analysts and subject matter experts during specification workshops. 5) Collaboratively manage scope, time, and other constraints with Product and Solution Management. 6) Help to define the Roadmap, Milestones, and Releases. 7) Communicate the economic logic behind the solution and help validate assumptions in the Economic Framework. 8) Review technical and financial status of the solution. 9) Participate in beta testing, UAT, other forms of solution validation.	Customer, Proxy Customer
Review quantitative metrics	RTE and the Solution Train Engineer gather information and analyze it to identify potential issues, and facilitate the presentation of the findings to the ART.	RTE
Measure program predictability	RTE measures the program predictability of each team (in the form of PI performance report), using planned vs actual business value.	RTE
Agree on the problem to solve	The Team spends a few minutes stating a self-selected problem they want to work on, thinking about the what, where, when, and the impact of succinctly as they can.	RTE

Continued on next page

Table I.37 – *Continued from previous page*

Step	Description	Role(s)
Perform root cause analysis	SAFe recommends using Fishbone Diagrams as tools to perform root cause analysis. Candidate issues are identified and then grouped into major categories as 'bones' off the main bone. The Team preloads the main bones with the following categories: People, Process, Tools, Program, and Environment. Team members then brainstorm factors that they think contribute to the problem to be solved. Once an issue/cause is identified, its root cause is identified with the five whys technique.	RTE
Identify biggest root cause	Once all possible issues/causes have been identified, Team members then vote on the item they think is the biggest factor causing the end problem. The team creates a Pareto Chart which is a collective consensus on the largest root causes.	Agile Team, RTE
Restate the new problem	The Team picks the largest cause from the list and restates it clearly as a problem.	Agile Team, RTE
Brainstorm solution	During the brainstorming session, the Team applies the following rules: generate as many ideas as possible, do not criticize or debate, let the imagination soar, mutate and combine ideas.	Agile Team, RTE
Create improvement backlog item	The Team votes on up to three most likely solutions. These will serve as improvement stories and features to be feed directly into the PI Planning session that follows. During that session, the Release Train Engineer helps to ensure that the relevant improvement stories are loaded onto the Iteration plans, thus ensuring that action will be taken and resources allocated, as with any other backlog item.	RTE
Describe new features	The Product Manager and PO describe an integrated and aggregated view of the new features that have been delivered by the all teams on the train in the most recent iteration.	Product Mgr, PO
Demonstrate new features	The Product Manager and PO demonstrate each new feature in an end-to-end use case.	Product Mgr, PO

Continued on next page

Table I.37 – *Continued from previous page*

Step	Description	Role(s)
Hold open forum discussion	At the end of the system demo, the Product Manager opens the forum for questions and comments.	Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, Dev Ops
Rate PI objective	During the PI system demo, business owners, customers, and the other vital stakeholders collaborate with each Agile Team to rate their actual business value achieved.	Business Owner, Customer, Agile Team
Identify issue	Initially, the Team identifies issues they would like to address, and a few significant problems that the team can potentially address.	Business Owner, Customer, Management
Select issue to solve	Based on the issues identified, the facilitator helps the group decide which issues they want to tackle. Each Team has a choice of resolving Team level or Program level problems. Then the candidate issues use in the problem-solving workshop.	RTE, Agile Team

I.6 Roadmap M2:

When defining the global strategy for dealing with conflict, different types of conflict have to be taken into account.

I.6.1 GTM Context

Fig. I.62 provides GTM context of the recommendation M2.

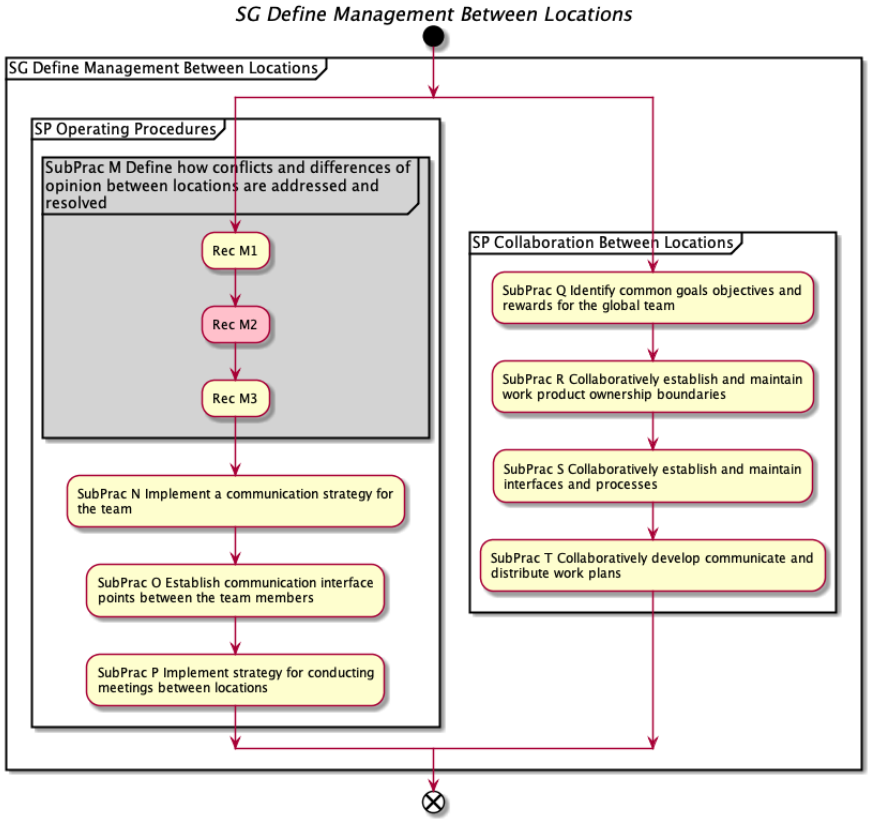

Figure I.62: M2 GTM context.

I.6.2 Why implement this recommendation?

From observations and interviews:

- Two people holding the same role leads to conflict and governance problems;

for example, a team might find it difficult to know what the priorities are, for find themselves working on two epics simultaneously.

- Small group of people are involved in strategical decision-making.

I.6.3 Facilitate trust among team members (S4T6)

I.6.3.1 Purpose

Creating a common mission, common iteration goals and team PI Objectives will facilitate trust amongst team members. Trust is important, especially in global teams, and relationships within the team are fundamentally based on trust.

I.6.3.2 Background

This practice comes under the SAFe banner of "intense collaboration." According to SAFe (Leffingwell, 2007, 2015), "collaboration is continuously improved using regular feedback loops that are built into the learning cycle of the teams. Each tangible delivery of value encourages trust, reduces uncertainty and risk, and builds confidence." Teams are likely to be motivated by having a shared vision and understanding of how their part in the development delivers value to customers.

I.6.3.3 As-is Context

I have not observed any "As-is" process for this practice.

I.6.3.4 To-be Context

The "To-be" model is shown in Fig. I.63.

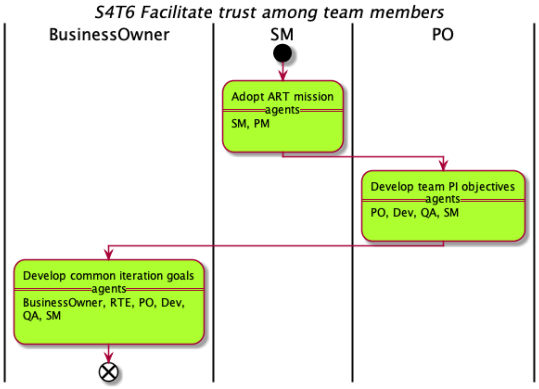

Figure I.63: S4T6 To-be context.

I.6.3.5 Stakeholders

Scrum Master, Project Manager, Product Owner, Developer, Quality Assurance, Business Owner, and Release Train Engineer (RTE).

I.6.3.6 Entry and Exit Criteria

- Entry Criteria: Common mission
- Exit Criteria: Iteration Goals

I.6.3.7 Implementation details

Table I.38: Implementation details for practice S4T6 – Facilitate trust among team members.

Step	Description	Role(s)
Adopt ART mission	Each member of the Team understands and agrees to the common ART mission.	SM, PM
Develop team PI objectives	The Team creates team PI objectives at the PI Planning meeting. PI Objectives are a summarized description of the specific business and technical goals that an Agile Team or Value Stream intends to achieve in the upcoming Program Increment.	PO, Dev, QA, SM

Continued on next page

Table I.38 – *Continued from previous page*

Step	Description	Role(s)
Develop common iteration goals	Develop iteration goals to align the Team and to provide continuous management information.	Business Owner, RTE, PO, Dev, QA, SM

I.6.4 Perform daily standup (S4T21)

I.6.4.1 Purpose

The purpose of this meeting is to understand team status/progress, escalate the problem, and get help from other team members.

I.6.4.2 Background

The daily stand-up ensures that the team knows what every member is doing, and is aware of any obstacles faced by individual team members. Potential conflicts are identified early, before they grow into problems.

Scrum teams perform a formal daily stand-up each day, where each team member describes what they did yesterday, what they are going to work on today, and any 'blockers' they are encountering. The daily stand-up meeting is time-boxed (to not more than 15 minutes), and is facilitated by the Scrum Master.

In a collocated environment, the team performs the daily stand-up actually standing up in front of the storyboard. However, if the team members are distributed, the team must use different communication tools (such as Skype or GoToMeeting) and share the storyboard with other team members online.

I.6.4.3 As-is and To-be Context

This practice is already performed as shown, so there are no new actions (see Fig. I.64).

I.6.4.4 Stakeholders

Team Member.

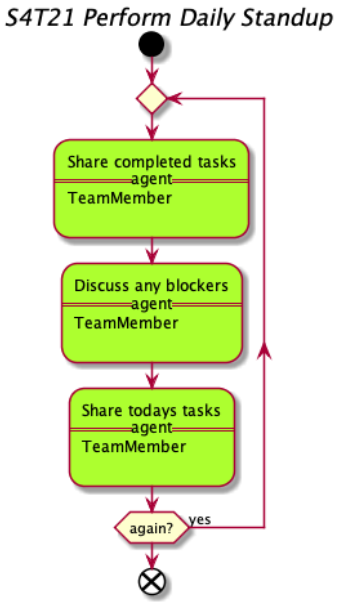

Figure I.64: S4T21 Generic SAFe model.

I.6.4.5 Entry and Exit Criteria

- Entry Criteria: Story
- Exit Criteria: Knowledge

I.6.4.6 Implementation details

Table I.39: Implementation details for practice S4T21 – Perform daily standup.

Step	Description	Role(s)
Share completed tasks	Team member describes what he or she did yesterday.	Team Member
Discuss any blockers	Team member describes any 'blockers' he or she encountered that prevented tasks from being completed.	Team Member
Share today's tasks	Team member describes what he or she is going to work on today.	Team Member

I.6.5 Develop program increment objectives (S4T58)

I.6.5.1 Purpose

The purpose of developing Program Increment (PI) objectives is to validate understanding of business and technical intent, focus alignment on outcomes (rather than process or tactical concerns), and summarize data into meaningful information that enhances alignment and provides visibility for all (Leffingwell, 2007, 2015).

I.6.5.2 Background

The Program Increment involves contributions from multiple teams. As such, having clearly defined Program Increment objectives ensures all teams know what they, as a group, are striving towards, thus eliminating tensions that might arise from conflicting priorities or objectives. PI objectives summarize specific business and technical goals that the Agile Team wants to achieve in the next program increment (Leffingwell, 2007, 2015).

This practice gives visibility into the process, and allows all stakeholders to know what to expect from each team, making all work in progress (WIP) visible. Teams execute against a current, known, aligned, and definitive set of feasible, agreed-upon objectives, based on plans that are created by, not for, the teams (Leffingwell, 2007, 2015). These PI objectives are defined during PI Planning or Post-PI planning. During PI planning, each team reviews the vision and other input objectives, defines initial stories, and places stories into iterations until capacity is full. Teams then reflect on the iteration plans, and synthesize and summarize specific technical and business objectives for their team for that particular PI. It is the combination of each team's objectives that becomes the Program PI Objectives that need to be approved by Business Owners (Leffingwell, 2007, 2015).

I.6.5.3 As-is Context

I have not observed any "As-is" process for this practice.

I.6.5.4 As-is and To-be Context

The “To-be” model shown in Fig. I.65

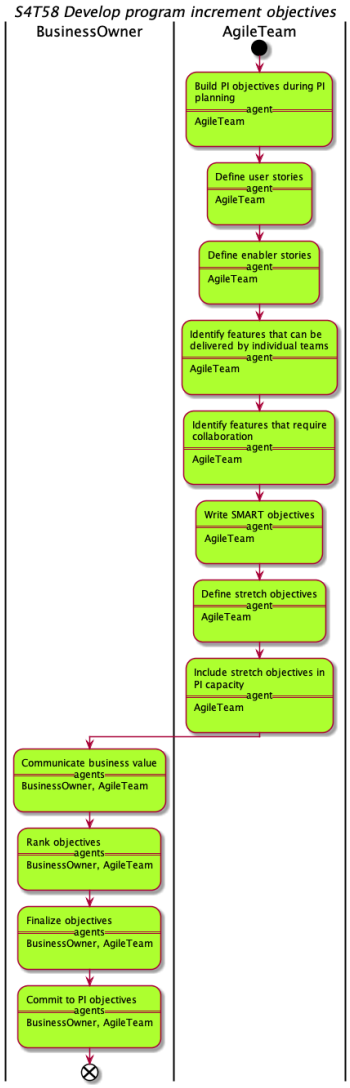

Figure I.65: S4T58 To-be context.

I.6.5.5 Stakeholders

Agile Team and Business Owner.

I.6.5.6 Entry and Exit Criteria

- Entry Criteria: Business Terms Glossary, Features, Program Vision, Stories, and Velocity.
- Exit Criteria: Committed PI Objectives created or updated.

I.6.5.7 Implementation details

Table I.40: Implementation details for practice S4T58 – Develop program increment objectives.

Step	Description	Role(s)
Build PI objectives during PI planning	Build PI Objectives during PI Planning, ensuring the following is in place: (1) solid estimating and planning; (2) well understood velocity; (3) analysis of coming features; (4) a synthesis of simple business terms; (5) visibility of the Program Vision, new features, stories that need to be delivered.	Agile Team
Define user stories	Create user stories for the backlog as follows: (1) create short, simple descriptions of a small piece of desired functionality; (2) tell the story from the user's perspective; (3) use the user's language; (4) provide just enough information for the intent to be understood by both business and technical people; (5) create a story with independent behaviour that can be implemented incrementally and that provides some value to the user or the solution; (6) write stories on an index card or sticky note.	Agile Team
Define enabler stories	Define enabler stories to reflect the technical functionality needed to implement the user stories or support other components of the system. Write stories on an index card or sticky note.	Agile Team
Identify features that can be delivered by individual teams	If a feature can be delivered without collaboration with other teams, mark it as a 'within team' feature.	Agile Team

Continued on next page

Table I.40 – *Continued from previous page*

Step	Description	Role(s)
Identify features that require collaboration	If a feature can only be delivered through collaboration with another team or teams, mark it as an 'across team' feature.	Agile Team
Write SMART objectives	Provide a concise and (S)imple description of the intended outcome (usually starting with an action verb); (M)easure what a Team needs to do to achieve the Team objectives (can be descriptive, yes or no, or quantitative, or within a range); objectives should be (A)chievable i.e, within the Team's control and influence; (R)ecognize factors that cannot be controlled; (T)ime period for achievement must be within the PI, and all objectives must be scoped appropriately.	Agile Team
Define stretch objectives	Stretch objectives are objectives that go beyond what the Team can commit to; they allow for additional work to be completed if the team finishes its committed work before the end of the sprint.	Agile Team
Include stretch objectives in PI capacity	Include stretch objectives in the capacity of the PI; allow 10-15% of the total capacity for the stretch objectives. Constantly keep in mind that stretch objectives are used to identify what can be variable within the scope of a plan; therefore, use stretch objectives to synchronize a delivery to a cadence through capacity margins: the Team commits to a capacity that aligns with the cadence, then uses stretch objectives to allow the team to deliver more if they can. Recognize stretch objectives are not the way for stakeholders to load the teams with more than they can do.	Agile Team
Communicate business value Rank objectives	Business Owner assigns business value to each of the Team's individual objectives in conversation with the team. Rank each objective according to the business owner ranking on a scale of 1 to 10. Communicate strategy and context behind the weighting decisions. Do not confuse business value with any other measure such as associated effort or total story points etc. In other words, do not calculate the business value, but rather assign business value.	Business Owner, Agile Team Business Owner, Agile Team

Continued on next page

Table I.40 – *Continued from previous page*

Step	Description	Role(s)
Finalize objectives	Finalize the Team PI Objectives with business value and stretch objectives.	Business Owner, Agile Team
Commit to PI objectives	Commit to PI Objectives by agreeing to do everything in the Team's power to meet the committed objectives.	Business Owner, Agile Team

I.6.6 Prepare preliminary iteration goals (S4T266)

I.6.6.1 Purpose

Iteration goals derive from PI objectives, and so are aligned with goals of other teams. This reduces the possibility of conflicts due to teams working at cross-purposes.

I.6.6.2 Background

According to SAFe (Leffingwell, 2007, 2015), iteration goals support "transparency, alignment, and program execution." It is not enough to commit to complete a set of stories in an iteration. Rather, it is necessary to continually review the business value of each iteration, and then communicate this value to stakeholders such as business owners, management, etc. Iteration goals can reflect–

- Features, feature slices, or feature aspects, such as research and necessary infrastructure Business or technical
- Milestones Architectural, infrastructure, exploration and compliance activities
- Iteration goals are achieved by completing backlog items, even though it may not be necessary to finish every story to meet the goals. In other words, the goals for the iteration override any particular story. On occasion, it may even be necessary to add new user stories to achieve the iteration's goals

Taking an ART view, iteration goals will help teams to focus on the bigger picture and plan for each iteration. This also feeds into the associated System Demo (for which I also have a "To Be" practice defined)

I.6.6.3 As-is and To-be Context

The "As is" actions are also performed in the "To be" process; the "To be" process also has additional tasks. New actions are shown in green.

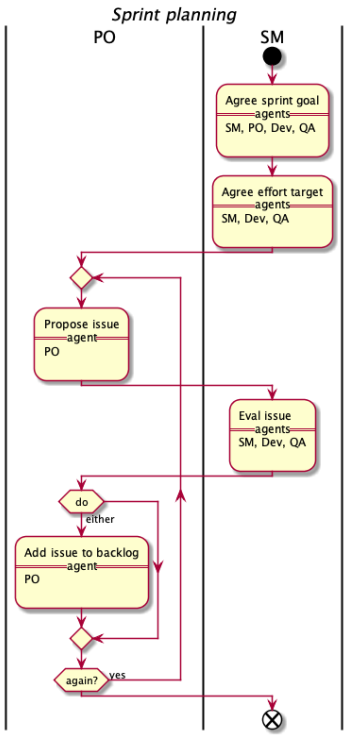

Figure I.66: S4T266 As-is context.

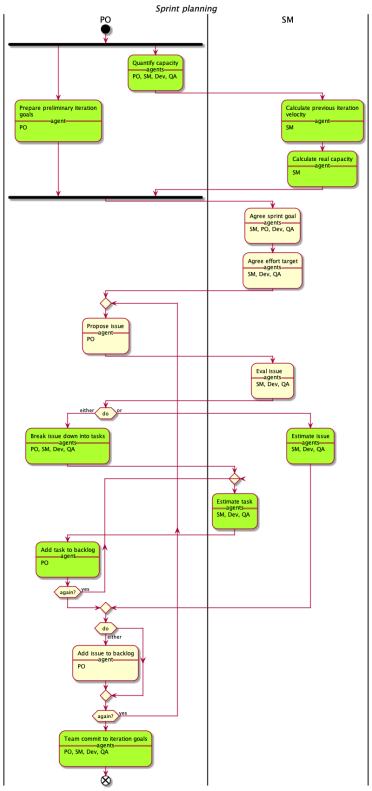

Figure I.67: S4T266 To-be context.

I.6.6.4 Stakeholders

Product Owner, Scrum Master, Developer, Quality Assurance, Architect, Senior Developer, Support Team, and External QA.

I.6.6.5 Entry and Exit Criteria

- Entry Criteria: PI objectives, Booked holidays list, and Team availability.
- Exit Criteria: Improvement Story created or updated

I.6.6.6 Implementation details

Table I.41: Implementation details for practice S4T266 – Prepare preliminary iteration goals.

Step	Description	Role(s)
Prepare preliminary iteration goals	The Product Owner prepares preliminary iteration goals.	PO
Quantify capacity	The Team quantifies (ideal) capacity.	PO, SM, Dev, QA
Calculate previous iteration velocity	The Scrum Master calculates previous iteration velocity.	SM
Calculate real capacity	The Scrum Master calculates real capacity for upcoming iteration using ideal capacity and velocity from previous iteration.	SM
Agree sprint goal	Team agree on objectives for current sprint.	SM, PO, Dev, QA
Agree effort target	The Scrum Master sets initial target (in points) based on velocity of last three sprints, then subtracts points if people will be off on leave or training, or working on other tasks. Process adjustments from retrospective will be factored in somehow.	SM, Dev, QA
Propose issue	The Product Owner proposes development issue for sprint backlog.	PO
Eval issue	The Team decides whether issue will fit into current sprint. The Scrum Master guides this decision. If there is a large high-priority task already on the backlog, lower-priority tasks will be added to fill the backlog. Also, the backlog should contain both large and small tasks so developers can see progress, and should not load QA at the end while leaving him idle at beginning.	SM, Dev, QA
Break issue down into tasks	Team breaks story down into smaller tasks.	PO, SM, Dev, QA
Estimate task	Team estimates task resulting from breakdown of story into tasks.	SM, Dev, QA
Add task to backlog	If issue now fits into current sprint window, the Product Owner adds issue to sprint backlog.	PO

Continued on next page

Table I.41 – *Continued from previous page*

Step	Description	Role(s)
Estimate issue	Team estimates stories recorded in the team backlog.	SM, Dev, QA
Add issue to backlog	If issue will fit into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Team commit to iteration goals	Team commits to iteration goals considering the capacity and estimates.	PO, SM, Dev, QA

I.6.7 Perform continuous integration (S4T272)

I.6.7.1 Purpose

Conflict between teams can arise when one team checks-in code to fix or enhance a specific product version, that breaks the build of other targets. Continuous integration identifies such issues as soon as code is committed, by building for all targets: any target build that fails is identified immediately.

In SAFe (Leffingwell, 2007, 2015), the continuous integration practice includes not only the technical infrastructure and process to achieve continuous integration, but also the culture to ensure failed integrations are fixed as soon as possible.

I.6.7.2 Background

Continuous Integration (CI) has two purposes: first, to reduce risk by identifying integration problems as soon as they arise; and second, to enable fast but sustainable development, by making new functionality that might be required by other teams available as soon as it is completed (Leffingwell, 2007, 2015).

