

# Protocol for a Review of Global Software Development Models

Lero Technical Report No.  
2020\_TR\_10\_GSD\_Review\_Protocol

Mohammad Abdur Razzak<sup>1</sup>, John Noll<sup>2</sup>, Ita  
Richardson<sup>3</sup>, Clodagh Nic Canna<sup>1</sup> and Sarah Beecham<sup>3</sup>

<sup>1</sup>Ocuco Limited, Dublin, Ireland  
*email: razzak.abdur@ocuco.com*

<sup>2</sup>University of Hertfordshire, UK

<sup>3</sup>Lero, University of Limerick, Limerick, Ireland

December 28, 2020



## 1 Introduction

We conduct a review of the literature to identify the reported Global Software Development (GSD) process models to include process capability, maturity, stage models, and frameworks. We are particularly interested in establishing the extent to which GSD process models are evaluated and applied in practice. Our study updates research conducted in 2010 [7] to now include GSD models developed between 2010-2019.

This protocol includes our research question, search terms, search strings, inclusion and exclusion criteria, and our preliminary results from our searches. We also include the references of the related work.

## 2 Research Question

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

## 3 Search Terms

**Table 1.** Search Terms.

<i>Category</i>	<i>Keywords</i>
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, Capability model, Maturity model, Framework

## 4 Search String

*( (Global OR gsd OR gse OR dsd) AND ( (model OR framework OR method) AND (capability OR process OR maturity) ) )*

## 5 Electronic Bibliographic Databases

Searches were conducted in the following databases:

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

## 6 Inclusion and Exclusion Criteria

### 6.1 Inclusion

- Publication year: 2010-2019
- Language: English
- Full text available and accessible
- Peer reviewed work
- Experience reports
- Answers our research question
- Empirical studies and theoretical studies will be included.

### 6.2 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

## 7 Results

**Table 2.** Process models in the practice of global software development<sup>1</sup>.

Model	Scope/Focus	Type	Levels	Practices	Empirically evaluated?
SITTO model [3]	Business/Organization	Capability	4 Levels	Not defined	Yes
Outsourcing Maturity Model [1]	Business/Organization	Capability	5 Levels	Not defined	No
Process Maturity Framework (PMF) [8]	Technical/Project	Maturity	3 Levels	24 Practices in 4 areas	Yes
Collaboration Maturity Model [5]	Business/Project	Maturity	15 Level	Not defined	No
SQM-CODE model [10]	(Business & Technical/Organization)	Capability	5 Levels	84 Practices in 4 areas	No
Meta-Capability model [2]	Business/Organization	Capability	Not defined	Not defined	No
Evolutionary Framework [6]	Business/Organization	Capability	3 Levels	Not defined	Yes
TAPPER Framework [4]	Business/Organization	Capability	5 Levels	Not defined	No
<b>Additional Models published after 2010</b>					
GTM [9]	Organization & Project	Capability	4 Levels	20 Sub-practices and 70 recommendations	Yes
ADAPT [11]	Project	Not defined	Not defined	10 guidelines and 29 practices	Yes

<sup>1</sup> Adapted and augmented from [7]

## References

1. Adalakun, O.: It outsourcing maturity model. ECIS 2004 Proceedings p. 25 (2003)
2. Balaji, S., Ranganathan, C.: Exploring the key capabilities for offshore is sourcing. ICIS 2006 Proceedings p. 37 (2006)
3. Carmel, E., Agarwal, R.: The maturation of offshore sourcing of information technology work. In: Information systems outsourcing, pp. 631–650. Springer (2006)
4. Hofner, G., Mani, V.: Taper: A generic framework for establishing an offshore development center. In: International Conference on Global Software Engineering (ICGSE 2007). pp. 162–172. IEEE (2007)
5. Lasser, S., Heiss, M.: Collaboration maturity and the offshoring cost barrier: the tradeoff between flexibility in team composition and cross-site communication effort in geographically distributed development projects. In: IPCC 2005. Proceedings. International Professional Communication Conference, 2005. pp. 718–728. IEEE (2005)
6. Mirani, R.: Client-vendor relationships in offshore applications development: An evolutionary framework. Information Resources Management Journal (IRMJ) **19**(4), 72–86 (2006)
7. Prikladnicki, R., Audy, J.L.N.: Process models in the practice of distributed software development: A systematic review of the literature. Information and Software Technology **52**(8), 779–791 (2010)
8. Ramasubbu, N., Krishnan, M.S., Kompalli, P.: Leveraging global resources: A process maturity framework for managing distributed development. IEEE software **22**(3), 80–86 (2005)
9. Richardson, I., Casey, V., McCaffery, F., Burton, J., Beecham, S.: A process framework for global software engineering teams. Information and Software Technology **54**(11), 1175–1191 (2012)
10. Siakas, K.V., Balstrup, B.: Software outsourcing quality achieved by global virtual collaboration. Software Process: Improvement and Practice **11**(3), 319–328 (2006)
11. Vallon, R., Strobl, S., Bernhart, M., Prikladnicki, R., Grechenig, T.: Adapt: A framework for agile distributed software development. IEEE Software **33**(6), 106–111 (2016)