I.6.7.3 As-is Context

Actions altered by this practice are shown in purple. New actions are shown in green. The “As-is” and “To-be” model are shown in Fig. I.68 and Fig. I.69.

I.6.7.4 Stakeholders

Developer.

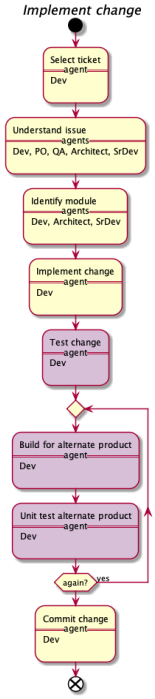

Figure I.68: S4T272 As-is context.

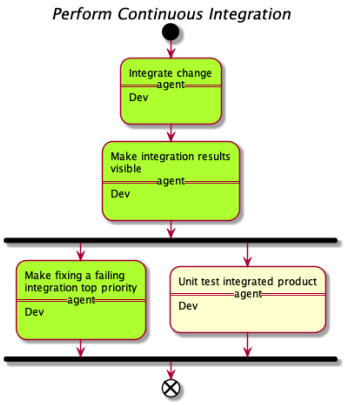

Figure I.69: S4T272 To-be context.

I.6.7.5 Entry and Exit Criteria

- Entry Criteria: Module ready for CI
- Exit Criteria: Build created

I.6.7.6 Implementation details

Table I.42: Implementation details for practice S4T272 – Perform continuous integration.

Step	Description	Role(s)
Integrate change	Commit kicks off automated build and continuous integration mechanism to build for all target platforms and configurations.	Dev

Continued on next page

Table I.42 – *Continued from previous page*

Step	Description	Role(s)
Make integration results visible	When the integration process breaks, everybody should know how and why it broke. And when it's fixed, they should know what fixed it, too.	Dev
Make fixing a failing integration top priority	When a build for any target platform fails, the team stops and fixes it immediately. Teams use flashing lights when a build is broken, and highly visible indicators of percentages of the time the system is broken.	Dev
Unit test integrated product	Developer unit-tests modified unit.	Dev

I.6.8 Participate in community of practice (S4T302)

I.6.8.1 Purpose

The purpose of a CoP is to provide a way for practitioners to share tips and best practices, ask questions of their colleagues, and provide support for each other (Leffingwell, 2007, 2015).

I.6.8.2 Background

Communities of Practice (CoPs) are organized groups of people who have a common interest in a specific technical or business domain. They collaborate regularly to share information, improve their skills, and actively work on advancing the general knowledge of the domain (Leffingwell, 2007, 2015).

Healthy CoPs have a culture built on professional networking, personal relationships, shared knowledge, and common skills. Combined with voluntary participation, CoPs provide knowledge workers with opportunities to experience autonomy, mastery, and purpose beyond their daily tasks on an Agile Release Train (ART).

CoPs enable practitioners to exchange knowledge and skills with people across the entire organization. This open membership offers access to a wide range of expertise to help with technical challenges, fuel continuous improvement and allows more meaningful contributions to the larger goals of the Enterprise. The result is

that organizations benefit from rapid problem-solving, improved quality, cooperation across multiple domains, and increased retention of top talent (Leffingwell, 2007, 2015).

I.6.8.3 As-is Context

I have not observed any "As-is" process for this practice

I.6.8.4 To-be Context

The "To-be" model is shown in Fig. I.70.

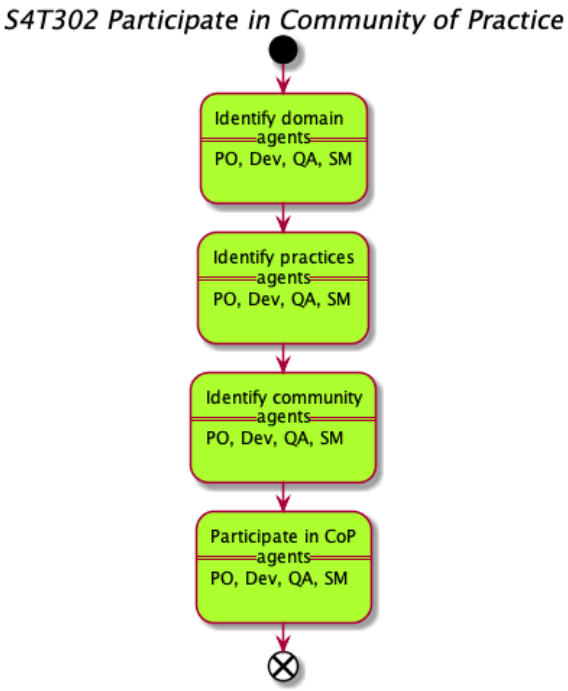

Figure I.70: S4T302 To-be context.

I.6.8.5 Stakeholders

Product Owner, Developer, Quality Assurance, and Scrum Master.

I.6.8.6 Entry and Exit Criteria

- Entry Criteria: CoP list
- Exit Criteria: Shared Explicit Knowledge created or updated

I.6.8.7 Implementation details

Table I.43: Implementation details for practice S4T302 – Participate in community of practice.

Step	Description	Role(s)
Identify domain	Identify a domain that cares about your interest.	PO, Dev, QA, SM
Identify practices	Identify a shared body of knowledge, experiences, and techniques.	PO, Dev, QA, SM
Identify community	Identify a self-selected group of individuals who care enough about the topic to participate in regular interactions.	PO, Dev, QA, SM
Participate in CoP	Access and share knowledge to promote learning in a particular area.	PO, Dev, QA, SM

I.6.9 Formulate strategic theme (S4PO3)

I.6.9.1 Purpose

The purpose of using a strategic theme is to specify and itemize business objectives that connect a SAFe portfolio to the enterprise business strategy (Leffingwell, 2007, 2015).

I.6.9.2 Background

Defining a strategic theme is an exercise in strategy formulation in the individual SAFe portfolio context. Strategic themes are an output of a collaborative process, whereby the enterprise portfolio stakeholders work with portfolio stakeholders to systematically analyze a set of inputs before arriving at conclusions. Inputs in the collaborative process include the business mission, financial objectives and

constraints, competitive environment, and portfolio context (Leffingwell, 2007, 2015).

At the Portfolio level, strategic themes are an important tool for communicating strategy, as they provide a simple and memorable reference frame.

I.6.9.3 As-is Context

I have not observed any "As-is" process for this practice.

I.6.9.4 To-be Context

The "To-be" model shown in Fig. I.71.

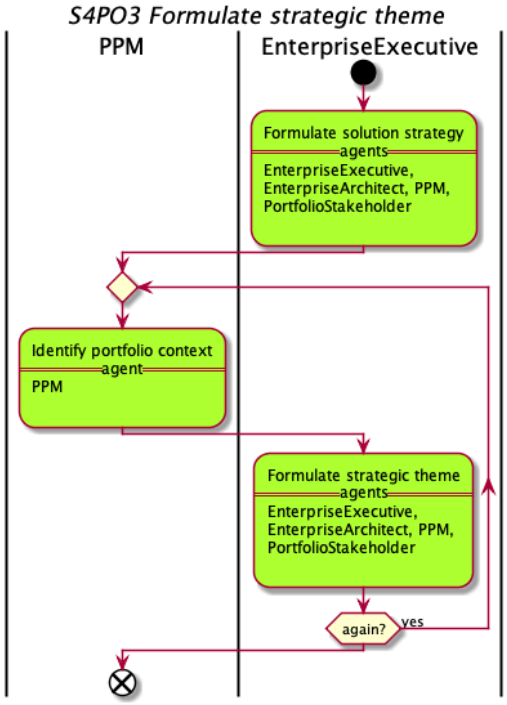

Figure I.71: S4PO3 To-be context.

I.6.9.5 Stakeholders

Enterprise Executive, Enterprise Architect, PPM, and Portfolio Stakeholder

I.6.9.6 Entry and Exit Criteria

- Entry Criteria: Business mission, Financial objective.
- Exit Criteria: Strategic theme created or updated

I.6.9.7 Implementation details

Table I.44: Implementation details for practice S4PO3 – Formulate strategic theme.

Step	Description	Role(s)
Formulate solution strategy	Defining strategic themes is an exercise in strategy formulation in the individual SAFe portfolio context. At this stage, enterprise executives provide guidance. Enterprise portfolio stakeholders work with portfolio stakeholders to systematically analyze a set of inputs before arriving at conclusions.	Enterprise Executive, Enterprise Architect, PPM, Portfolio Stakeholder
Identify portfolio context	PPM identifies portfolio context from portfolio level.	PPM
Formulate strategic theme	Portfolio stakeholders formulate strategic themes based on the solution portfolio strategy and portfolio context.	Enterprise Executive, Enterprise Architect, PPM, Portfolio Stakeholder

I.6.10 Program Portfolio Management (PPM) responsibility (S4PO25)

I.6.10.1 Purpose

Program Portfolio Management (PPM) represents the people who have the highest-level strategy and fiduciary decision-making responsibility.

I.6.10.2 Background

PPM represents those individuals who have the primary responsibility for:

- *Strategy and Investment Funding*: to allocate and assure funding to strategy.
- *Program Management*: to derive, assist, or support program execution.
- *Governance*: to close the loop on funding and program execution, measures and reporting, necessary compliance.

PPM has the responsibility to:

- participate in the establishment of the strategic themes (that guide the enterprises investment and strategy)
- determine the relevant value stream
- allocate budgets
- define and prioritize crosscutting portfolio backlog epics
- report to the business on investment spend
- report progress via key performance indicators (KPIs)

I.6.10.3 As-is and To-be Context

The “As-is” and “To-be” model are shown in Fig. I.72 and Fig. I.73.

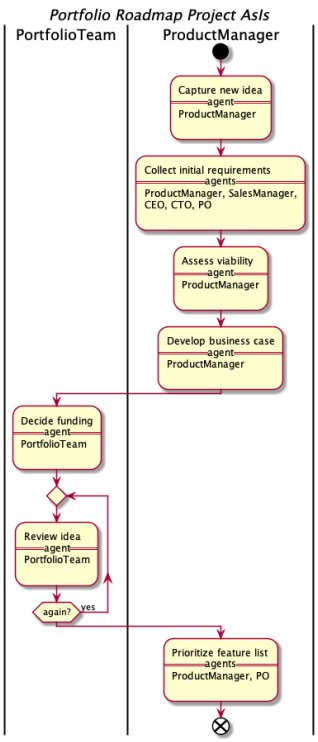

Figure I.72: S4PO25 As-is context.

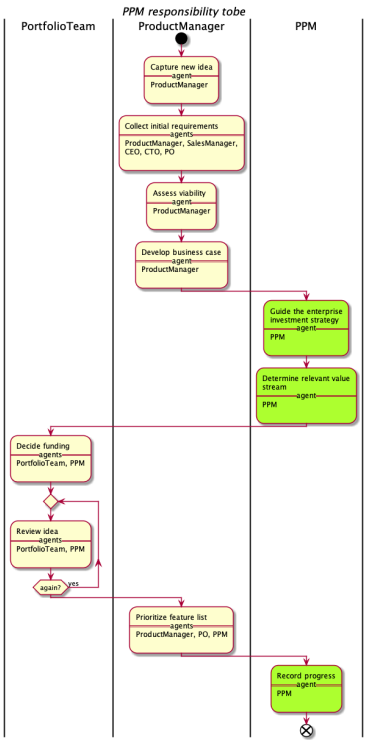

Figure I.73: S4PO25 To-be context.

I.6.10.4 Stakeholders

Product Manager, Sales Manager, CEO, CTO, PO, PPM, and Portfolio Team.

I.6.10.5 Entry and Exit Criteria

- Entry Criteria: New idea available
- Exit Criteria: Progress report created or updated

I.6.10.6 Implementation details

Table I.45: Implementation details for practice S4PO25 – PPM responsibility.

Step	Description	Role(s)
Capture new idea	The Product Manager maintains a funnel of possible new features and new integration with other products, that are submitted by sales people, management, or developers. These may be ideas they learn from customers, at trade shows, or as a result of working on the product. It is possible for anyone to add an idea to the funnel, but in practice the Product Manager is the one who does this, based on input from other parts of the organization.	Product Manager
Collect initial requirements	The Product Manager collects and records initial requirements from the CEO, CTO, a sales person, or anyone from the company who can contribute. This starts with a very high level idea, then proceeds to initial review to see if it is viable.	Product Manager, Sales Manager, CEO, CTO, PO
Assess viability	The Product Manager does an initial assessment of the viability of a new idea.	Product Manager
Develop business case	For ideas that seem viable, the Product Manager develops a business case for review by the Portfolio Team. The business case may be as simple as an email message, or an elaborate presentation to the Portfolio Team.	Product Manager
Guide the enterprise investment strategy	Program Portfolio Management (PPM) participates in the solution portfolio strategy formulation meeting to guide the enterprise for investment strategy.	PPM
Determine relevant value stream	Based on the portfolio context and portfolio strategy PPM determines relevant value stream.	PPM
Decide funding	Based on the feasibility study (which includes the business case), the Portfolio Team decides whether to fund development or not. The CFO and DoD support the CEO during this process. PPM – in-collaboration with other portfolio stakeholders – allocates budget to each value stream.	Portfolio Team, PPM

Continued on next page

Table I.45 – *Continued from previous page*

Step	Description	Role(s)
Review idea	The Portfolio Team reviews the portfolio list once each quarter for roadmapping direction and add-ons. They also gather revenue information by comparing with the previous meeting. In this stage, the CTO and DoD also decide how each portfolio item is decomposed into smaller tickets, and identify in which phase/quarter that ticket will be delivered. Due to their scope and typically cross-cutting nature, epics usually require substantial investment and have considerable impact on both the development programs and business outcomes. Therefore, in order to reach the portfolio backlog, epics must first make their way through the portfolio kanban, where they are analyzed to determine feasibility and potential ROI. Epics that reach this boundary are in a mature state, in that they have been identified, elaborated, estimated and analyzed as necessary to achieve a 'go recommendation' from PPM.	Portfolio Team, PPM
Prioritize feature list	The Product Manager and Product Owner prioritize the roadmap feature list. A requirement would get prioritized if there is clear business case, such as value and clear return (revenue) for the company.	Product Manager, PO, PPM
Record progress	Finally, PPM reports to the business on investment spend and progress via Key Performance Indicators (KPIs).	PPM

I.7 Roadmap O2

Ensure that relevant team members are made aware of how and when they will receive inputs to products they are working on, and when they need to distribute outputs from these products and when complete work products are required.

I.7.1 GTM Context

Fig. I.74 provides GTM context of the recommendation O2.

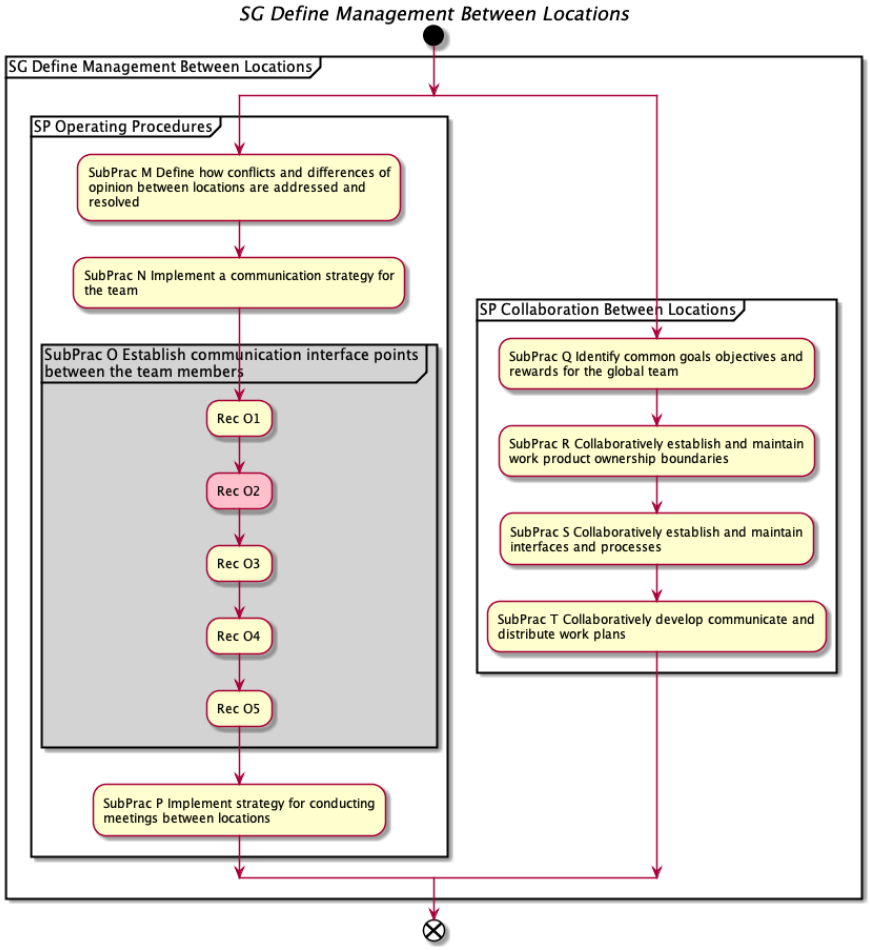

Figure I.74: O2 GTM context.

I.7.2 Why implement this recommendation?

From observations and interviews:

- Project is not well defined at the early stage of the project.
- Missing important updates due to divided time between two roles.
- Customer collaboration is a challenge due to geographical distance.
- There is no visibility of the release planning at the Program level.
- Dedicated team members in a different timezone.
- There is no Agile Release Train (ART) at the Program level.
- Lack of synchronisation in the product line.
- Third party integration.

I.7.3 Perform continuous integration (S4T272)

I.7.3.1 Purpose

Conflict between teams can arise when one team checks-in code to fix or enhance a specific product version, that breaks the build of other targets. Continuous integration identifies such issues as soon as code is committed, by building for all targets: any target build that fails is identified immediately.

In SAFe, the continuous integration practice includes not only the technical infrastructure and process to achieve continuous integration, but also the culture to ensure failed integrations are fixed as soon as possible.

I.7.3.2 Background

Continuous Integration (CI) has two purposes: first, to reduce risk by identifying integration problems as soon as they arise; and second, to enable fast but sustainable development, by making new functionality that might be required by other teams available as soon as it is completed.

I.7.3.3 As-is Context

Actions altered by this practice are shown in purple.

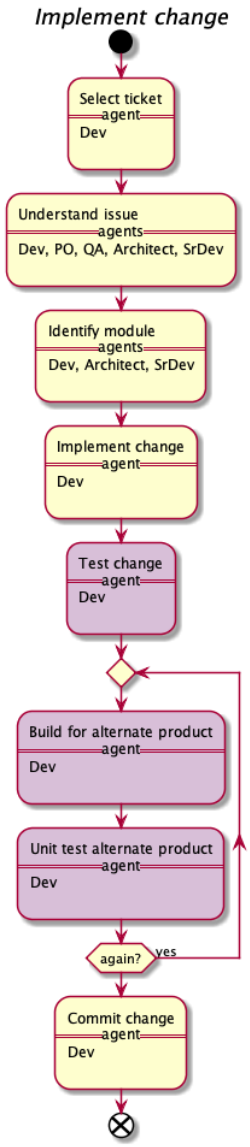

Figure I.75: S4T272 As-is context.

I.7.3.4 To-be Context

New actions are shown in green.

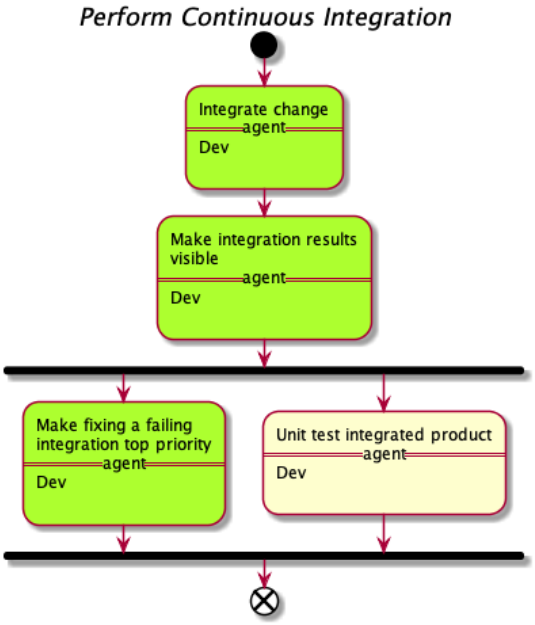

Figure I.76: S4T272 To-be context.

I.7.3.5 Stakeholders

Developer.

I.7.3.6 Entry and Exit Criteria

- Entry Criteria: PI Objective
- Exit Criteria: Improvement story

I.7.3.7 Implementation details

Table I.46: Implementation details for practice S4T272 – Perform continuous integration.

Step	Description	Role(s)
Integrate change	Commit kicks off automated build and continuous integration mechanism to build for all target platforms and configurations.	Dev

Continued on next page

Table I.46 – *Continued from previous page*

Step	Description	Role(s)
Make integration results visible	When the integration process breaks, everybody should know how and why it broke. And when it's fixed, they should know what fixed it, too.	Dev
Make fixing a failing integration top priority	When a build for any target platform fails, the team stops and fixes it immediately. Teams use flashing lights when a build is broken, and highly visible indicators of percentages of the time the system is broken.	Dev
Unit test integrated product	Developer unit-tests modified uni	Dev

I.7.4 Visualize and Limit Work in Progress (WIP) (S4T19)

I.7.4.1 Purpose

The purpose of this practice is to achieve the sustainably shortest lead time. Lean systems builders strive to achieve a state of continuous flow, whereby new system capabilities move quickly from concept to cash. Achieving continuous flow requires the elimination of the traditional start-stop-start project initiation and development process, along with the incumbent stage gates that hinder flow.

I.7.4.2 Background

During execution, the team builds and tests stories, with a goal of delivering a story or two every few days. This serialization limits work in process and helps avoid "waterfalling" the iteration. Teams use "big visual information radiators" (BVIRs) to understand and track progress during iteration execution. The team's storyboard visualizes the stories and their progress throughout the iteration. In so doing, they often use development steps as the columns and move stories left to right over time.

Some teams also apply Work-in-Progress limits to some steps to create a "pull" process within the iteration, and to continuously balance the work to increase throughput. Indeed, many teams integrate the best practices of Scrum and Kanban to facilitate the flow of work through the iterations. In this case, the simple story board above evolves into a more structured Kanban board. See the

Team Kanban article for more on the use of Kanban by ScrumXP teams (Leffingwell, 2007, 2015).

I.7.4.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

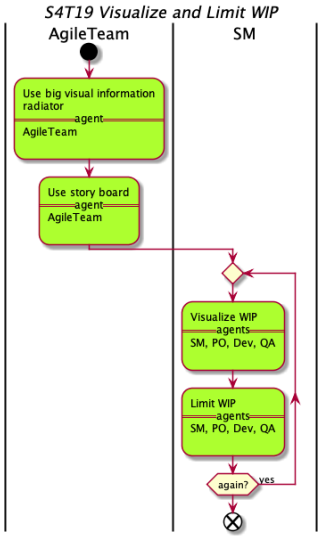

Figure I.77: S4T19 To-be context.

I.7.4.4 Stakeholders

Agile Team, SM, PO, Dev, and QA.

I.7.4.5 Entry and Exit Criteria

- Entry Criteria: Big visual information radiator
- Exit Criteria: Work In Progress List

I.7.4.6 Implementation details

Table I.47: Implementation details for practice S4T19 – Visualize and Limit Work in Progress (WIP).

Step	Description	Role(s)
Use big visual information radiator	Teams use Big Visual Information Radiators (BVIRs) to understand and track progress during iteration execution.	Agile Team
Use story board	The Team's storyboard visualizes the stories and their progress throughout the iteration. In so doing, they often use development steps as the columns (Team backlog, development, testing, and accepted), moving stories from left to right over time.	Agile Team
Visualize WIP	The first step is to make the current work in progress (WIP) visible to all stakeholders. This visualization illustrates the total amount of work at each step (Team Backlog, Development, Testing, and Accepted) and also serves as an initial process diagnostic, showing the current bottlenecks.	SM, PO, Dev, QA
Limit WIP	SM starts balancing the amount of work in progress against the available development capacity. If any step reaches its WIP limit, no new work is taken on.	SM, PO, Dev, QA

I.7.5 Participate in community of practice (S4T302)

I.7.5.1 Purpose

The purpose of a CoP is to provide a way for practitioners to share tips and best practices, ask questions of their colleagues, and provide support for each other (Leffingwell, 2007, 2015).

I.7.5.2 Background

Communities of Practice (CoPs) are organized groups of people who have a common interest in a specific technical or business domain. They collaborate regularly to share information, improve their skills, and actively work on advancing the general knowledge of the domain (Leffingwell, 2007, 2015).

Healthy CoPs have a culture built on professional networking, personal relationships, shared knowledge, and common skills. Combined with voluntary

participation, CoPs provide knowledge workers with opportunities to experience autonomy, mastery, and purpose beyond their daily tasks on an Agile Release Train (ART).

CoPs enable practitioners to exchange knowledge and skills with people across the entire organization. This open membership offers access to a wide range of expertise to help with technical challenges, fuel continuous improvement and allows more meaningful contributions to the larger goals of the Enterprise. The result is that organizations benefit from rapid problem-solving, improved quality, cooperation across multiple domains, and increased retention of top talent (Leffingwell, 2007, 2015).

I.7.5.3 As-is and To-be Context

I have not observed any as is process for this practice.

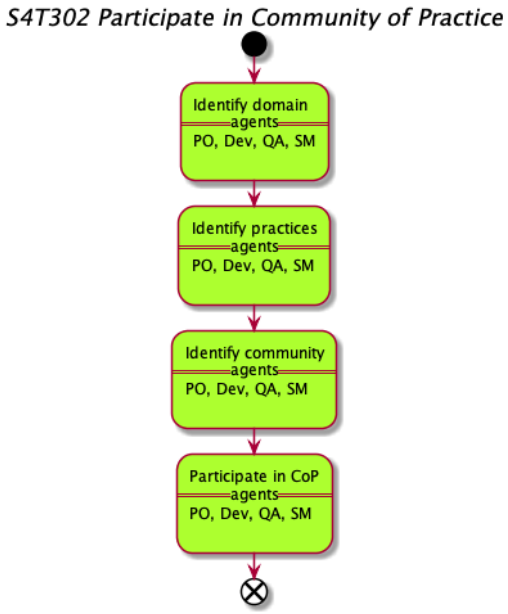

Figure I.78: S4T302 To-be context.

I.7.5.4 Stakeholders

Product Owner, Developer, Quality Assurance, and Scrum Master.

I.7.5.5 Entry and Exit Criteria

- Entry Criteria: CoP list available
- Exit Criteria: Shared Explicit Knowledge created or updated

I.7.5.6 Implementation details

Table I.48: Implementation details for practice S4T302 – Participate in community of practice.

Step	Description	Role(s)
Identify domain	Identify a domain that cares about your interest.	PO, Dev, QA, SM
Identify practices	Identify a shared body of knowledge, experiences, and techniques.	PO, Dev, QA, SM
Identify community	Identify a self-selected group of individuals who care enough about the topic to participate in regular interactions.	PO, Dev, QA, SM
Participate in CoP	Access and share knowledge to promote learning in a particular area.	PO, Dev, QA, SM

I.7.6 Perform system demo (S4PR137)

I.7.6.1 Purpose

The purpose of the System Demo is to test and evaluate the full system that the Agile Release Train (ART) is working on, and to get feedback from the primary stakeholders, including business owners, executive sponsors, other agile teams, development management, customers, and customer proxies.

I.7.6.2 Background

In addition to serving as a test of the full system that the Agile Release Train (ART) is working on, it also provides a way for the primary stakeholders – including business owners, executive sponsors, other agile teams, development management, customers, and customer proxies – to provide feedback on the evolving product.

The System Demo occurs at the end of every Iteration. It provides an integrated, aggregated view of the new Features that have been delivered by all the teams on the train in the most recent iteration (Leffingwell, 2007, 2015).

The system demo is the integrated demo of the work of all teams on the train. The system demo does not replace each team's local Team Demo, which also occurs at the end of every iteration (Leffingwell, 2007, 2015).

I.7.6.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

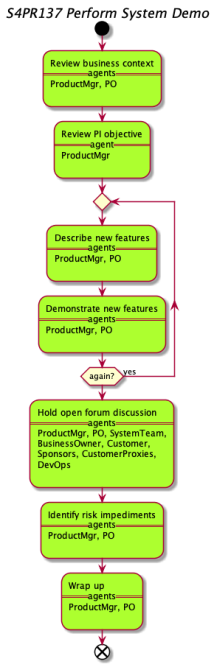

Figure I.79: S4PR137 To-be context.

I.7.6.4 Stakeholders

Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps

I.7.6.5 Entry and Exit Criteria

- Entry Criteria: Requirement
- Exit Criteria: Action items, Feedback, Progress report

I.7.6.6 Implementation details

Table I.49: Implementation details for practice S4PR137 –Perform system demo.

Step	Description	Role(s)
Review business context	The Product Manager and Product Owner review initial business context by going through each requirement of each integrated feature.	Product Mgr, PO
Review PI objective	During the system demo, the Product Manager briefly reviews the PI objectives.	Product Mgr
Describe new features	The Product Manager and PO describe integrated and aggregated view of the new features that have been delivered by the all teams on the train in the most recent iteration.	Product Mgr, PO
Demonstrate new features	The Product Manager and PO demonstrate each new feature in an end-to-end use case.	Product Mgr, PO
Hold open forum discussion	At the end of the system demo, the Product Manager opens the forum for questions and comments.	Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps

Continued on next page

Table I.49 – *Continued from previous page*

Step	Description	Role(s)
Identify risk impediments	Based on the feedback from the open forum, the Product Manager and PO note the current risks and impediments.	Product Mgr, PO
Wrap up	The Product Manager and PO wrap up by summarizing progress, feedback, and action items.	Product Mgr, PO

I.7.7 Develop PI planning (S4PR100)

I.7.7.1 Purpose

The purpose of the program increment (PI) planning is to create an emergent roadmap to deliver a prioritized and agreed set of outcomes.

I.7.7.2 Background

"PI planning is a routine, face-to-face event with a standardized agenda that includes presentation of business context and vision, followed by team planning breakouts wherein the teams create the plans for the upcoming Program Increment (PI) (Leffingwell, 2015)."

The PI planning event aligns all the teams on the ART to a shared mission and vision, facilitated by the Release Train Engineer (RTE). All members of the Agile Release Train (ART) attend whenever possible. The result is a commitment to an agreed-to set of Program PI objectives for the next PI. When the teams contributing to the Agile Release Train are geographically distributed, the PI planning meeting takes place at multiple locations simultaneously, with frequent communication among both co-located and remote teams.

A successful PI planning event delivers two primary outputs: 1) Committed PI objectives-A set of SMART objectives that are created by each team with the business value assigned by the Business Owners. The product management uses the program PI objectives to update the roadmap and will improve the forecast for the next two PIs based on what was just learned. 2) Program board-This highlights the new feature delivery dates, feature dependencies among teams and with other ARTs, and relevant milestones. The program board is often used during the Scrum of Scrums meetings to track dependencies, or it may not be

maintained (manually) after that time. This depends upon the Agile project management tooling in place and the needs of the ART.

I.7.7.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

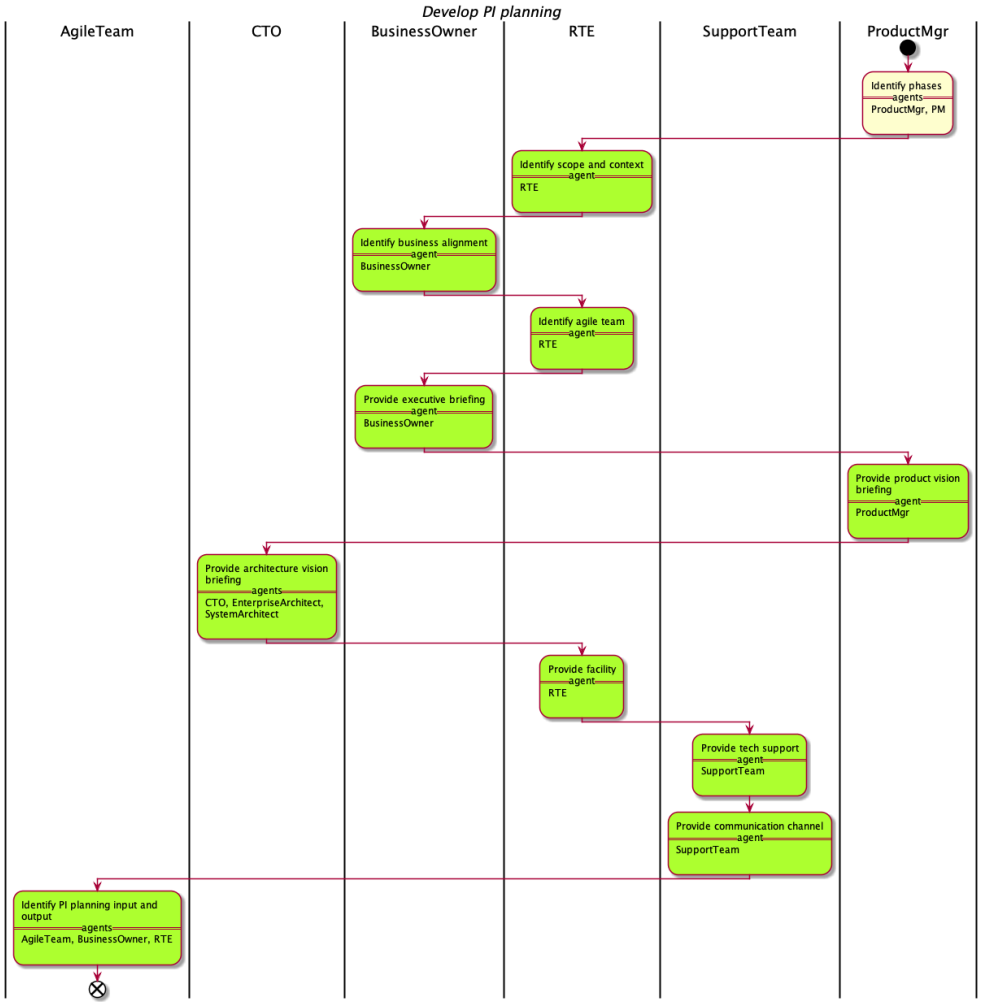

Figure I.80: S4PR100 To-be context.

I.7.7.4 Stakeholders

Product Mgr, PM, RTE, Business Owner, CTO, Enterprise Architect, System Architect, Support Team, and Agile Team.

I.7.7.5 Entry and Exit Criteria

- Entry Criteria: Customer To Be context, Prioritize list of Dev requirements
- Exit Criteria: Committed PI Objectives, Program board

I.7.7.6 Implementation details

Table I.50: Implementation details for practice S4PR100 – Develop PI planning.

Step	Description	Role(s)
Identify phases	PM or Product Manager divides the requirements into two phases. Phase 1 contains the requirements that are out of the box; phase 2 contains the bespoke development requirements.	Product Mgr, PM
Identify scope and context	The Release Train Engineer (RTE) identifies whether the scope of the planning process is understood across the train and which teams need to plan together.	RTE
Identify business alignment	Business Owners prioritize the list of topics they want to discuss or on-board during PI planning event.	Business Owner
Identify agile team	In preparation for the PI planning event, the RTE identifies the agile teams working on the agile release train, and dedicated resources such as developer, QA, Scrum Master, and Product Owner.	RTE
Provide executive briefing	A senior executive or line-of-business owner describes the current state of the business and presents a perspective on how well existing solutions are addressing current customer needs.	Business Owner

Continued on next page

Table I.50 – *Continued from previous page*

Step	Description	Role(s)
Provide product vision briefing	Product Management presents the current program vision (typically represented by the next top 10 upcoming features) and highlights any changes from the previous PI planning meeting, as well as any forthcoming Milestones.	Product Mgr
Provide architecture vision briefing	The CTO, Enterprise Architect, or System Architect present to communicate new Enablers, features, and Nonfunctional Requirements (NFRs). System Architect/Engineering also presents the architecture vision, and a senior development manager may introduce Agile-supportive changes to development practices, such as test automation, DevOps, Continuous Integration, and Continuous Deployment, which are being advanced in the upcoming PI.	CTO, Enterprise Architect, System Architect
Provide facility	The RTE confirms that there is enough space for all attendees including breakout room.	RTE
Provide tech support	The support team helps during setup and testing.	Support Team
Provide communication channel	The support team sets up primary and secondary audio, video, and presentation during distributed planning meetings.	Support Team
Identify PI planning input and output	A successful PI planning event delivers two primary outputs: 1) Committed PI objectives and 2) Program board	Agile Team, Business Owner, RTE

I.7.8 Perform inspect and adapt (I&A) workshop (S4PR152)

I.7.8.1 Purpose

The purpose of the Inspect and Adapt workshop is to create a mindset of continuous improvement requires you to spend time thinking about "what could be done better (Leffingwell, 2015)." The Inspect & Adapt workshop provides this opportunity.

I.7.8.2 Background

The Inspect and Adapt workshop is held at the end of each Program Increment. It is important that this workshop is held as it allows time to reflect on the

execution and results of the previous PI, and build improvement backlog items for the next PI. This workshop can be held at both the Program Level and Value Stream Level (Leffingwell, 2007).

The I&A workshop has three steps:

1. PI System Demo (we have covered this in practice S4PR137).
2. Quantitative measurement.
3. Retrospective and problem-solving workshop.

I.7.8.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

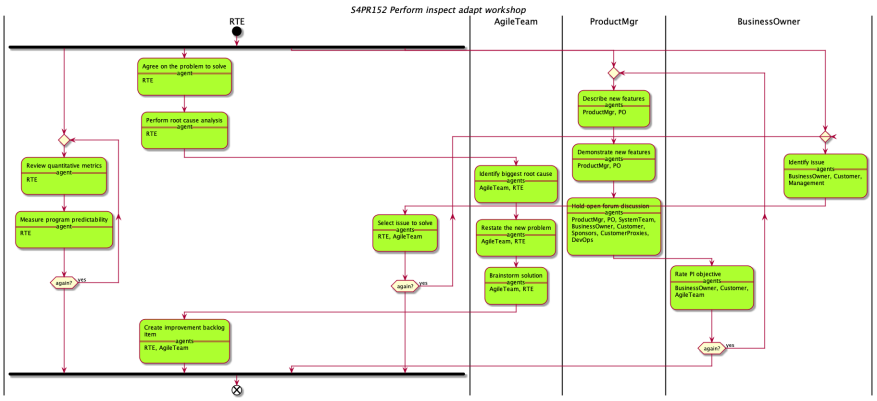

Figure I.81: S4PR152 To-be context.

I.7.8.4 Stakeholders

RTE, Agile Team, Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, DevOps, and Management.

I.7.8.5 Entry and Exit Criteria

- Entry Criteria: Candidate issue, Issues, Iteration report, Metric agreed, Program predictability, and Team PI performance report.

- Exit Criteria: Business value, Candidate issue, Program predictability, Team backlog, Program backlog.

I.7.8.6 Implementation details

Table I.51: Implementation details for practice S4PR152 – Perform inspect and adapt (I&A) workshop.

Step	Description	Role(s)
Review quantitative metrics	In preparation for this, the Release Train Engineer (RTE) and the Solution Train Engineer are often responsible for gathering the information, analyzing it to identify potential issues, and facilitate the presentation of the findings to the ART.	RTE
Measure program predictability	The RTE also measures the program predictability of each team using planned vs actual business value (in the form of PI performance report.)	RTE
Agree on the problem to solve	At this point, the teams have a self-selected problem they want to work on. The team spend a few minutes stating the problem, thinking about the what, where, when, and the impact of succinctly as they can.	RTE
Perform root cause analysis	SAFe recommend using Fishbone Diagrams tools to perform root cause analysis. Candidate issues are identified and then grouped into major categories as boned off the main bone. Team preload the main bones with the categories: People, Process, Tools, Program, and Environment. Team members then brainstorm factors that they think contribute to the problem to be solved. Once an issue/cause is identified, its root cause is identified with the five whys technique.	RTE
Identify biggest root cause	Once all possible issues/causes have been identified, Team members then vote on the item they think is the biggest factor causing the end problem. The team creates a Pareto Chart which is a collective consensus on the largest root causes.	Agile Team, RTE
Restate the new problem	In this stage, the Team picks the largest cause from the list and restates it clearly as a problem.	Agile Team, RTE

Continued on next page

Table I.51 – *Continued from previous page*

Step	Description	Role(s)
Brainstorm solution	At this stage, the root cause will start to imply some potential solutions. During the brainstorming session, the Team applies the following rules: generate as many ideas as possible, do not criticize or debate, let the imagination soar, mutate and combine ideas.	Agile Team, RTE
Create improvement backlog item	The Team then votes on up to three most likely solutions. These will serve as improvement stories and features to be fed directly into the PI Planning session that follows. During that session, the Release Train Engineer helps to ensure that the relevant improvement stories are loaded onto the Iteration plans, thus ensuring that action will be taken and resources allocated, as with any other backlog item.	RTE, Agile Team
Describe new features	The Product Manager and PO describe integrated and aggregated view of the new features that have been delivered by the all teams on the train in the most recent iteration.	Product Mgr, PO
Demonstrate new features	The Product Manager and PO demonstrate each new feature in an end-to-end use case.	Product Mgr, PO
Hold open forum discussion	At the end of the system demo, the Product Manager opens the forum for questions and comments.	Product Mgr, PO, System Team, Business Owner, Customer, Sponsors, Customer Proxies, Dev Ops
Rate PI objective	During the PI system demo, the business owners, customers, and the other vital stakeholders collaborate with each Agile Team to rate their actual business value achieved.	Business Owner, Customer, Agile Team
Identify issue	Initially, the Team identifies issues they would like to address, and team identify few significant problems that the team can potentially address.	Business Owner, Customer, Management

Continued on next page

Table I.51 – *Continued from previous page*

Step	Description	Role(s)
Select issue to solve	Based on the issues identified, the facilitator helps the group decide which issues they want to tackle. Each Team has a choice of resolving Team level or Program level problems. Then the candidate issues use in the problem-solving workshop.	RTE, Agile Team

I.7.9 Develop portfolio kanban (S4PO80)

I.7.9.1 Purpose

The purpose of the Portfolio Kanban system is to capture, analyze, approve, and track epics.

I.7.9.2 Background

Overall, there are a number of stages that an epic passes through on the way to implementation or rejection. The typical collaborators of the portfolio kanban system are: Business Owner, Product and Solution Management, Epic Owner, Development Team, and Solution/System Architect. There are in total six phases in the Portfolio Kanban system (Leffingwell, 2015).

- **Funnel** – The funnel queue is the capture queue where all ideas come from different sources i.e. portfolio strategic theme. At this stage the epic does not need to have any business case or estimates; it can be stated as a short keyword or phrase. All ideas are captured for consideration and can be represented in different formats, i.e. a document, spreadsheet, or visual system on the wall. Program Portfolio Management (PPM) typically discusses funnel epics on a periodic cadence, and epics that meet the decision criteria are moved to the review queue.
- **Review** – In this state, ideas get preliminary estimates of opportunity, effort, and cost of delay.
- **Analysis** – At this stage, more work is done to establish viability, measurable benefits, development and deployment impact, and potential availabil-

ity of resources. A lightweight business case is developed and the epic is either approved or rejected at the end of this state.

- **Portfolio Backlog** – Epics with a "go" decision are kept in the portfolio backlog and reviewed on a periodic basis. The queue in the portfolio backlog represents a low-cost holding pattern for upcoming implementation work. Epics are moved to the implementing queue when there is sufficient capacity on one or more value streams or Agile Release Trains.

- **Implementation** – When the capacity becomes available then epics are transitioned to either the Program or Value Stream Kanban, where implementation begins.

- **Done** – The Epic Owner checks the implemented epic against epic success criteria. The epic is considered done when it has met all its success criteria. However, due to the scope of epics, completion to the original intent is not always the desired case. At this stage, some identified capabilities and features might be discarded. Regardless, the epic eventually reaches a done state.

I.7.9.3 As-is Context

Actions altered by this practice are shown in purple.

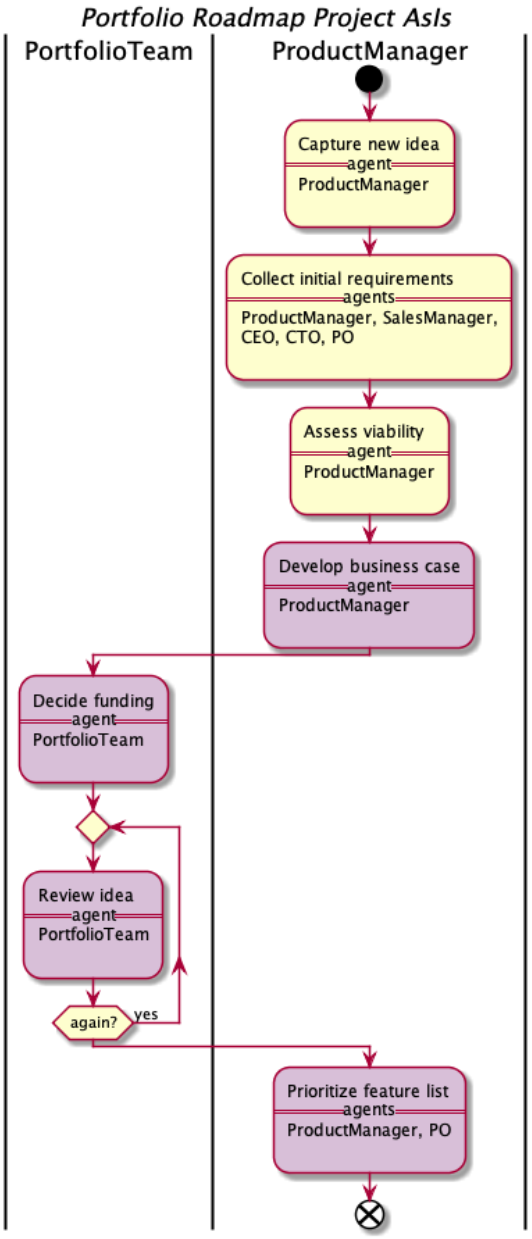

Figure I.82: S4PO80 As-is context.

I.7.9.4 To-be Context

New actions are shown in green.

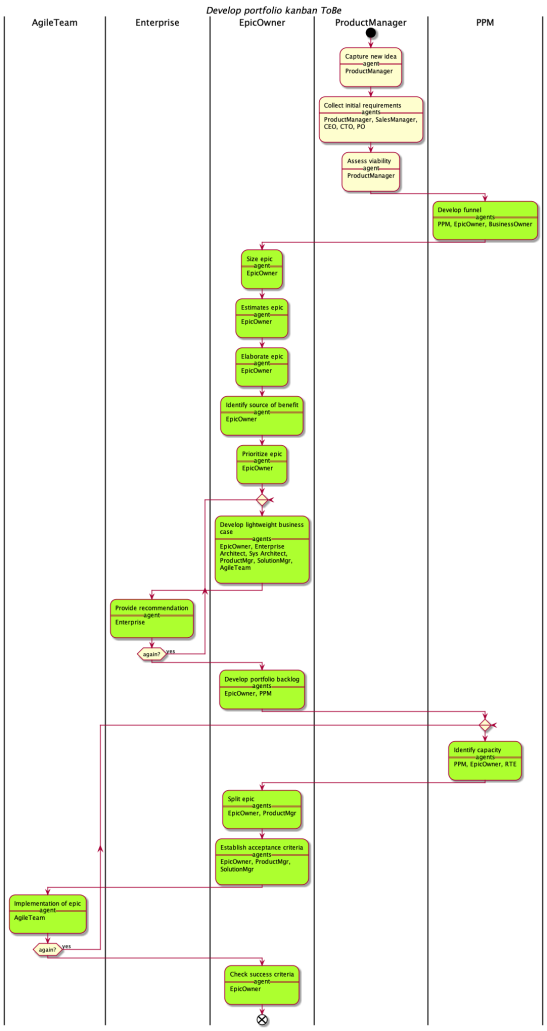

Figure I.83: S4PO80 To-be context.

I.7.9.5 Stakeholders

Sales Manager, CEO, CTO, PO, PPM, Epic Owner, Business Owner, Enterprise Architect, Sys Architect, Product Mgr, Solution Mgr, Agile Team, Enterprise, and RTE.

I.7.9.6 Entry and Exit Criteria

- Entry Criteria: New idea

- Exit Criteria: Done Epic

I.7.9.7 Implementation details

Table I.52: Implementation details for practice S4PO80 – Develop portfolio kanban.

Step	Description	Role(s)
Capture new idea	The Product Manager maintains a funnel of possible new features and new integrations with other products, that are submitted by sales people, management, or developers. These may be ideas learned from customers, at trade shows, or as a result of working on the product. It's possible for anyone to add an idea to the funnel, but in practice the Product Manager is the one who does this, based on input from other parts of the organization.	Product Mgr
Collect initial requirements	The Product Manager collects and records initial requirements from the CEO, CTO, a sales person, or anyone from the company who can contribute. This starts with a very high level idea, then proceeds to initial review to see if it is viable.	Product Mgr, Sales Manager, CEO, CTO, PO
Assess viability	The Product Manager does an initial assessment of the viability of a new idea.	Product Mgr
Develop funnel	The funnel queue is the capture queue where all ideas come from different sources, such as the portfolio strategic themes. At this stage an epic does not need to have any business case or estimates. It can be stated as a short keyword or phrase. All ideas are captured for consideration and can be represented in different formats, i.e. a document, spreadsheet, or visual system on the wall. Program Portfolio Management (PPM) typically discusses funnel epics on a periodic cadence, and epics that meet the decision criteria are moved to the review queue.	PPM, Epic Owner, Business Owner
Size epic	The Epic Owner roughly sizes the epic.	Epic Owner
Estimates epic	The Epic Owner estimates size of the epic in terms of epic value.	Epic Owner

Continued on next page

Table I.52 – *Continued from previous page*

Step	Description	Role(s)
Elaborate epic	The Epic Owner elaborates the epic using the epic value statement format. This value stream statement can be used to capture, organize, and communicate key information about an epic. There are two parts to the value statement template: 1) Forward-Looking position statement: for (customer), who (do something), the (solution), is a (something–the how), that (provides this value), unlike (competitor, current solution, or non-existing solution), our solution (does something better–the why) 2) Scope: Success criteria, In scope, Out of scope, and NFRs.	Epic Owner
Identify source of benefit	The Epic Owner identifies the business benefits from epic value statement.	Epic Owner
Prioritize epic	The Epic Owner prioritizes the epic based on the needs or business benefits.	Epic Owner
Develop lightweight business case	For ideas that seem viable, the Product Manager develop a business case for review by the Portfolio Team. The business case may be as simple as an email message, or an elaborate presentation to the Portfolio Team. An Epic Owner takes responsibility for the ongoing work. Then an action collaboration is initiated among Enterprise Architects, System Architects, Agile Teams, Product and Solution Management, and key stakeholders on the agile release train. Through this collaboration the key stakeholders also explore solution, design, and implementation alternatives. The options for internal and (or) outsourcing development are considered.	Epic Owner, Enterprise Architect, Sys Architect, Product Mgr, Solution Mgr, Agile Team
Provide recommendation	The Enterprise provides a go or no-go recommendation based on the business case and resource availability (i.e. business analyst, development team, and enterprise architect.)	Enterprise

Continued on next page

Table I.52 – *Continued from previous page*

Step	Description	Role(s)
Develop portfolio backlog	Epic with a 'go' decision is kept in the portfolio backlog and reviewed on a periodic basis. The queue in the portfolio backlog represents a low-cost holding pattern for upcoming implementation work. Epics are moved to the implementing queue when there is sufficient capacity from one or more value stream or Agile Release Trains.	Epic Owner, PPM
Identify capacity	PPM or Epic Owner identifies the capacity in the value stream. As capacity becomes available, epics are pulled into the relevant value stream or program kanban.	PPM, Epic Owner, RTE
Split epic	The epics are split into capabilities and features.	Epic Owner, Product Mgr
Establish acceptance criteria	Epic Owner, Solution Manager, and Product Manager develop acceptance criteria for capabilities and features.	Epic Owner, Product Mgr, Solution Mgr
Implementation of epic	Finally, the Team performs the actual development.	Agile Team
Check success criteria	The Epic Owner checks the implemented epic against epic success criteria. The epic is considered done when it has met all its success criteria. However, due to scope of epics, completion to the original intent is not always the desired case. At this stage, some identified capabilities and features might be discarded. Regardless, the epic reaches a done state.	Epic Owner

I.7.10 Track epic and enabler (S4PO55)

I.7.10.1 Purpose

SAFe applies a *Portfolio Kanban* (Leffingwell, 2015) system to:

- make the strategic business initiatives backlog visible;
- bring structure and visibility to the analysis and decision-making;
- provide work in progress (WIP) limits;
- drive collaboration among the key stakeholders;

- provide a qualitative and transparent basis for economic decision-making.

I.7.10.2 Background

The Epic Owner uses a Kanban board to list the number of epics and enablers. Based on the status of the epic, the Epic Owner changes the status of the epic into different phases. This is how epics and enablers can be tracked to understand which ones are being worked on and which ones have been completed (Leffingwell, 2015).

I.7.10.3 As-is Context

Actions altered by this practice are shown in purple.

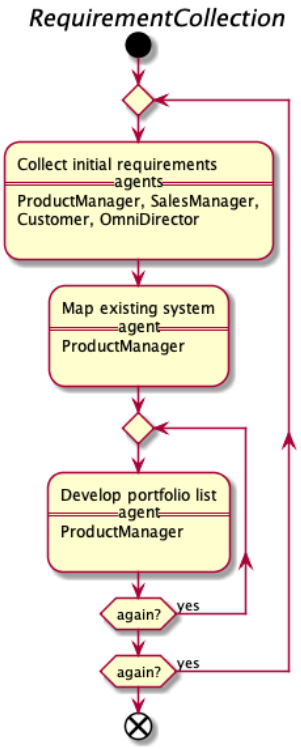

Figure I.84: S4PO55 As-is context.

I.7.10.4 To-be Context

New actions are shown in green.

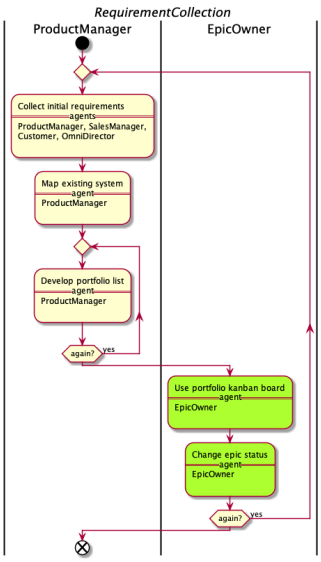

Figure I.85: S4PO55 To-be context.

I.7.10.5 Stakeholders

Portfolio Team, Product Manager, Sales Manager, Customer, and Epic Owner.

I.7.10.6 Entry and Exit Criteria

- Entry Criteria: Customer Need
- Exit Criteria: Reviewed Epic Or Analysed Epic Or Portfolio backlog Or Implemented Epic Or Completed Epic.

I.7.10.7 Implementation details

Table I.53: Implementation details for practice S4PO55 – Track epic and enabler.

Step	Description	Role(s)
Assess customer driven project	If it appears that the customer is willing to pay and there are sufficient development resources, a revenue project should be approved.	Portfolio Team

Continued on next page

Table I.53 – *Continued from previous page*

Step	Description	Role(s)
Collect initial requirements	The Product Manager collects and records initial requirements from sales person(s) in advance of contract negotiation.	Product Mgr, Sales Manager, Customer
Map existing system	The Product Manager maps the customer's current as-is system to Ocuco's existing system to identify how the requirements will fit with the existing product. In this stage, requirements are classified by the customer as must have, or nice to have.	Product Mgr
Develop portfolio list	The Product Manager develops a portfolio list incorporating all requirements. The Product Manager also makes an initial estimate for each task before putting it onto the portfolio list for resourcing by the PMO.	Product Mgr
Use portfolio kanban board	The Epic Owner uses a kanban board to list the number and status of epics and enablers.	EPic Owner
Change epic status	Based on the status of the epic, Epic Owner changes the status of the epic into a different phase. This is how epics and enablers can be tracked to understand which ones are being worked on and which one have been completed.	Epic Owner

I.7.11 Collaborate with the development team (S4RR176)

I.7.11.1 Purpose

The purpose of this practice is to collaborate with development teams to size the epic and provide input for economic prioritization based on WSJF prior to approval of an epic.

I.7.11.2 Background

In SAFe, epics drive much of the economic value for the enterprise. "Epics are containers for significant initiatives, initiatives that are large, and typically cross cutting, crossing multiple Value Streams and ARTs (Leffingwell, 2015)." The Epic Owner collaborates with the development team to prioritise the epic based

on value delivered, using techniques such as weighted shortest job first (WSJF) and the lean business case.

Epics often affect multiple products or large components. In such cases, splitting by economic value can be an effective implementation technique. As such, the epic owner works with the development team to understand when and how they will receive input to the component they are working on, including a release plan.

I.7.11.3 As-is & To-be Context

I have not observed any as-is process model for this practice. New actions are shown in green.

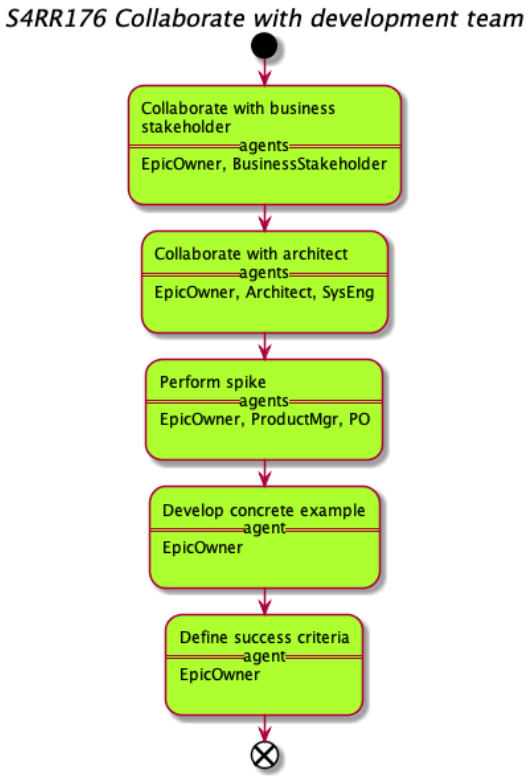

Figure I.86: S4RR176 To-be context.

I.7.11.4 Stakeholders

Epic Owner, Business Stakeholder, Architect, Sys Eng, Product Mgr, and PO.

I.7.11.5 Entry and Exit Criteria

- Entry Criteria: Business epic, Epic
- Exit Criteria: Success criteria

I.7.11.6 Implementation details

Table I.54: Implementation details for practice S4RR176 – Collaborate with the development team.

Step	Description	Role(s)
Collaborate with business stakeholder	The Epic owner collaborates with business stakeholders to understand and describe the business benefits of the business epic.	Epic Owner, Business Stakeholder
Collaborate with architect	The Epic owner collaborates with Architects or Systems Engineering, from the Value Stream and Program Levels, and Agile Team to understand implementation effort and impact on current solutions.	Architect, Sys Eng
Perform spike	The Epic owner, Product Manager, or Product owner perform implementation of a spike, research inquiry, or exploration to identify challenges and an optimal solution.	Epic Owner, Product Mgr, PO
Develop concrete example	The Epic Owner develops a concrete example to resolve ambiguities/Challenges (specification by example).	Epic Owner
Define success criteria	The Epic Owner defines the success criteria for an epic.	Epic Owner

I.7.12 Build a high performing team (S4RR32)

I.7.12.1 Purpose

This practice helps to manage intrapersonal conflicts, challenges, and opportunities for growth.

I.7.12.2 Background

Scrum Master focuses on ever-improving team dynamics and performance to build a high-performing team. Scrum Master also helps individuals and teams through personnel changes.

I.7.12.3 As-is & To-be Context

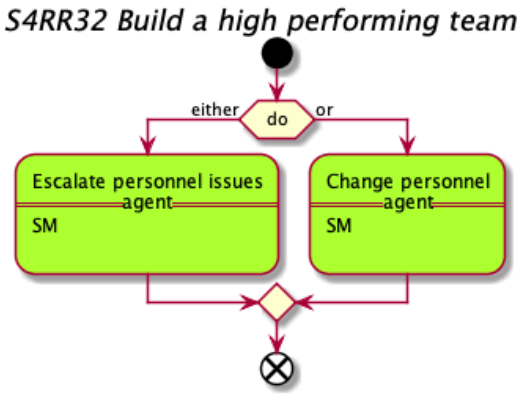

Figure I.87: S4RR32 To-be context.

I.7.12.4 Stakeholders

SM.

I.7.12.5 Entry and Exit Criteria

- Entry Criteria: Escalation, Intangible personnel issue
- Exit Criteria: Escalation, New team member

I.7.12.6 Implementation details

Table I.55: Implementation details for practice S4RR32 – Build a high performing team.

Step	Description	Role(s)
Escalate personnel issues	SM escalates people problems to management where necessary, but only after internal processes have failed to achieve the objective.	SM
Change personnel	If an individual or team member is not happy with the current team, then SM helps individuals and teams through personnel changes.	SM

I.8 Roadmap Q3

Project goals and objectives communicated, understood and agreed across all team members regardless of location.

I.8.1 GTM Context

Fig. I.88 provides GTM context of the recommendation Q3.

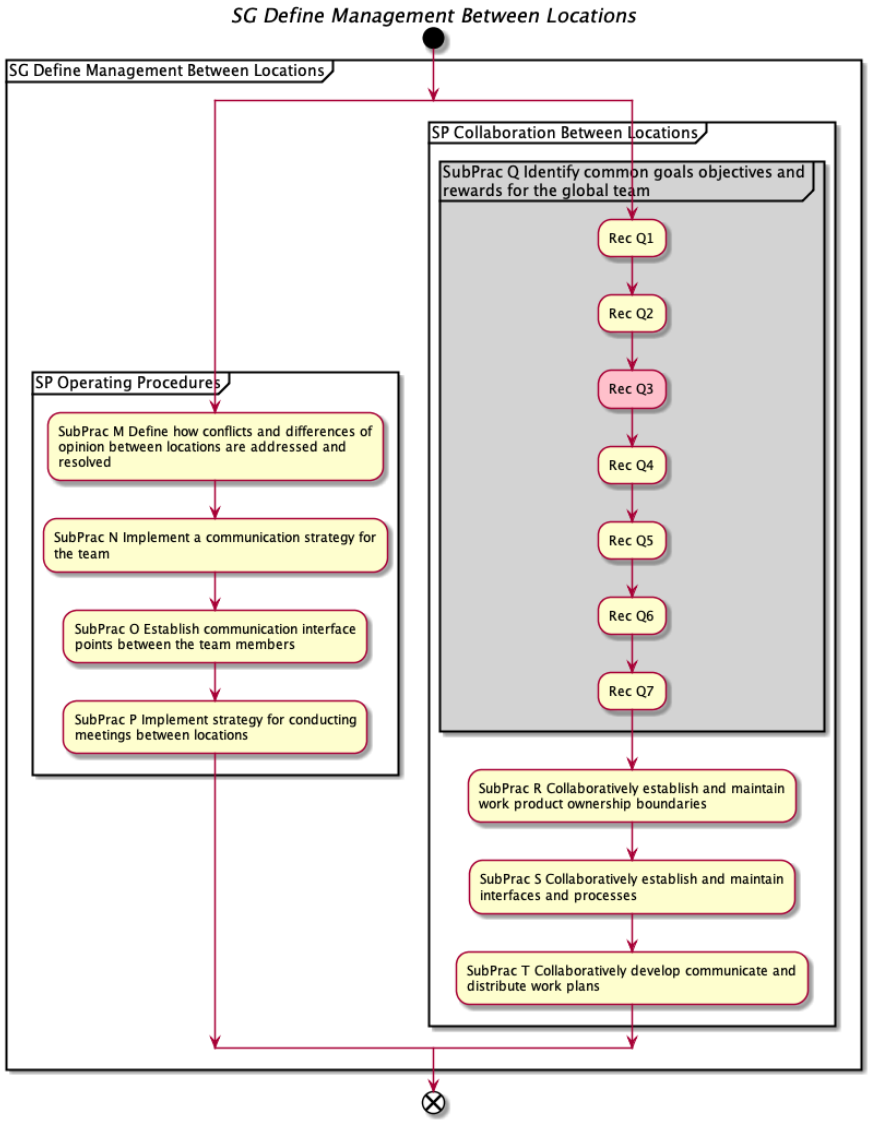

Figure I.88: Q3 GTM context.

I.8.2 Why implement this recommendation?

From observations and interviews:

- Ad-hoc processes for new work lead to confusion, with the result that the project is not well defined.
- Over commitment leads to late of delivery of promised product.
- Internal projects are always kind of suffer due to multiple responsibilities.
- Lack of synchronisation in the product line.
- Risk identification is crucial especially when dealing with third parties.
- Requirements from customer are misunderstood by the development team or recorded the wrong request.

I.8.3 Prepare preliminary iteration goals (S4T266)

I.8.3.1 Purpose

Iteration goals derive from PI objectives, and so are aligned with goals of other teams. This reduces the possibility of conflicts due to teams working at cross-purposes. According to SAFe, "Iteration Goals are a high-level summary of the business and technical goals that the Agile Team agrees to accomplish in an Iteration. They are vital to coordinating an Agile Release Train (ART) as a self-organizing, self-managing team of teams.Iteration goals provide the following benefits:

- Align the team members and the Product Owner to the mission
- Align the people to the Program Increment (PI) Objectives
- Provide context for understanding and addressing cross-team dependencies
- Whether the teams apply Scrum or Kanban, iteration goals give program stakeholders, management, and Agile teams a shared language for maintaining alignment, managing dependencies, and making necessary adjustments during the execution of the program increment (Leffingwell, 2015).

I.8.3.2 Background

According to SAFe, iteration goals support "transparency, alignment, and program execution (Leffingwell, 2015)." It is not enough to commit to complete a set of stories in an iteration. Rather, it is necessary to continually review the business value of each iteration, and then communicate this value to stakeholders such as business owners, management, etc. Iteration goals can reflect:

- Features, feature slices, or feature aspects, such as research and necessary infrastructure Business or technical;

- Milestones Architectural, infrastructure, exploration and compliance activities;

- Iteration goals are achieved by completing backlog items, even though it may not be necessary to finish every story to meet the goals. In other words, the goals for the iteration override any particular story. On occasion, it may even be necessary to add new user stories to achieve the iteration's goals (Leffingwell, 2015)."

Taking an ART view, iteration goals will help teams to focus on the bigger picture and plan for each iteration. This also feeds into the associated System Demo (for which we also have a 'To Be' practice defined).

I.8.3.3 As-is Context

The 'As is' actions are also performed in the 'To be' process; the 'To be' process also has additional tasks.

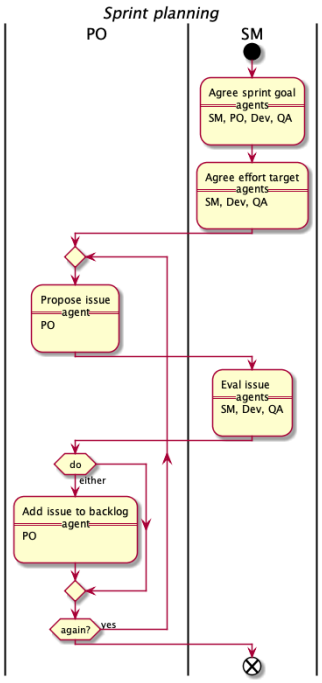

Figure I.89: S4T266 As-is context.

I.8.3.4 To-be Context

New actions are shown in green.

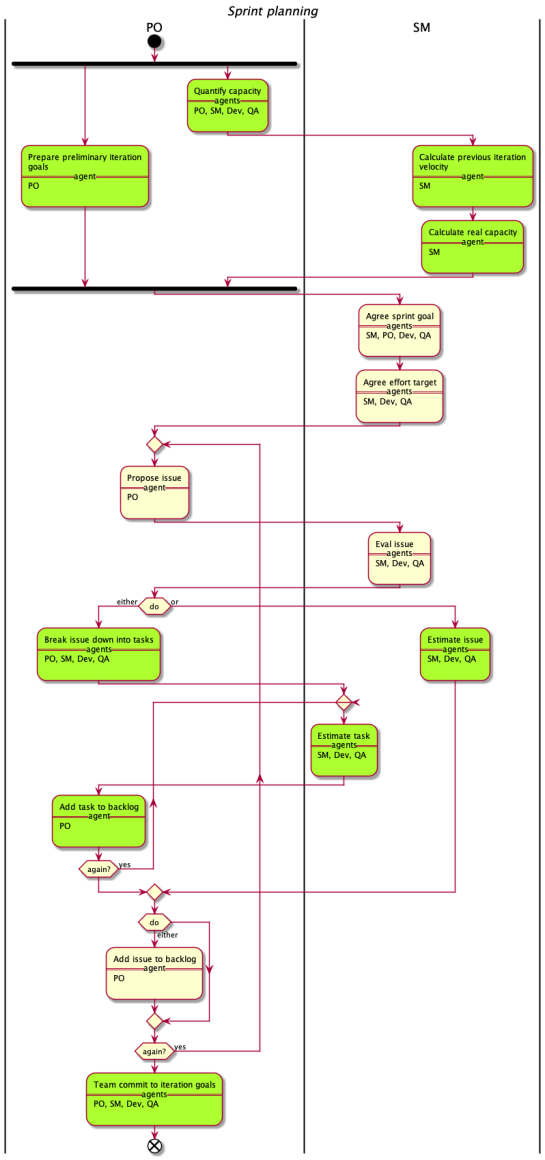

Figure I.90: S4T266 To-be context.

I.8.3.5 Stakeholders

PO, SM, Developer, and QA.

I.8.3.6 Entry and Exit Criteria

- Entry Criteria: PI Objectives, Booked holidays, Team availability
- Exit Criteria: Improvement story

I.8.3.7 Implementation details

Table I.56: Implementation details for practice S4T266 – Prepare preliminary iteration goals.

Step	Description	Role(s)
Prepare preliminary iteration goals	The Product Owner prepares preliminary iteration goals.	PO
Quantify capacity	The Team quantifies (ideal) capacity.	PO, SM, Dev, QA
Calculate previous iteration velocity	The Scrum Master calculates previous iteration velocity.	SM
Calculate real capacity	The Scrum Master calculates real capacity for upcoming iteration using ideal capacity and velocity from previous iteration.	SM
Agree sprint goal	Team agree on objectives for current sprint.	SM, PO, Dev, QA
Agree effort target	The Scrum Master sets initial target (in points) based on velocity of last three sprints, then subtracts points if people will be off on leave or training, or working on other tasks. Process adjustments from retrospective will be factored in somehow.	SM, Dev, QA
Propose issue	The Product Owner proposes development issue for sprint backlog.	PO
Eval issue	The Team decides whether issue will fit into current sprint. The Scrum Master guides this decision. If there is a large high-priority task already on the backlog, lower-priority tasks will be added to fill the backlog. Also, the backlog should contain both large and small tasks so developers can see progress, and should not load QA at the end while leaving him idle at beginning.	SM, Dev, QA

Continued on next page

Table I.56 – *Continued from previous page*

Step	Description	Role(s)
Break issue down into tasks	Team breaks story down into smaller tasks.	PO, SM, Dev, QA
Estimate task	Team estimates task resulting from breakdown of story into tasks.	SM, Dev, QA
Add task to backlog	If issue now fits into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Estimate issue	Team estimates stories recorded in the team backlog.	SM, Dev, QA
Add issue to backlog	If issue will fit into current sprint window, the Product Owner adds issue to sprint backlog.	PO
Team commit to iteration goals	Team commits to iteration goals considering the capacity and estimates.	PO, SM, Dev, QA

I.8.4 Develop program increment objectives (S4T58)

I.8.4.1 Purpose

The purpose of developing Program Increment (PI) objectives is "to validate understanding of business and technical intent, focus alignment on outcomes (rather than process or tactical concerns), and summarize data into meaningful information that enhances alignment and provides visibility for all (Leffingwell, 2015).

I.8.4.2 Background

“Making and meeting small commitments builds trust (Leffingwell, 2007).” The Program Increment involves contributions from multiple teams. As such, having clearly defined Program Increment objectives ensures all teams know what they, as a group, are striving towards, thus eliminating tensions that might arise from conflicting priorities or objectives.

PI objectives summarize specific business and technical goals that the Agile Team wants to achieve in the next program increment. This practice gives visibility into the process, and allows all stakeholders to know what to expect from each team, making all work in progress (WIP) visible. Teams execute against a current, known, aligned, and definitive set of feasible, agreed-upon objectives, based on plans that are created *by*, not *for*, the teams.

These PI objectives are defined during PI Planning or Post-PI planning. During PI planning, each team reviews the vision and other input objectives, defines initial stories, and places stories into iterations until capacity is full. Teams then reflect on the iteration plans, and synthesize and summarize specific technical and business objectives for their team for that particular PI. It is the combination of each team's objectives that becomes the Program PI Objectives that need to be approved by Business Owners.

I.8.4.3 As-is Context

I have not observed any as-is process for this practice.

I.8.4.4 To-be Context

New actions are shown in green.

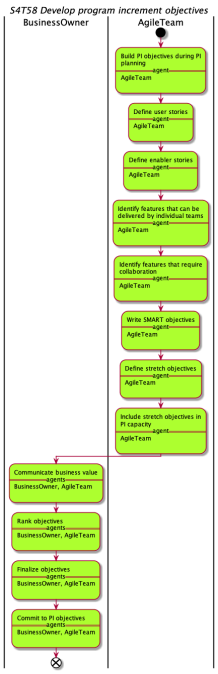

Figure I.91: S4T58 To-be context.

I.8.4.5 Stakeholders

Agile Team and Business Owner.

I.8.4.6 Entry and Exit Criteria

- Entry Criteria: Business Terms Glossary available, Features available, Program Vision available, Stories available, and Velocity available.
- Exit Criteria: Committed PI Objectives created or updated.

I.8.4.7 Implementation details

Table I.57: Implementation details for practice S4T58 – Develop program increment objectives.

Step	Description	Role(s)
Build PI objectives during PI planning	Build PI Objectives during PI Planning, ensuring the following is in place: (1) solid estimating and planning; (2) well understood velocity; (3) analysis of coming features; (4) a synthesis of simple business terms; (5) visibility of the Program Vision, new features, stories that need to be delivered.	Agile Team
Define user stories	Create user stories for the backlog as follows: (1) create short, simple descriptions of a small piece of desired functionality; (2) tell the story from the user's perspective; (3) use the user's language; (4) provide just enough information for the intent to be understood by both business and technical people; (5) create a story with independent behaviour that can be implemented incrementally and that provides some value to the user or the solution; (6) write stories on an index card or sticky note.	Agile Team
Define enabler stories	Define enabler stories to reflect the technical functionality needed to implement the user stories or support other components of the system. Write stories on an index card or sticky note.	Agile Team
Identify features that can be delivered by individual teams	If a feature can be delivered without collaboration with other teams, mark it as a 'within team' feature.	Agile Team

Continued on next page

Table I.57 – *Continued from previous page*

Step	Description	Role(s)
Identify features that require collaboration	If a feature can only be delivered through collaboration with another team or teams, mark it as an 'across team' feature.	Agile Team
Write SMART objectives	Provide a concise and (S)imple description of the intended outcome (usually starting with an action verb); (M)easure what a Team needs to do to achieve the Team objectives (can be descriptive, yes or no, or quantitative, or within a range); objectives should be (A)chievable i.e, within the Team's control and influence; (R)ecognize factors that cannot be controlled; (T)ime period for achievement must be within the PI, and all objectives must be scoped appropriately.	Agile Team
Define stretch objectives	Stretch objectives are objectives that go beyond what the Team can commit to; they allow for additional work to be completed if the team finishes its committed work before the end of the sprint.	Agile Team
Include stretch objectives in PI capacity	Include stretch objectives in the capacity of the PI; allow 10-15% of the total capacity for the stretch objectives. Constantly keep in mind that stretch objectives are used to identify what can be variable within the scope of a plan; therefore, use stretch objectives to synchronize a delivery to a cadence through capacity margins: the Team commits to a capacity that aligns with the cadence, then uses stretch objectives to allow the team to deliver more if they can. Recognize stretch objectives are not the way for stakeholders to load the teams with more than they can do.	Agile Team
Communicate business value Rank objectives	Business owner assigns business value to each of the Team's individual objectives in conversation with the team. Rank each objective according to the business owner ranking on a scale of 1 to 10. Communicate strategy and context behind the weighting decisions. Do not confuse business value with any other measure such as associated effort or total story points etc. In other words, don't calculate the business value, but rather assign business value.	Business Owner, Agile Team Business Owner, Agile Team

Continued on next page

Table I.57 – *Continued from previous page*

Step	Description	Role(s)
Finalize objectives	Finalize the Team PI Objectives with business value and stretch objectives.	Business Owner, Agile Team
Commit to PI objectives	Commit to PI Objectives by agreeing to do everything in the Team's power to meet the committed objectives.	Business Owner, Agile Team

I.8.5 Develop PI planning (S4PR100)

I.8.5.1 Purpose

The purpose of the program increment (PI) planning is to create an emergent roadmap to deliver a prioritized and agreed set of outcomes.

I.8.5.2 Background

"PI planning is a routine, face-to-face event with a standardized agenda that includes presentation of business context and vision, followed by team planning breakouts wherein the teams create the plans for the upcoming Program Increment (PI) (Leffingwell, 2015)."

The PI planning event aligns all the teams on the ART to a shared mission and vision, facilitated by the Release Train Engineer (RTE). All members of the Agile Release Train (ART) attend whenever possible. The result is a commitment to an agreed-to set of Program PI objectives for the next PI. When the teams contributing to the Agile Release Train are geographically distributed, the PI planning meeting takes place at multiple locations simultaneously, with frequent communication among both co-located and remote teams.

A successful PI planning event delivers two primary outputs: 1) Committed PI objectives-A set of SMART objectives that are created by each team with the business value assigned by the Business Owners. The product management uses the program PI objectives to update the roadmap and will improve the forecast for the next two PIs based on what was just learned. 2) Program board-This highlights the new feature delivery dates, feature dependencies among teams and with other ARTs, and relevant milestones. The program board is often used

during the Scrum of Scrums meetings to track dependencies, or it may not be maintained (manually) after that time. This depends upon the Agile project management tooling in place and the needs of the ART.

I.8.5.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

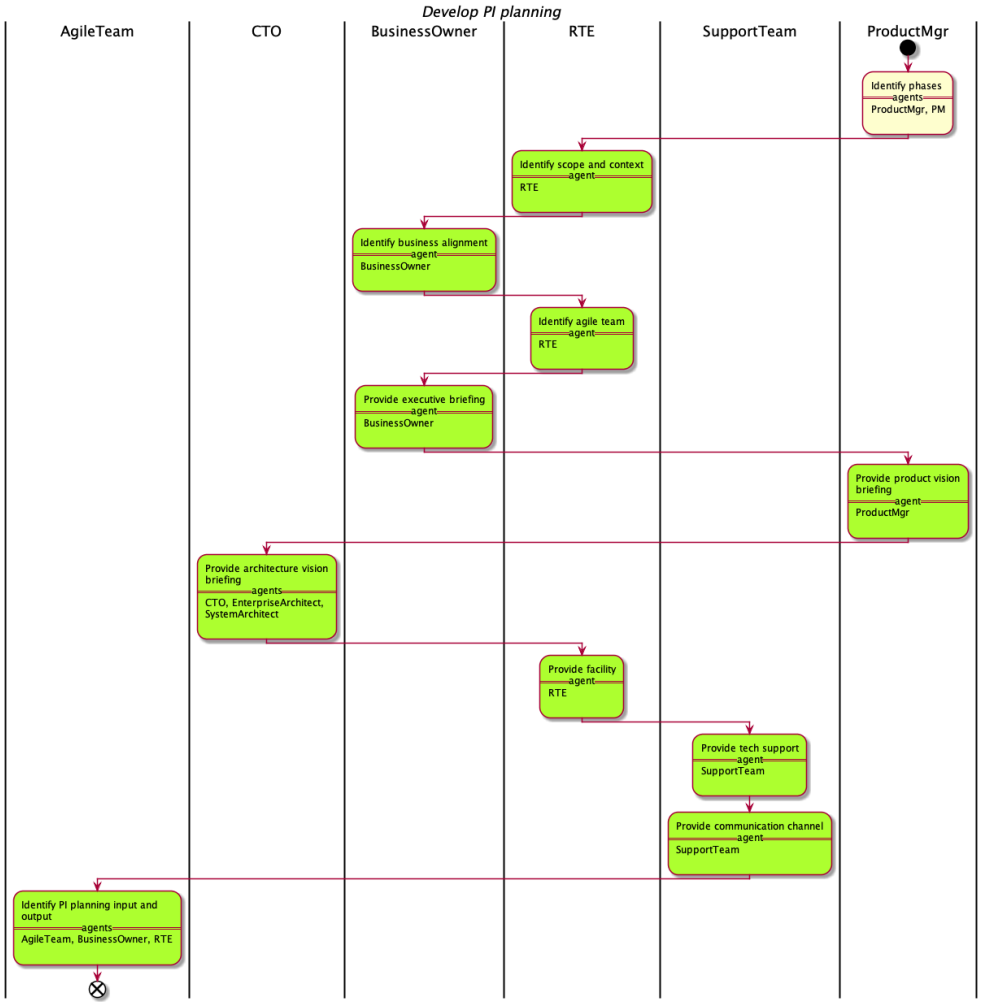

Figure I.92: S4PR100 To-be context.

I.8.5.4 Stakeholders

Product Mgr, PM, RTE, Business Owner, CTO, Enterprise Architect, System Architect, Support Team, and Agile Team.

I.8.5.5 Entry and Exit Criteria

- Entry Criteria: Customer To Be context, Prioritize list of Dev requirements
- Exit Criteria: Committed PI Objectives, Program board

I.8.5.6 Implementation details

Table I.58: Implementation details for practice S4PR100 – Develop PI planning.

Step	Description	Role(s)
Identify phases	PM or Product Manager divides the requirements into two phases. Phase 1 contains the requirements that are out of the box; phase 2 contains the bespoke development requirements.	Product Mgr, PM
Identify scope and context	The Release Train Engineer (RTE) identifies whether the scope of the planning process is understood across the train and which teams need to plan together.	RTE
Identify business alignment	Business Owners prioritize the list of topics they want to discuss or on-board during PI planning event.	Business Owner
Identify agile team	In preparation for the PI planning event, the RTE identifies the agile teams working on the agile release train, and dedicated resources such as developer, QA, Scrum Master, and Product Owner.	RTE
Provide executive briefing	A senior executive or line-of-business owner describes the current state of the business and presents a perspective on how well existing solutions are addressing current customer needs.	Business Owner

Continued on next page

Table I.58 – *Continued from previous page*

Step	Description	Role(s)
Provide product vision briefing	Product Management presents the current program vision (typically represented by the next top 10 upcoming features) and highlights any changes from the previous PI planning meeting, as well as any forthcoming Milestones.	Product Mgr
Provide architecture vision briefing	The CTO, Enterprise Architect, or System Architect present to communicate new Enablers, features, and Nonfunctional Requirements (NFRs). System Architect/Engineering also presents the architecture vision, and a senior development manager may introduce Agile-supportive changes to development practices, such as test automation, DevOps, Continuous Integration, and Continuous Deployment, which are being advanced in the upcoming PI.	CTO, Enterprise Architect, System Architect
Provide facility	The RTE confirms that there is enough space for all attendees including breakout room.	RTE
Provide tech support	The support team helps during setup and testing.	Support Team
Provide communication channel	The support team sets up primary and secondary audio, video, and presentation during distributed planning meetings.	Support Team
Identify PI planning input and output	A successful PI planning event delivers two primary outputs: 1) Committed PI objectives and 2) Program board	Agile Team, Business Owner, RTE

I.8.6 Use capability acceptance criteria (S4VS111)

I.8.6.1 Purpose

The purpose of developing an *Acceptance criteria* is to determine whether the implementation is correct and delivers the business benefits.

I.8.6.2 Background

In SAFe, each capability have associated acceptance criteria that further applied by the Solution Manager to determine whether the functionality has been properly implemented (Leffingwell, 2015).

I.8.6.3 As-is Context

I have not observed any as-os process model for this practice.

I.8.6.4 To-be Context

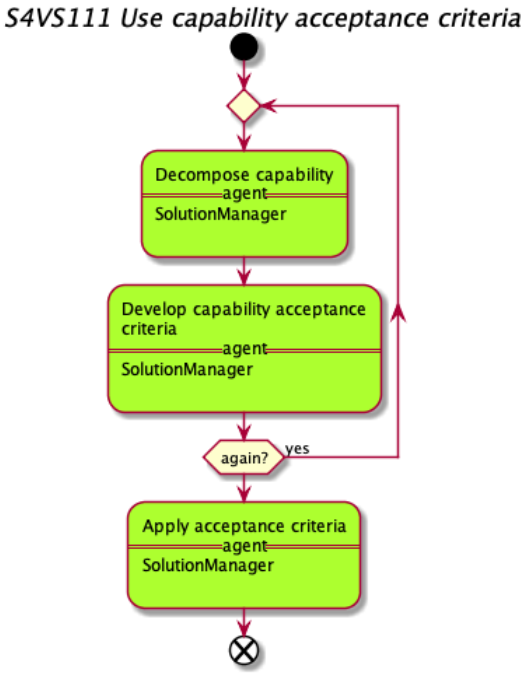

Figure I.93: S4VS111 To-be context.

I.8.6.5 Stakeholders

Solution Manager.

I.8.6.6 Entry and Exit Criteria

- Entry Criteria: Capability
- Exit Criteria: Accepted functionality

I.8.6.7 Implementation details

Table I.59: Implementation details for practice S4VS111 – Use capability acceptance criteria.

Step	Description	Role(s)
Decompose capability	Solution manager decomposes capability into a business capability or an enabler capability.	Solution Manager
Develop capability acceptance criteria	Solution manager develops acceptance criteria by providing a name, context, and a short description of the benefit to the user and the business.	Solution Manager
Apply acceptance criteria	Acceptance criteria is used to determine whether the implementation is correct and delivers the business benefits. Solution Managers, who use the acceptance criteria to determine whether the functionality has been properly implemented. To apply acceptance criteria at this stage all functionality at the feature level needed to be completed.	Solution Manager

I.8.7 PPM responsibility (S4PO25)

I.8.7.1 Purpose

Program Portfolio Management (PPM) represents the people who have the highest-level strategy and fiduciary decision-making responsibility (Leffingwell, 2015).

I.8.7.2 Background

PPM represents those individuals who have the primary responsibility for:

- Strategy and Investment Funding: to allocate and assure funding to strategy.
- Program Management: to derive, assist, or support program execution.
- Governance: to close the loop on funding and program execution, measures and reporting, necessary compliance.

PPM has the responsibility to:

- participate in the establishment of the strategic themes (that guide the enterprises investment and strategy);

- determine the relevant value stream;
- allocate budgets;
- define and prioritize crosscutting portfolio backlog epics;
- report to the business on investment spend;
- report progress via key performance indicators (KPIs).

I.8.7.3 As-is and To-be Context

Actions altered by this practice are shown in purple and new actions are shown in green.

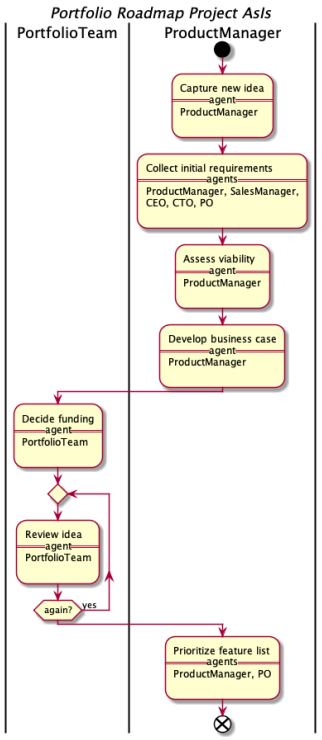

Figure I.94: S4PO25 As-is context.

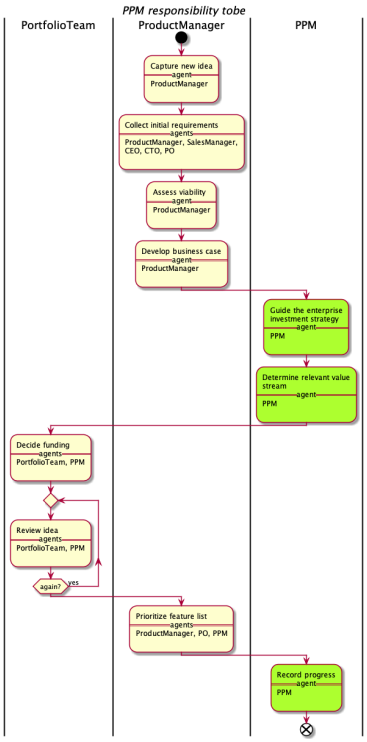

Figure I.95: S4PO25 To-be context.

I.8.7.4 Stakeholders

Product Manager, Sales Manager, CEO, CTO, PO, PPM, and Portfolio Team.

I.8.7.5 Entry and Exit Criteria

- Entry Criteria: New idea available
- Exit Criteria: Progress report created or updated

I.8.7.6 Implementation details

Table I.60: Implementation details for practice S4PO25 – PPM responsibility.

Step	Description	Role(s)
Capture new idea	The Product Manager maintains a funnel of possible new features and new integrations with other products, that are submitted by sales people, management, or developers. These may be ideas they learn from customers, at trade shows, or as a result of working on the product. It's possible for anyone to add an idea to the funnel, but in practice the Product Manager is the one who does this, based on input from other parts of the organization.	Product Manager
Collect initial requirements	The Product Manager collects and records initial requirements from the CEO, CTO, a sales person, or anyone from the company who can contribute. This starts with a very high level idea, then proceeds to initial review to see if it is viable.	Product Manager, Sales Manager, CEO, CTO, PO
Assess viability	The Product Manager does an initial assessment of the viability of a new idea.	Product Manager
Develop business case	For ideas that seem viable, the Product Manager develops a business case for review by the Portfolio Team. The business case may be as simple as an email message, or an elaborate presentation to the Portfolio Team.	Product Manager
Guide the enterprise investment strategy	Program Portfolio Management (PPM) participates in the solution portfolio strategy formulation meeting to guide the enterprise for investment strategy.	PPM

Continued on next page

Table I.60 – *Continued from previous page*

Step	Description	Role(s)
Determine relevant value stream	Based on the portfolio context and portfolio strategy PPM determines relevant value stream.	PPM
Decide funding	Based on the feasibility study (which includes the business case), the Portfolio Team decides whether to fund development or not. The CFO and DoD support the CEO during this process. PPM – in-collaboration with other portfolio stakeholders – allocates budget to each value stream.	Portfolio Team, PPM
Review idea	The Portfolio Team reviews the portfolio list once each quarter for roadmapping direction and add-ons. They also gather revenue information by comparing with the previous meeting. In this stage, the CTO and DoD also decide how each portfolio item is decomposed into smaller tickets, and identify in which phase/quarter that ticket will be delivered. Due to their scope and typically cross-cutting nature, epics usually require substantial investment and have considerable impact on both the development programs and business outcomes. Therefore, in order to reach the portfolio backlog, epics must first make their way through the portfolio kanban, where they are analyzed to determine feasibility and potential ROI. Epics that reach this boundary are in a mature state, in that they have been identified, elaborated, estimated and analyzed as necessary to achieve a 'go recommendation' from PPM.	Portfolio Team, PPM
Prioritize feature list	The Product Manager and Product Owner prioritize the roadmap feature list. A requirement would get prioritized if there is clear business case, such as value and clear return (revenue) for the company.	Product Manager, PO, PPM
Record progress	Finally, PPM reports to the business on investment spend and progress via Key Performance Indicators (KPIs).	PPM

I.8.8 Reduce time to market (S4PO125)

I.8.8.1 Purpose

The purpose of this practice is to provide an identifiable and measurable flow of value to a customer.

I.8.8.2 Background

The process of value stream mapping is an analytical process that helps the team to understand and improve time-to-market. The most important organizational construct in SAFe is the value stream, it provides an understanding from feature approval to development through to deployment, all the way to release. Moreover, this practice also provides an identifiable and measurable flow of value to a customer, as such, it can be systematically improved to increase delivery velocity and quality (Leffingwell, 2015).

To reduce the time-to-market with value stream mapping and make continuous improvement, SAFe enterprises can follow these steps (Leffingwell, 2015):

- From the receipt of a customer request to release, map the current state by identifying all the steps, value-added times, handoffs, and delays.
- Identify the largest sources of delays and handoffs as the feature moves through the system.
- Pick the biggest delay, and perform root-cause analysis.
- Create improvement backlog items to reduce the delay.
- Reduce batch sizes wherever possible.
- Implement the new improvement backlog items.
- Measure again, and repeat the process.

Using this process, the maturing Lean enterprise can systematically improve time-to-market.

I.8.8.3 As-is & To-be Context

I have not observed any as-is model for this practice. New actions are shown in green.

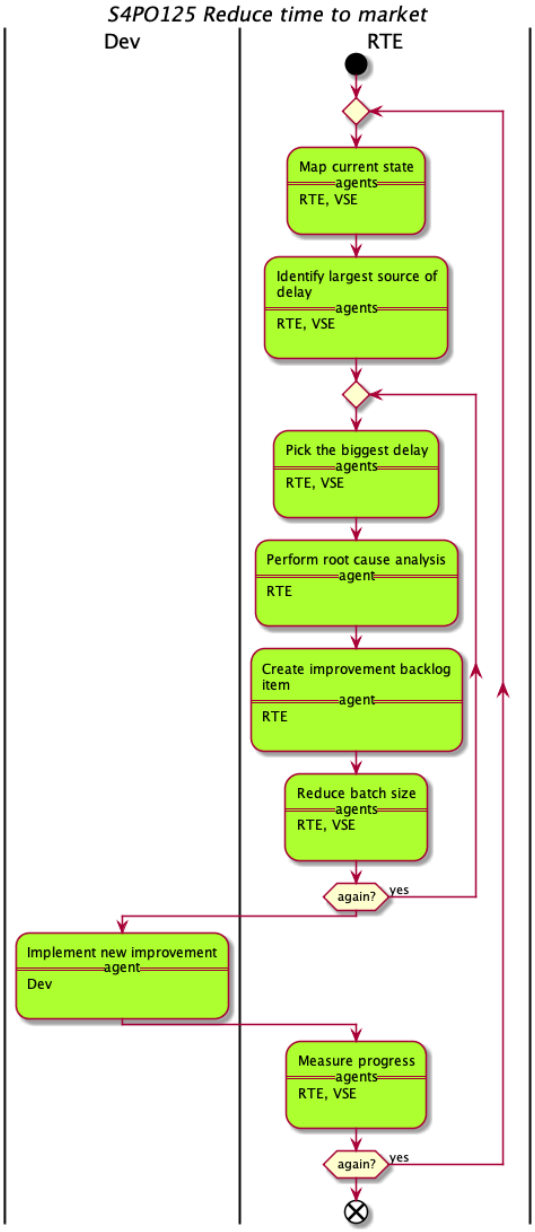

Figure I.96: S4PO125 To-be context.

I.8.8.4 Stakeholders

RTE, VSE, and Developer.

I.8.8.5 Entry and Exit Criteria

- Entry Criteria: Customer request, Release
- Exit Criteria: Progress report

I.8.8.6 Implementation details

Table I.61: Implementation details for practice S3PO125 – Reduce time to market.

Step	Description	Role(s)
Map current state	The Release Train Engineer (RTE) and Value Stream Engineer (VSE) map the current state by identifying all steps, value-added item, hand offs, and delays from the receipt of a customer request to release.	RTE, VSE
Identify largest source of delay	The Release Train Engineer (RTE) and Value Stream Engineer (VSE) identify the sources of delays and hand offs as the feature moves through the system.	RTE, VSE
Pick the biggest delay	The RTE or VSE picks the biggest delay.	RTE, VSE
Perform root cause analysis	SAFe recommends using Fish-bone Diagram tools to perform root cause analysis. Candidate issues are identified and then grouped into major categories as boned off the main bone. The Team pre-loads the main bones with the categories: People, Process, Tools, Program, and Environment. Team members then brainstorm factors that they think contribute to the problem to be solved. Once an issue/cause is identified, its root cause is identified with the five whys technique.	RTE

Continued on next page

Table I.61 – *Continued from previous page*

Step	Description	Role(s)
Create improvement backlog item	The Team votes on up to three most likely solutions. These will serve as improvement stories and features to feed directly into the PI Planning session that follows. During that session, the Release Train Engineer helps to ensure that the relevant improvement stories are loaded onto the Iteration plans, thus ensuring that action will be taken and resources allocated, as with any other backlog item.	RTE
Reduce batch size	The RTE and VSE reduce the batch size to increase the flow of the work i.e. requirements, designs, code, and test that move through the system. Small batches go through the system more quickly and with less variability that foster faster learning as well as faster learning. However, the economically optimal batch size depends on both the holding cost (the cost for delayed feedback, inventory decay, and delayed value delivery) and the transaction cost (the cost of preparing and implementing the batch).	RTE, VSE
Implement new improvement	A Developer takes the new improvements into the development phase to develop the actual functionality.	Dev
Measure progress	The RTE and VSE measure again to identify the progress.	RTE, VSE

I.8.9 Perform value stream coordination (S4PO124)

I.8.9.1 Purpose

The purpose of this practice is to ensure that the enterprise moves forward, with each value stream in lockstep with the enterprise objectives (Leffingwell, 2015).

I.8.9.2 Background

Value Stream coordination provides guidance to manage dependencies and exploit the opportunities in a portfolio (Leffingwell, 2015). The dependencies can be divided into three stages:

- What gets built : Product Owner > Product Management > Solution Management
- How it gets built : Agile Team > System Architect > Solution Architect
- Operation and Execution : Scrum Master > Release Train Engineer (RTE) > Solution Train Engineer (STE)

However, when a significant degree of coordination is required, the following additional roles appear in large portfolios:

- Solution Portfolio Management
- Enterprise Architect
- Agile Program Management Office (APMO)

I.8.9.3 As-is & To-be Context

I have not observed any as-is process model for this practice. New actions are shown in green.

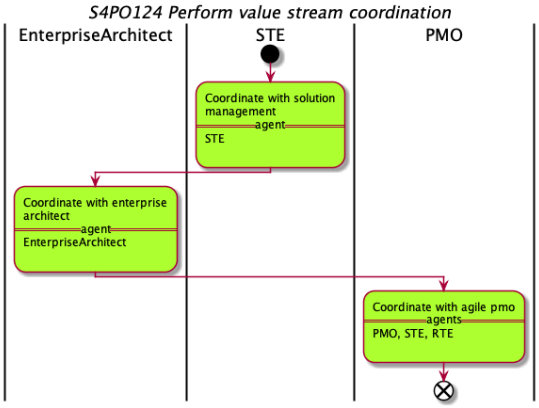

Figure I.97: S4PO124 To-be context.

I.8.9.4 Stakeholders

STE, Enterprise Architect, PMO, and RTE.

I.8.9.5 Entry and Exit Criteria

- Entry Criteria: Solution
- Exit Criteria: Decentralized PI

I.8.9.6 Implementation details

Table I.62: Implementation details for practice S4PO124 – Perform value stream coordination.

Step	Description	Role(s)
Coordinate with solution management	The Solution Train Engineer (STE) has the overall responsibility for guiding a portfolio to a set of integrated solutions.	STE
Coordinate with enterprise architect	The Enterprise Architect provides technical guidance for the long-term evolution of the technologies, platforms, and the larger non-functional requirements (security, compliance, and performance) for the portfolio solution set.	Enterprise Architect
Coordinate with agile pmo	The Agile PMO, along with STEs and RTEs, is responsible for supporting decentralized program execution. The Agile PMO also provides support for standard reporting patterns, shared best practices, and growth and dissemination of institutional knowledge.	PMO, STE, RTE

I.8.10 Develop portfolio kanban (S4PO80)

I.8.10.1 Purpose

The purpose of the Portfolio Kanban system is to capture, analyze, approve, and track epics.

I.8.10.2 Background

Overall, there are a number of stages that an epic passes through on the way to implementation or rejection. The typical collaborators of the portfolio kanban system are: Business Owner, Product and Solution Management, Epic Owner,

Development Team, and Solution/System Architect. There are in total six phases in the Portfolio Kanban system (Leffingwell, 2015).

- **Funnel** – The funnel queue is the capture queue where all ideas come from different sources i.e. portfolio strategic theme. At this stage the epic does not need to have any business case or estimates; it can be stated as a short keyword or phrase. All ideas are captured for consideration and can be represented in different formats, i.e. a document, spreadsheet, or visual system on the wall. Program Portfolio Management (PPM) typically discusses funnel epics on a periodic cadence, and epics that meet the decision criteria are moved to the review queue.

- **Review** – In this state, ideas get preliminary estimates of opportunity, effort, and cost of delay.

- **Analysis** – At this stage, more work is done to establish viability, measurable benefits, development and deployment impact, and potential availability of resources. A lightweight business case is developed and the epic is either approved or rejected at the end of this state.

- **Portfolio Backlog** – Epics with a "go" decision are kept in the portfolio backlog and reviewed on a periodic basis. The queue in the portfolio backlog represents a low-cost holding pattern for upcoming implementation work. Epics are moved to the implementing queue when there is sufficient capacity on one or more value streams or Agile Release Trains.

- **Implementation** – When the capacity becomes available then epics are transitioned to either the Program or Value Stream Kanban, where implementation begins.

- **Done** – The Epic Owner checks the implemented epic against epic success criteria. The epic is considered done when it has met all its success criteria. However, due to the scope of epics, completion to the original intent is not always the desired case. At this stage, some identified capabilities and features might be discarded. Regardless, the epic eventually reaches a done state.

I.8.10.3 As-is Context

Actions altered by this practice are shown in purple.

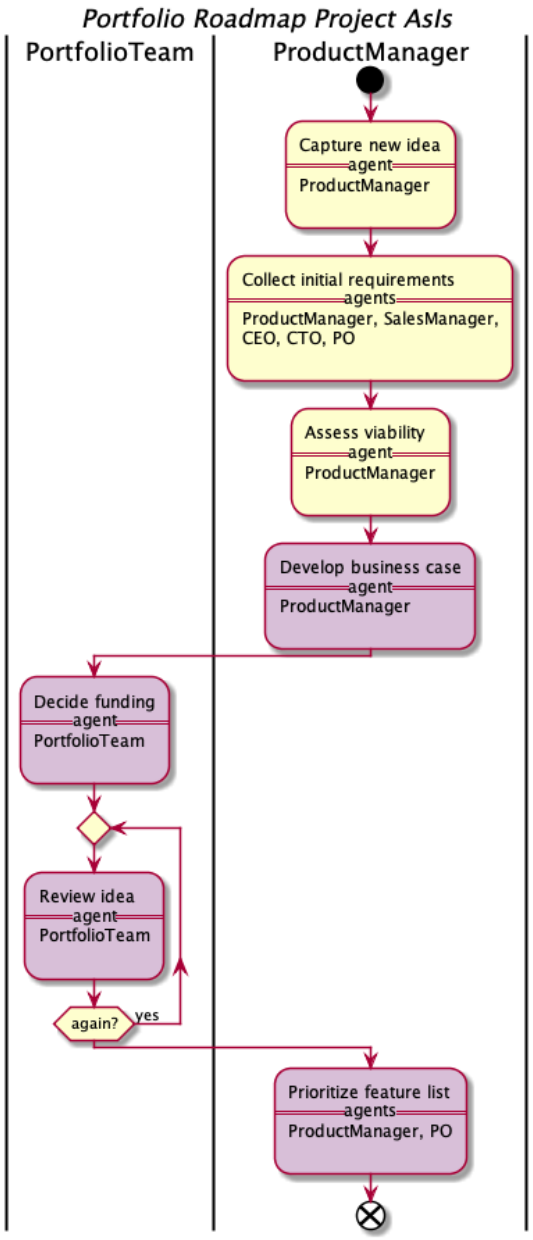

Figure I.98: S4PO80 As-is context.

I.8.10.4 To-be Context

New actions are shown in green.

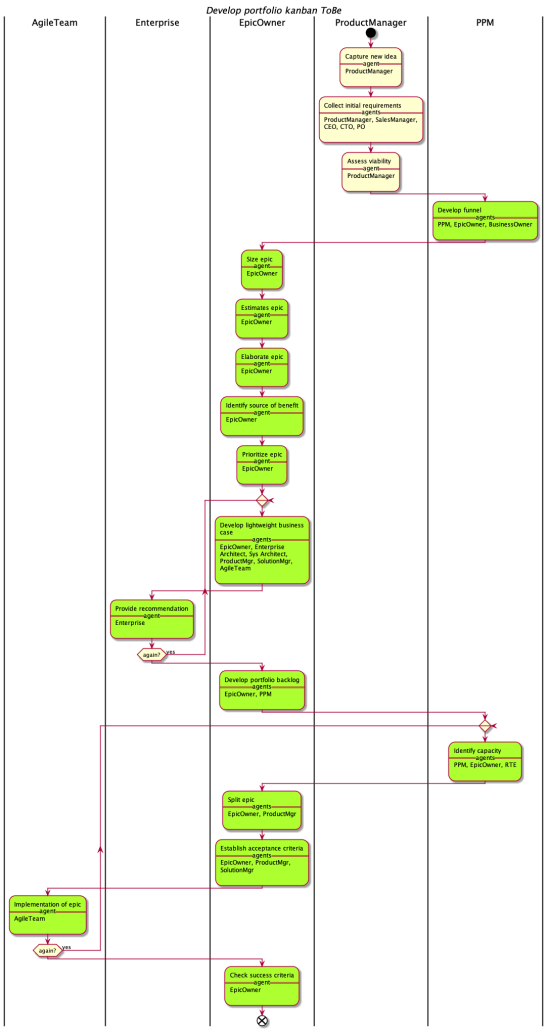

Figure I.99: S4PO80 To-be context.

I.8.10.5 Stakeholders

Sales Manager, CEO, CTO, PO, PPM, Epic Owner, Business Owner, Enterprise Architect, Sys Architect, Product Mgr, Solution Mgr, Agile Team, Enterprise, and RTE.

I.8.10.6 Entry and Exit Criteria

- Entry Criteria: New idea
- Exit Criteria: Done Epic

I.8.10.7 Implementation details

Table I.63: Implementation details for practice S4PO80 – Develop portfolio kanban.

Step	Description	Role(s)
Capture new idea	The Product Manager maintains a funnel of possible new features and new integrations with other products, that are submitted by sales people, management, or developers. These may be ideas learned from customers, at trade shows, or as a result of working on the product. It's possible for anyone to add an idea to the funnel, but in practice the Product Manager is the one who does this, based on input from other parts of the organization.	Product Mgr
Collect initial requirements	The Product Manager collects and records initial requirements from the CEO, CTO, a sales person, or anyone from the company who can contribute. This starts with a very high level idea, then proceeds to initial review to see if it is viable.	Product Mgr, Sales Manager, CEO, CTO, PO
Assess viability	The Product Manager does an initial assessment of the viability of a new idea.	Product Mgr
Develop funnel	The funnel queue is the capture queue where all ideas come from different sources, such as the portfolio strategic themes. At this stage an epic does not need to have any business case or estimates. It can be stated as a short keyword or phrase. All ideas are captured for consideration and can be represented in different formats, i.e. a document, spreadsheet, or visual system on the wall. Program Portfolio Management (PPM) typically discusses funnel epics on a periodic cadence, and epics that meet the decision criteria are moved to the review queue.	PPM, Epic Owner, Business Owner

Continued on next page

Table I.63 – *Continued from previous page*

Step	Description	Role(s)
Size epic	The Epic Owner roughly sizes the epic.	Epic Owner
Estimates epic	The Epic Owner estimates size of the epic in terms of epic value.	Epic Owner
Elaborate epic	The Epic Owner elaborates the epic using the epic value statement format. This value stream statement can be used to capture, organize, and communicate key information about an epic. There are two parts to the value statement template: 1) Forward-Looking position statement: for (customer), who (do something), the (solution), is a (something–the how), that (provides this value), unlike (competitor, current solution, or non-existing solution), our solution (does something better–the why) 2) Scope: Success criteria, In scope, Out of scope, and NFRs.	Epic Owner
Identify source of benefit	The Epic Owner identifies the business benefits from epic value statement.	Epic Owner
Prioritize epic	The Epic Owner prioritizes the epic based on the needs or business benefits.	Epic Owner
Develop lightweight business case	For ideas that seem viable, the Product Manager develop a business case for review by the Portfolio Team. The business case may be as simple as an email message, or an elaborate presentation to the Portfolio Team. An Epic Owner takes responsibility for the ongoing work. Then an action collaboration is initiated among Enterprise Architects, System Architects, Agile Teams, Product and Solution Management, and key stakeholders on the agile release train. Through this collaboration the key stakeholders also explore solution, design, and implementation alternatives. The options for internal and (or) outsourcing development are considered.	Epic Owner, Enterprise Architect, Sys Architect, Product Mgr, Solution Mgr, Agile Team
Provide recommendation	The Enterprise provides a go or no-go recommendation based on the business case and resource availability (i.e. business analyst, development team, and enterprise architect.)	Enterprise

Continued on next page

Table I.63 – *Continued from previous page*

Step	Description	Role(s)
Develop portfolio backlog	Epic with a 'go' decision is kept in the portfolio backlog and reviewed on a periodic basis. The queue in the portfolio backlog represents a low-cost holding pattern for upcoming implementation work. Epics are moved to the implementing queue when there is sufficient capacity from one or more value stream or Agile Release Trains.	Epic Owner, PPM
Identify capacity	PPM or Epic Owner identifies the capacity in the value stream. As capacity becomes available, epics are pulled into the relevant value stream or program kanban.	PPM, Epic Owner, RTE
Split epic	The epics are split into capabilities and features.	Epic Owner, Product Mgr
Establish acceptance criteria	Epic Owner, Solution Manager, and Product Manager develop acceptance criteria for capabilities and features.	Epic Owner, Product Mgr, Solution Mgr
Implementation of epic	Finally, the Team performs the actual development.	Agile Team
Check success criteria	The Epic Owner checks the implemented epic against epic success criteria. The epic is considered done when it has met all its success criteria. However, due to scope of epics, completion to the original intent is not always the desired case. At this stage, some identified capabilities and features might be discarded. Regardless, the epic reaches a done state.	Epic Owner

I.9 Roadmap V

Collaboratively plan, develop and validate systems and changes.

I.9.1 GTM Context

Fig. I.100 provides GTM context of the recommendation V.

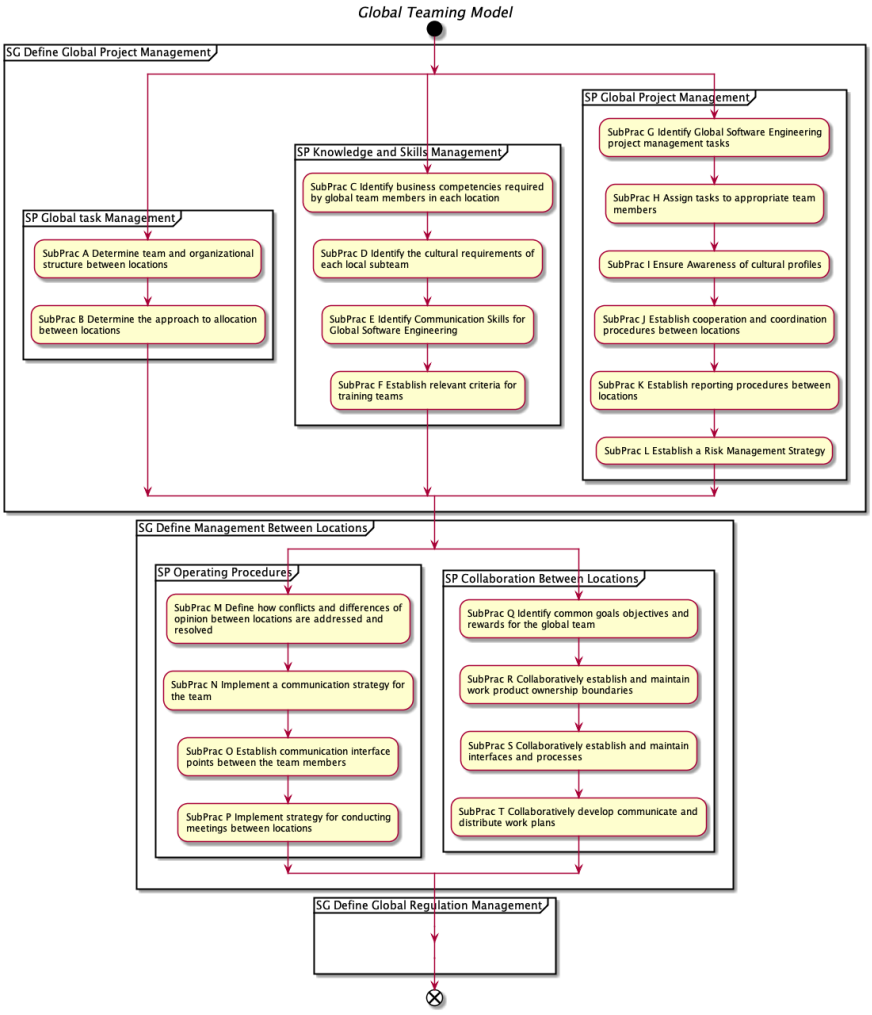

Figure I.100: V GTM context.

I.9.2 Why implement this recommendation?

From observations and interviews:

- There is no visibility of the release planning at the Program level.
- Lack of synchronisation in the product line
- Country specific regulations
- Geographical distance hinder regulation especially functionality
- Regulation has an impact on technical requirements

I.9.3 Perform continuous integration (S4T272)

I.9.3.1 Purpose

Conflict between teams can arise when one team checks-in code to fix or enhance a specific product version, that breaks the build of other targets. Continuous integration identifies such issues as soon as code is committed, by building for all targets: any target build that fails is identified immediately.

In SAFe, the continuous integration practice includes not only the technical infrastructure and process to achieve continuous integration, but also the culture to ensure failed integrations are fixed as soon as possible.

I.9.3.2 Background

Continuous Integration (CI) has two purposes: first, to reduce risk by identifying integration problems as soon as they arise; and second, to enable fast but sustainable development, by making new functionality that might be required by other teams available as soon as it is completed.

I.9.3.3 As-is Context

Actions altered by this practice are shown in purple.

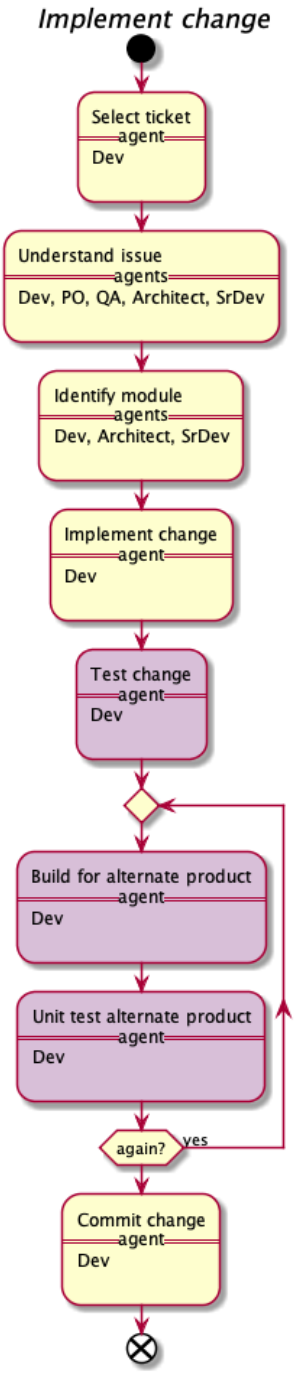

Figure I.101: S4T272 As-is context.

I.9.3.4 To-be Context

New actions are shown in green.

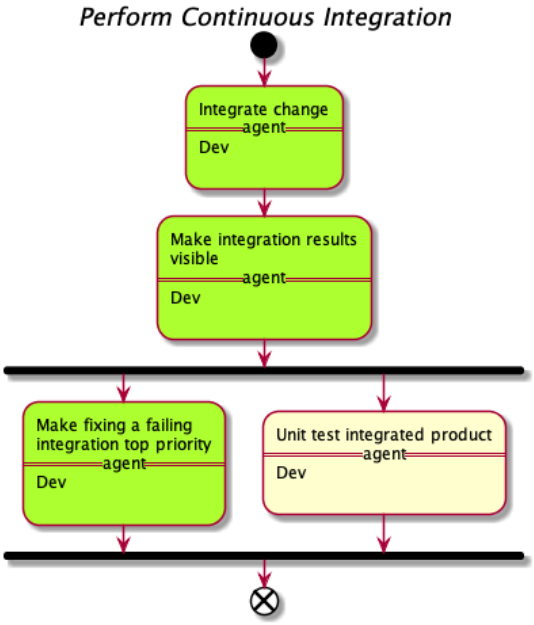

Figure I.102: S4T272 To-be context.

I.9.3.5 Stakeholders

Developer.

I.9.3.6 Entry and Exit Criteria

- Entry Criteria: PI Objective
- Exit Criteria: Improvement story

I.9.3.7 Implementation details

Table I.64: Implementation details for practice S4T272 – Perform continuous integration.

Step	Description	Role(s)
Integrate change	Commit kicks off automated build and continuous integration mechanism to build for all target platforms and configurations.	Dev
Make integration results visible	When the integration process breaks, everybody should know how and why it broke. And when it's fixed, they should know what fixed it, too.	Dev
Make fixing a failing integration top priority	When a build for any target platform fails, the team stops and fixes it immediately. Teams use flashing lights when a build is broken, and highly visible indicators of percentages of the time the system is broken.	Dev
Unit test integrated product	Developer unit-tests modified uni	Dev

I.9.4 Develop PI commitment (S4PR13)

I.9.4.1 Purpose

The purpose of PI commitment is to develop a near-term plan that impacts the upcoming functionality.

I.9.4.2 Background

PI Objectives are a summarised description of the specific business and technical goals that an Agile Team, Agile Release Train (ART), or Value Stream intends to achieve in the upcoming Program Increment. PI objectives are formulated during PI Planning and, where applicable, Post-PI Planning. During PI planning, each team reviews the vision and input objectives, defines the initial stories, and plans them into iterations until their capacity is full. Based on the capacity, the Agile Team commits for the upcoming program increment.

I.9.4.3 As-is Context

I have not observed any as-is model for this practice.

I.9.4.4 To-be Context

New actions are shown in green.

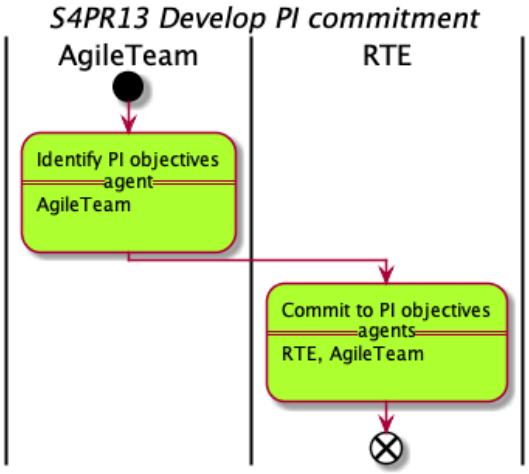

Figure I.103: S4PR13 To-be context.

I.9.4.5 Stakeholders

Agile Team, RTE.

I.9.4.6 Entry and Exit Criteria

- Entry Criteria: Upcoming PI plan
- Exit Criteria: Committed PI objectives

I.9.4.7 Implementation details

Table I.65: Implementation details for practice S4PR13 – Develop PI commitment.

Step	Description	Role(s)
Identify PI objectives	PI objectives are formulated by the Agile Team during the PI Planning meeting, indicating what they will have ready by the end of the PI. As SAFe PI objectives is build bottom up, so the objectives created by the Team are aggregated up to the Program level and aggregated again to the value stream level.	Agile Team
Commit to PI objectives	During the PI planning, each Agile Team reviews the vision, input objectives, defines the initial stories, and plans them into iterations until their capacity is full. The Agile team commit to the set of PI objectives based on the capacity for upcoming PI. This PI planning meeting is facilitated by the RTE and member of the ART is participate whenever possible.	RTE, Agile Team

I.9.5 Forecast PIs (S4PR15)

I.9.5.1 Purpose

The purpose of forecasting multiple program increments is to respond to changing markets and planning for the future with reasonable accuracy.

I.9.5.2 Background

Release Train Engineers (RTEs) apply the feature size (from program backlog) and ART velocity to develop a roadmap that forecasts several program increments or business quarters (Leffingwell, 2015).

"RTE use Agile Release Train (ARTs) velocities, the PI predictability measure, relative priorities, and the history of how much work is devoted to maintenance and other business-as-usual activities to forecast PIs. ARTs can generally lay the future features into the roadmap without too much difficulty" (Leffingwell, 2015)..

I.9.5.3 As-is Context

Actions altered by this practice are shown in purple.

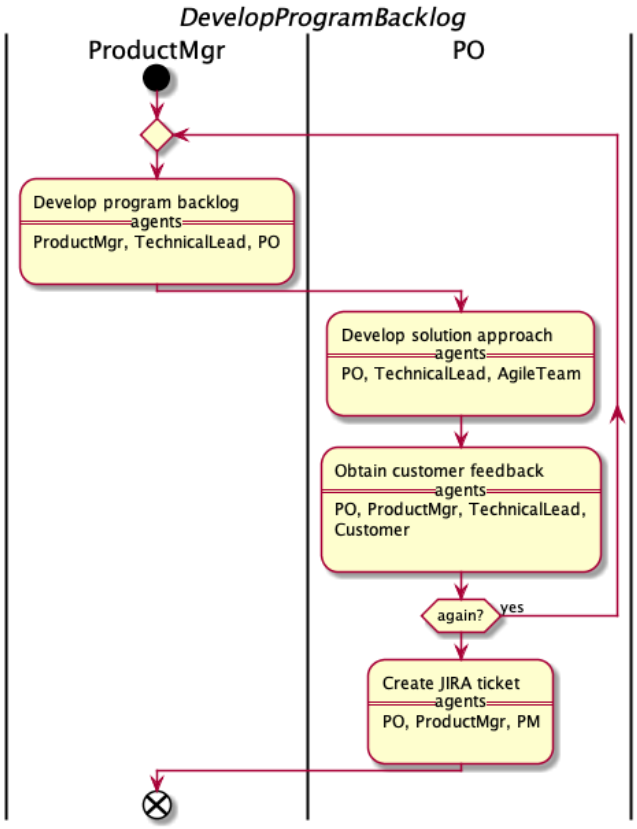

Figure I.104: S4PR15 As-is context.

I.9.5.4 To-be Context

New actions are shown in green.

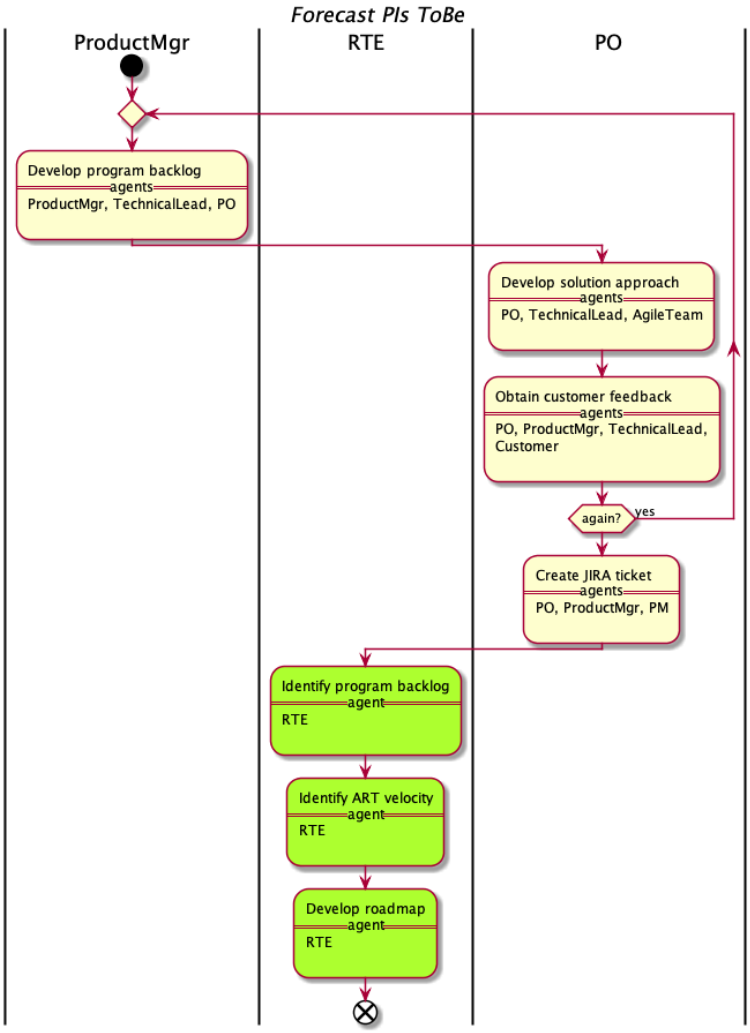

Figure I.105: S4PR15 To-be context.

I.9.5.5 Stakeholders

Product Mgr, Technical Lead, PO, Agile Team, Customer, PM, and RTE.

I.9.5.6 Entry and Exit Criteria

- Entry Criteria: List of development requirements
- Exit Criteria: Roadmap

I.9.5.7 Implementation details

Table I.66: Implementation details for practice S4PR15 – Forecast PIs.

Step	Description	Role(s)
Develop program backlog	Product Manager or PO and Technical lead explore requirements to identify development needs.	Product Mgr, Technical Lead, PO
Develop solution approach	Product Owner goes through each of the development needs with the Technical Lead to create a specification.	PO, Technical Lead, Agile Team
Obtain customer feedback	Product Manager and Technical Lead send the solution approach to the Customer. If they agree, then the specification is logged into JIRA.	PO, Product Mgr, Technical Lead, Customer
Create JIRA ticket	Product Manager, PO, or Project Manager creates tickets in JIRA for each agreed requirement. Each ticket contains an initial estimate, time, and release plan (in which phase specific ticket will be released).	PO, Product Mgr, PM
Identify program backlog	Release Train Engineer (RTE) identifies the program backlog to check whether each feature contains acceptance criteria and preliminary estimates for size in story points.	RTE
Identify ART velocity	RTE identifies the ART velocity by looking into the number of resources they have in each train.	RTE
Develop roadmap	RTE uses ART velocities, the PI predictability measure, relative priorities, and the history of how much work is devoted to maintenance and other business-as-usual activities, to forecast PIs. ARTs can generally lay the future features into the roadmap without too much difficulty. The result is that most trains have roadmaps with a reasonable degree of confidence over about a three-PI period.	RTE

I.9.6 Estimate longer term initiatives (S4PR19)

I.9.6.1 Purpose

The purpose of estimating the longer term initiatives is to forecast longer-term work.

I.9.6.2 Background

Usually, in SAFe, the Portfolio Team estimates the larger initiatives at the epic level. An epic is defines as "a container for a Solution development initiative large enough to require analysis, the definition of a Minimum Viable Product (MVP), and financial approval before implementation." Epics are broken down into potential features during the Portfolio Kanban analysis. Then features are estimated in story points. Finally, feature estimates are aggregated back into the epic estimate as part of the lightweight business case (Leffingwell, 2015).

I.9.6.3 As-is Context

Actions altered by this practice are shown in purple.

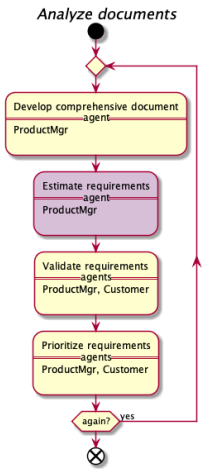

Figure I.106: S4PR19 As-is context.

I.9.6.4 To-be Context

New actions are shown in green.

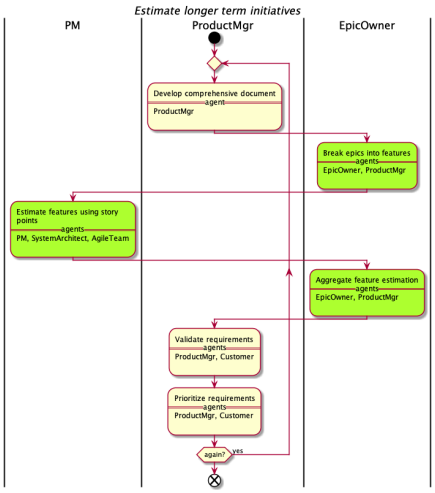

Figure I.107: S4PR19 To-be context.

I.9.6.5 Stakeholders

Product Mgr, Epic Owner, PM, System Architect, Agile Team, and Customer.

I.9.6.6 Entry and Exit Criteria

- Entry Criteria: Existing system
- Exit Criteria: Feedback

I.9.6.7 Implementation details

Table I.67: Implementation details for practice S4PR19 – Estimate longer term initiatives.

Step	Description	Role(s)
Develop comprehensive document	Product Manager develops a comprehensive document of the list of requirements based on the Customer's to-be context. The Product Manager provides a detailed functional requirement document to the Customer, showing what is implementable and the gap between the existing product and the to-be product.	Product Mgr
Break epics into features	Epic Owner and Product Manager split epics into potential features during the Portfolio Kanban analysis stage.	Epic Owner, Product Mgr
Estimate features using story points	Potential features are estimated in story points typically by the PM and System Architect based on history and relative size. Individual teams are engaged as necessary.	PM, System Architect, Agile Team
Aggregate feature estimation	At this stage, feature estimates are aggregated back into the epic estimate as part of the lightweight business case.	Epic Owner, Product Mgr
Validate requirements	Customer validates the list of development requirements.	Product Mgr, Customer
Prioritize requirements	Customer prioritizes the list of development requirements.	Product Mgr, Customer

I.9.7 Avoid the queue (S4PR21)

I.9.7.1 Purpose

In theory, the waiting time for a service is calculated based on the ratio of the length of the queue divided by the average processing rate. That means, as the length of the queue increases, so does the waiting time.

I.9.7.2 Background

In solution development, if the team is waiting to implement sometime in the queue then the waiting time gets longer, no matter how efficient the team is in processing the work. So, to provide a faster service, the team either needs to reduce the length of the queue or increase the processing time. The team can manage queue lengths by keeping the backlog short and largely uncommitted.

Overall, reducing the queue lengths decreases delays, reduces waste, and increases predictability of outcomes.

I.9.7.3 As-is Context

I have not observed any as-is for this practice.

I.9.7.4 To-be Context

New actions are shown in green.

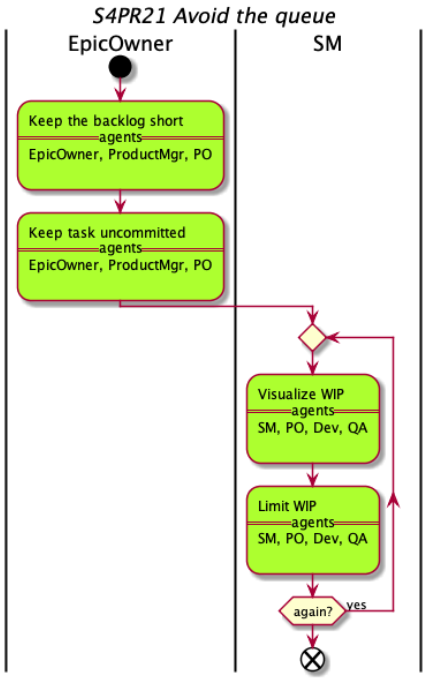

Figure I.108: S4PR21 To-be context.

I.9.7.5 Stakeholders

Epic Owner, Product Mgr, PO, SM, Dev, and QA.

I.9.7.6 Entry and Exit Criteria

- Entry Criteria: Epic or Feature or Story
- Exit Criteria: Work in progress (WIP)

I.9.7.7 Implementation details

Table I.68: Implementation details for practice S4PR21 – Avoid the queue.

Step	Description	Role(s)
Keep the backlog short	To reduce waiting time, the Team keeps the backlog short.	Epic Owner, Product Mgr, PO
Keep task uncommitted	The Team keeps tasks largely uncommitted because if the team is fully committed and running at nearly full capacity, then the wait time for a new capability/feature/story will increase.	Epic Owner, Product Mgr, PO
Visualize WIP	The first step is to make the current work in progress (WIP) visible to all stakeholders. This visualization illustrates the total amount of work at each step (Team Backlog, Development, Testing, and Accepted) and also serves as an initial process diagnostic, showing the current bottlenecks.	SM, PO, Dev, QA
Limit WIP	The Scrum Master starts balancing the amount of work in progress against the available development capacity. If any step reaches its WIP limit, no new work is taken on.	SM, PO, Dev, QA

I.9.8 Identify regulated milestones (S4PR47)

I.9.8.1 Purpose

The purpose of this practice is to identify the relevant rules and regulations to avoid risk resulting from non-compliance.

I.9.8.2 Background

Requirements elicitation is one of the integral parts of the requirement engineering as the requirements can become “useless” if the requirement does not comply with the government’s rules and regulations. Moreover, maintenance of the regulatory requirements has become a great challenge to the organization as avoidance or negligence of the rules may cause problems.

However, in SAFe, the "Release Train Engineer (RTE) identifies the milestones that need to be achieved for the economic success of product development, such as filing patents, certifying the system, and auditing specific regulatory requirements. In many instances these milestones influence content or priorities of work; they may even alter the development process itself. For example, the need to perform solution certification may increase the transaction cost of accepting a new Release into production and may drive the system builder to seek alternative ways of acquiring feedback before release.” (Leffingwell, 2015).

I.9.8.3 As-is Context

I have not observed any as-is model for this practice.

I.9.8.4 To-be Context

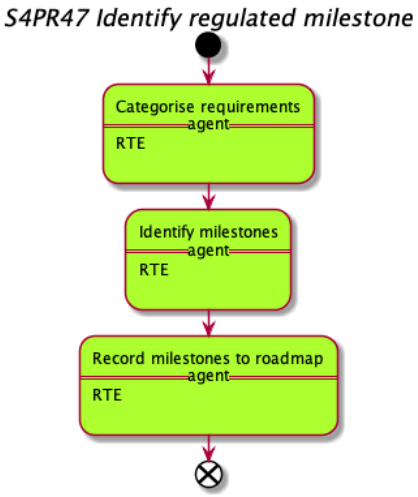

Figure I.109: S4PR47 To-be context.

I.9.8.5 Stakeholders

RTE.

I.9.8.6 Entry and Exit Criteria

- Entry Criteria: Requirements
- Exit Criteria: Roadmap milestones

I.9.8.7 Implementation details

Table I.69: Implementation details for practice S4PR47 – Identify regulated milestones.

Step	Description	Role(s)
Categorise requirements	RTE is responsible for categorizing requirements as functional, non-functional, and regulatory requirements.	RTE
Identify milestones	RTE identifies the milestones that need to be achieved for economic success of product development, such as filing patents, certifying the system, and auditing specific regulatory requirements. In many instances these milestones influence content or priorities of work; they may even alter the development process itself. For example, the need to perform solution certification may increase the transaction cost of accepting a new Release into production and may drive the system builder to seek alternative ways of acquiring feedback before release.	RTE
Record milestones to roadmap	RTE records or places the milestones on the relevant roadmap.	RTE

I.9.9 Develop program backlog (S4PR76)

I.9.9.1 Purpose

The purpose of having a program backlog is to store all upcoming work that affects the behavior of the solution.

I.9.9.2 Background

One of the core responsibilities of the product management team is to develop, maintain, and prioritize the program backlog. However, the items in the backlog arise from the various stakeholders such as Customers, Business Owners, Product Management, Product Owners, and Architects (Leffingwell, 2015).

I.9.9.3 As-is Context

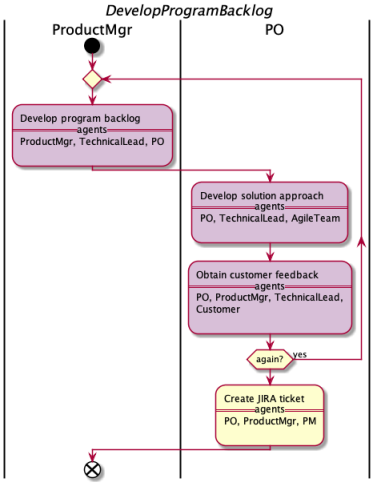

Figure I.110: S4PR76 As-is context.

I.9.9.4 To-be Context

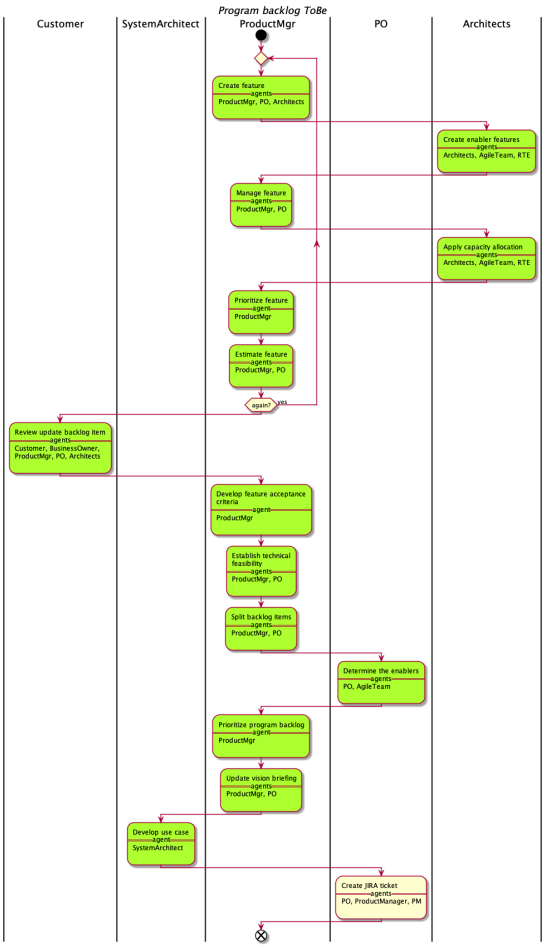

Figure I.111: S4PR76 To-be context.

I.9.9.5 Stakeholders

Product Mgr, PO, Architects, Agile Team, RTE, Customer, Business Owner, System Architect, Product Manager, and PM.

I.9.9.6 Entry and Exit Criteria

- Entry Criteria: Epic, Portfolio backlog
- Exit Criteria: Product backlog

I.9.9.7 Implementation details

Table I.70: Implementation details for practice S4PR76 – Develop program backlog.

Step	Description	Role(s)
Create feature	In collaboration with Product Owners and other key stakeholders, features are created by Product Managers in the local context of an ART. Enabler features pave the Architectural Runway, support exploration, or may provide the infrastructure needed to develop, test, and integrate the initiative. Enabler features are generally created by Architects or Engineers and maintained in the program backlog alongside business features.	Product Mgr, PO, Architects
Create enabler features	Architects or Engineers create enabler features to pave the architectural runway, support exploration, or to provide the infrastructure needed to develop, test, and integrate the initiative. Enabler features are maintained in the program backlog alongside business features.	Architects, Agile Team, RTE
Manage feature	Through a program backlog Product Manager manages all features.	Product Mgr, PO
Apply capacity allocation	Apply capacity allocation for the enabler to work as a whole or to differentiate between various types of enablers.	Architects, Agile Team, RTE
Prioritize feature	Prioritize features in the program backlog using weighted shortest job first (WSJF).	Product Mgr
Estimate feature	Feature estimation usually occurs in the 'refinement' state of the Program Kanban and relies on normalized estimation techniques, equivalent to the approach used by Agile Teams for estimating stories. Feature estimation at this point, however, does not require full breakdown into stories or involve all the teams that possibly will be included in feature development. Instead, select subject matter experts may be engaged in basic exploration and sizing.	Product Mgr, PO

Continued on next page

Table I.70 – *Continued from previous page*

Step	Description	Role(s)
Review update backlog item	The Product Manager reviews and updates backlog items in preparation for the next PI planning.	Customer, Business Owner, Product Mgr, PO, Architects
Develop feature acceptance criteria	In preparation of the PI planning event, the Product Manager also develops acceptance criteria for each feature in the program backlog.	Product Mgr
Establish technical feasibility	Product Manager and PO work with the team to establish technical feasibility and scope estimates.	Product Mgr, PO
Split backlog items	The Product Management team analyzes ways to split backlog items into smaller chunks of incremental value. If the feature is big, then the team split that feature into multiple features otherwise a story.	Product Mgr, PO
Determine the enablers	PO and Agile Team Determine the Enablers spawned by new features and capabilities, and establish their capacity allocation.	PO, Agile Team
Prioritize program backlog	Product Management uses the WSJF prioritisation method for job sequencing.	Product Mgr
Update vision briefing	Product Management does final backlog preparation, updates the vision briefings, and works with Product Owners to further socialize the backlog prior to the event.	Product Mgr, PO
Develop use case	System Architects/Engineering update enabler definitions and models and often develop use cases that illustrate how the features and capabilities work together to deliver end user value.	System Architect
Create JIRA ticket	Product Manager, PO, or Project Manager creates tickets in JIRA for each agreed requirement. Each ticket contains an initial estimate, time, and release plan (in which phase specific ticket will be released).	PO, Product Manager, PM

I.9.10 Develop solution intent (S4VS70)

I.9.10.1 Purpose

The purpose of this practice is to define the key solution behaviours and requirements that can be used to validate and verify the delivered solution.

I.9.10.2 Background

Solution intent is the repository for sorting, managing, and communicating the knowledge of current and intended solution behaviour. This includes both fixed and variable specifications and design; references to applicable standards, system models, and functional and nonfunctional tests; and traceability.

Building a large-scale software and cyber-physical systems is complex and challenging task. It requires alignment on two central questions:

- What exactly is this we are building?
- How are we going to build it?

The solution intent begins with a vision, critical nonfunctional requirements, roadmap, and critical milestone; that provide sufficient guidance to the teams for initial PI planning and execution. The teams use feature, capabilities, stories, and enablers to define and realise the system behaviour. Moreover, if you are dealing with the regulated environment then in solution intent is required.

I.9.10.3 As-is Context

Actions altered by this practice are shown in purple.

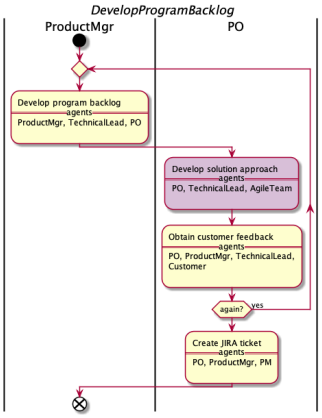

Figure I.112: S4VS70 As-is context.

I.9.10.4 To-be Context

New actions are shown in green.

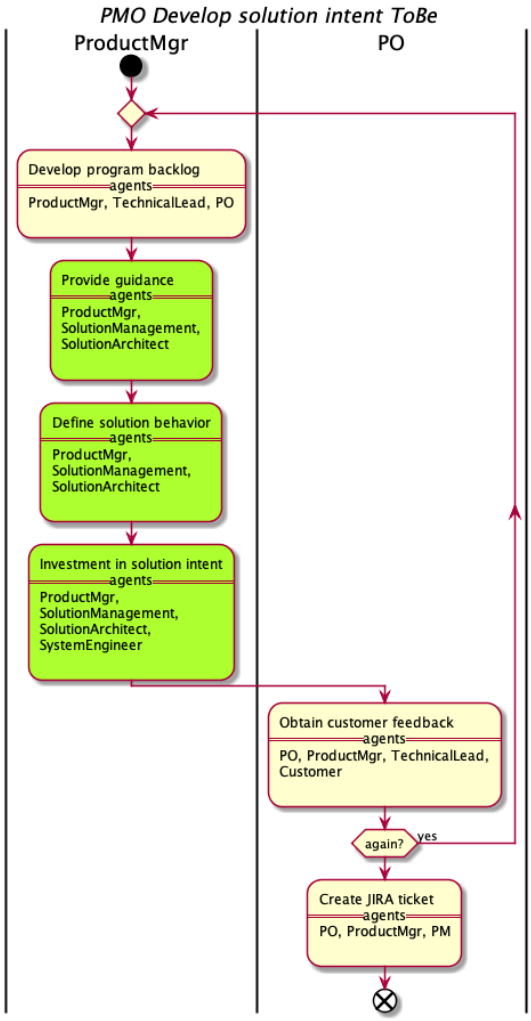

Figure I.113: S4VS70 To-be context.

I.9.10.5 Stakeholders

Product Mgr, Technical Lead, PO, Solution Management, Solution Architect, System Engineer, Customer, and PM.

I.9.10.6 Entry and Exit Criteria

- Entry Criteria: List of development requirements
- Exit Criteria: Product backlog

I.9.10.7 Implementation details

Table I.71: Implementation details for practice S4VS70 – Develop solution intent.

Step	Description	Role(s)
Develop program backlog	Product Manager or PO and Technical lead spends time to dig in more detail into requirements to identify the development needs.	Product Mgr, Technical Lead, PO
Provide guidance	Use vision, critical nonfunctional requirements, roadmap, and milestone to provide sufficient guidance to the teams for initial PI Planning and execution.	Product Mgr, Solution Management, Solution Architect
Define solution behavior	Use features, capabilities, stories, and enablers to define and realize the solution behaviour.	Product Mgr, Solution Management, Solution Architect
Investment in solution intent	If the team is dealing with the regulated environment then in solution intent is required.	Product Mgr, Solution Management, Solution Architect, System Engineer
Obtain customer feedback	Product Manager and Technical Lead send the solution approach to the Customer. If they agree, then the specification is logged into JIRA.	PO, Product Mgr, Technical Lead, Customer

Continued on next page

Table I.71 – *Continued from previous page*

Step	Description	Role(s)
Create JIRA ticket	Product Manager, PO, or Project Manager creates tickets in JIRA for each agreed requirement. Each ticket contains an initial estimate, time, and release plan (in which phase specific ticket will be released).	PO, Product Mgr, PM

I.9.11 Engage in continuous collaboration (S4VS182)

I.9.11.1 Purpose

The purpose of this practice is to ensure that a solution will operate correctly in the deployed environment.

I.9.11.2 Background

It is rare that, systems builders build systems for themselves; they build systems for other people. That means, the system builders do not typically control nor deeply understand the context for deployment and its use. For many reasons, the production environment is not the same as the development environment. To ease this issue, engagement in the continuous collaboration will ensure the solution can be deployed in the customer's context.

There are many collaboration points between the development teams and the various roles within the customer organization. Several SAFe roles carry that responsibility along with their customer counterparts (Leffingwell, 2015). Therefore, effective collaboration between customer and SAFe roles helps ensure that the system meets the customers' needs in their context.

I.9.11.3 As-is Context

I have not observed any as-is for this practice.

I.9.11.4 To-be Context

New actions are shown in green.

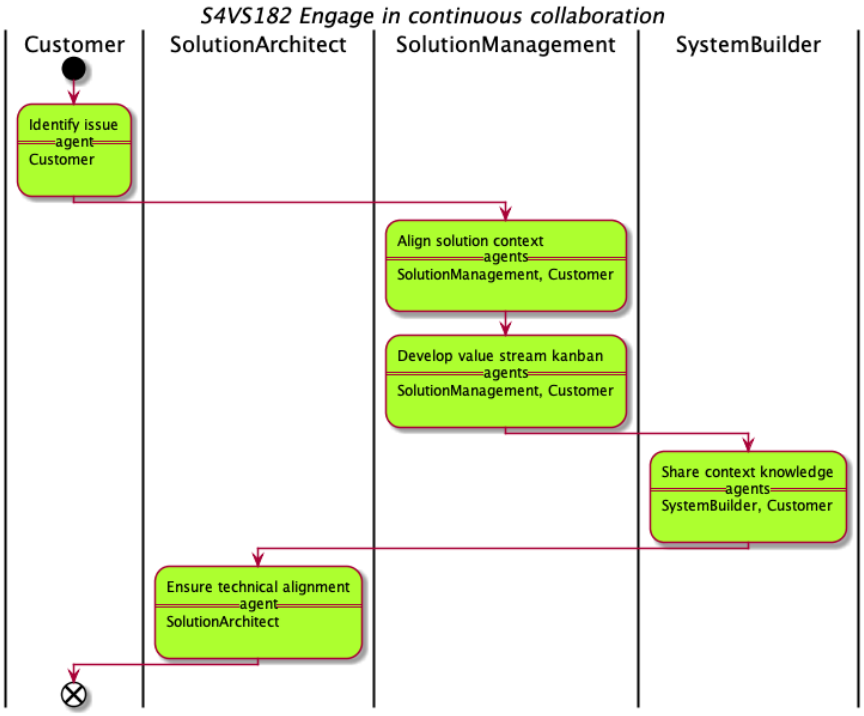

Figure I.114: S4VS182 To-be context.

I.9.11.5 Stakeholders

Customer, Solution Management, System Builder, and Solution Architect.

I.9.11.6 Entry and Exit Criteria

- Entry Criteria: Ceremony, Issue
- Exit Criteria: Technical alignment

I.9.11.7 Implementation details

Table I.72: Implementation details for practice S4VS182 – Engage in continuous collaboration.

Step	Description	Role(s)
Identify issue	The customer raises and discusses context issues during PI planning and solution demos.	Customer
Align solution context	Solution Management and customer continuously ensure that the vision, solution intent, roadmap, and value stream backlog align with the solution context.	Solution Management, Customer
Develop value stream kanban	Issues discovered in the Customer's context run through the value stream kanban system for impact and resolution.	Solution Management, Customer
Share context knowledge	System builders and the Customer share relevant context knowledge, environment, and infrastructure, such as interface mock-ups, test and integration environments, test and deployment scripts, etc.	System Builder, Customer
Ensure technical alignment	Solution Architect ensures technical alignment with solution context-interfaces, constraints, etc.	Solution Architect

I.9.12 Evaluate integrated solution (S4VS207)

I.9.12.1 Purpose

The purpose of this practice is to ensure the business flow throughout the entire solution.

I.9.12.2 Background

Solution development is effective when the stakeholders and team frequently evaluate the integrated increments of the entire solution. SAFe prescribes cadence based PI planning and team integrates at the PI boundary usually via solution demo.

I.9.12.3 As-is Context

I have not observed any as-is model for this practice.

I.9.12.4 To-be Context

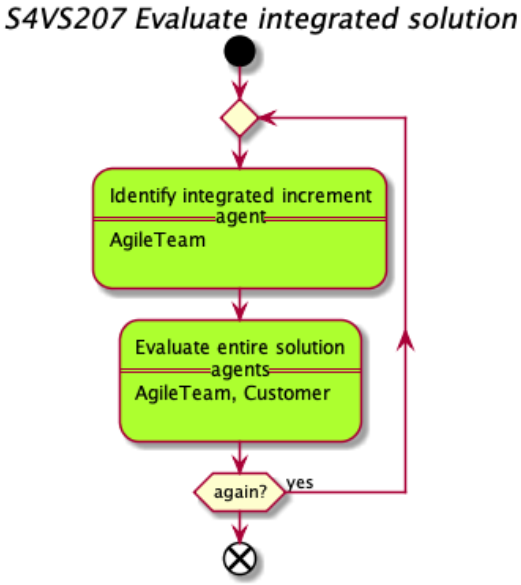

Figure I.115: S4VS207 To-be context.

I.9.12.5 Stakeholders

Agile Team and Customer.

I.9.12.6 Entry and Exit Criteria

- Entry Criteria: Build
- Exit Criteria: Report

I.9.12.7 Implementation details

Table I.73: Implementation details for practice S4VS207 – Evaluate integrated solution.

Step	Description	Role(s)
Identify integrated increment	Agile Team identifies the build that integrated incrementally.	Agile Team

Continued on next page

Table I.73 – *Continued from previous page*

Step	Description	Role(s)
Evaluate entire solution	Stakeholders (customer) and Agile Teams evaluate integrated increment of the entire solution.	Agile Team, Customer

I.9.13 Influence vision (S4PO14)

I.9.13.1 Purpose

The purpose of this practice is to influence the vision and roadmap.

I.9.13.2 Background

Strategic themes have an influence on the vision and backlogs for development at every Portfolio level. Solution and program epics that flow from the Portfolio, or arise locally, are also influenced by the strategic themes. Due to their importance, strategic themes often be presented by the Business Owner during Program Increment (PI) planning. Moreover, strategic themes provide vital conceptual alignment across the trains in a large solution/Value Stream, and across the teams on an Agile Release Train.

I.9.13.3 As-is Context

I have not observed any as-is process for this practice.

I.9.13.4 To-be Context

New actions are shown in green.

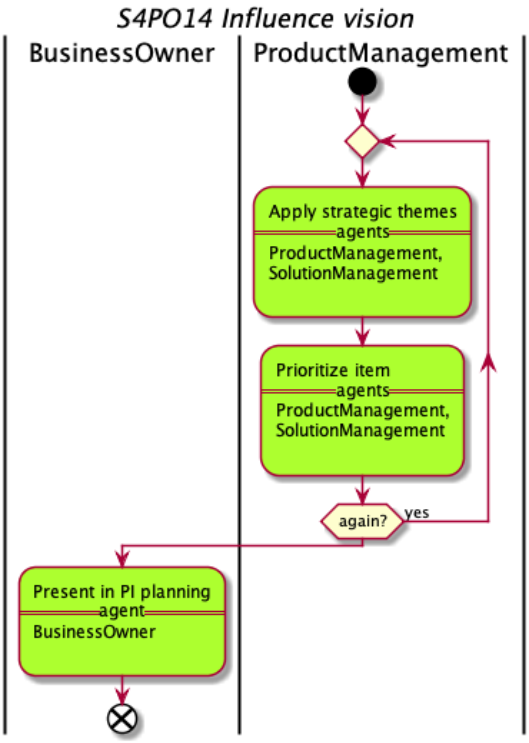

Figure I.116: S4PO14 To-be context.

I.9.13.5 Stakeholders

Product Management, Solution Management, and Business Owner.

I.9.13.6 Entry and Exit Criteria

- Entry Criteria: Strategic Themes, Agreed Portfolio Backlog
- Exit Criteria: Visibility Between Agile Release Trains

I.9.13.7 Implementation details

Table I.74: Implementation details for practice S4PO14 – Influence vision.

Step	Description	Role(s)
Apply strategic themes	Value stream and trains also operate fully within the context of the portfolio vision. So, evolving strategic themes may directly impact them. Product and Solution management applies strategic themes to influence the vision and roadmap.	Product Management, Solution Management
Prioritize item	To drive attributes of weighted shortest job first (WSJF) prioritization for items in the value stream and program backlogs.	Product Management, Solution Management
Present in PI planning	Strategic themes provide an important means of conceptual alignment between the trains. Often presented by Business Owners during PI Planning.	Business Owner

I.9.14 Facilitate technical aspects of collaboration (S4RR44)

I.9.14.1 Purpose

This purpose of this practice is to collaborate with third parties, such as data or service providers.

I.9.14.2 Background

A *System Team* is a special agile team on the Agile Release Train (ART) or Value Stream; provides assistance in building and using agile development infrastructure such as continuous integration (Leffingwell, 2015). The System Team's primary responsibilities are building development infrastructure, solution integration, end-to-end testing, system and solution demos, and release.

I.9.14.3 As-is Context

I have not observed any as-is process for this practice.

I.9.14.4 To-be Context

New actions are shown in green.

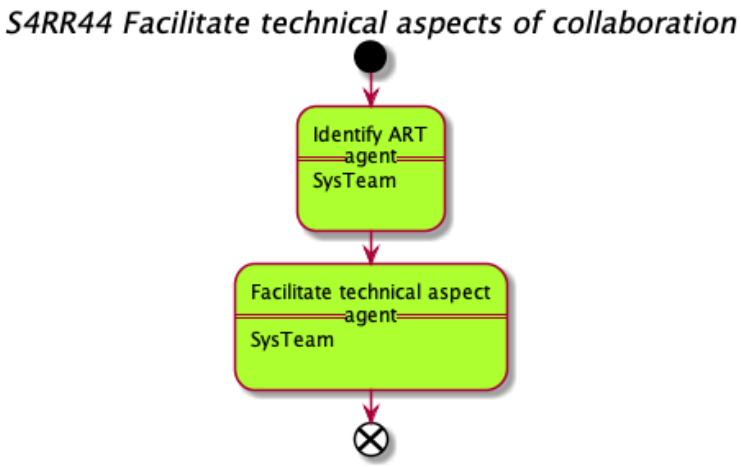

Figure I.117: S4RR44 To-be context.

I.9.14.5 Stakeholders

System Team.

I.9.14.6 Entry and Exit Criteria

- Entry Criteria: Product or Project
- Exit Criteria: Technical support

I.9.14.7 Implementation details

Table I.75: Implementation details for practice S4T128 – Accept stories continuously to improve flow.

Step	Description	Role(s)
Identify ART	System Team identifies an agile release train where they need to collaborate.	Sys Team
Facilitate technical aspect	System Team collaborates with third parties such as data or service providers to facilitate the technical aspects. That helps to build development infrastructure and support to increase ART velocity.	Sys Team

References

Derby, E., Larsen, D. and Schwaber, K. (2006), *Agile retrospectives: Making good teams great*, Pragmatic Bookshelf. 267, 268

Leffingwell, D. (2007), *Scaling software agility: best practices for large enterprises*, Pearson Education. 170, 198, 206, 216, 218, 220, 223, 225, 226, 248, 249, 255, 258, 261, 264, 281, 290, 292, 298, 302, 306, 309, 311, 312, 313, 314, 325, 326, 327, 329, 335, 358

Leffingwell, D. (2015), 'Scaled Agile Framework® 3.0', `http://v3.scaledagileframework.com/`, accessed 2016-04-15. 163, 170, 185, 191, 198, 203, 223, 226, 229, 238, 243, 248, 249, 255, 258, 261, 263, 281, 289, 290, 298, 302, 306, 309, 311, 312, 313, 314, 325, 326, 327, 329, 331, 334, 338, 344, 345, 347, 353, 354, 358, 362, 365, 367, 371, 374, 377, 389, 393, 398, 400, 407, 413

Razzak, M. A., Richardson, I., Noll, J., Canna, C. N. and Beecham, S. (2018), Scaling agile across the global organization: an early stage industrial SAFe self-assessment, *in* 'Proceedings of the 13th Conference on Global Software Engineering', ACM, pp. 121–130. 238