Software Development Processes in Globally Distributed Environment

Zane Galviņa[1]**, Darja Šmite**[1, 2]
[1] University of Latvia, Latvia, [2] Blekinge Institute of Technology, Sweden
zane.galvina@lu.lv, darja.smite@{lu.lv | *bth.se}*

As a result of globalization, software is nowadays more often produced by development effort from multiple locations. While global software development is regarded as more challenging than even the most complex project managed entirely in-house, standards or methodologies dedicated for this type of projects are still lacking. Based on an extensive literature review towards an understanding of industrial practice regarding software development processes, authors of this paper conclude that the evidence on how such projects are organized is scarce. Despite the limited evidence, authors present the deduced models of development processes presented in selected literature and summarize the main challenges that may affect project management processes.

Keywords: Software Engineering, Distributed development, Software Development Process, Global Software development.

1 Introduction

Due to continuous increase in competition in the field of Information Technologies, companies are forced to provide products and services, which coincide with efficient and effective development, and high quality standards that are economically viable. Agile development is a movement that has entered software engineering to bring faster and cheaper development through lightweight process thinking and entrusting work to skilled people rather than enforcing heavy documented standardized process models. Another way to achieve competitiveness is to start distributing development globally in order to gain a benefit from getting more and cheaper resources. In this paper, the focus is brought on the latter. Unfortunately, many of the assumed benefits of global software development are associated with significant challenges [1] that hinder smooth project performance. Thus, a better understanding of processes undergoing in software projects in distributed environment may help to find the necessary improvements and reach the promised benefits.

A closer look at the main difficulties reveals that geographical distance inherent in distributed environment, in which software project team members are separated in space and time, has significant impacts on communication, coordination and other processes

[2, 3]. Many reams have been written to describe these challenges while solutions in this area are still not well-represented [1]. While traditional software engineering is managed with the help of various well-known life cycle models, development and management methodologies and standards, the area of distributed work is relatively unexplored and there is still no standard approach to run distributed projects.

The underlying assumption of software process research that stresses the importance of this area is the direct correlation between the quality of the process and the quality of the developed software [4]. Since the invention of the waterfall lifecycle [5], many process models have been introduced varying from disciplined well-defined processes to undisciplined and ill-defined processes, and from heavy and slow to light and agile processes [6]. Software lifecycle, in fact, is a skeleton and a philosophy, which defines, how software processes will be carried out, and specifies such characteristics as tools, infrastructure, environment, methods and techniques, organizations and people etc. [4]. Along with the distribution of software projects, the role of a clear understanding of software lifecycle and processes emerges. Because distributed software projects are regarded as more complex than even the most complicated projects performed entirely in-house [7], we conjure that it is also important to explore whether the very notion of the lifecycle does not change with distribution. In other words, the dualistic (or multiplistic) nature of distributed software projects could also affect the way one should view, specify, and perhaps execute distributed software project lifecycles.

Motivated by this gap, authors of this paper aim to explore different lifecycles and work division approaches and their relation to existing process models. Thus, the following research questions are put forward:

RQ1: How are distributed projects organized in terms of process patterns?
RQ2: Are there successful or unsuccessful process patterns?

The rest of the paper is structured as follows. Section 2 presents the overview of the research process. Section 3 presents the results of the literature review – presented development process models and challenges affecting project management as well as process specifics. Finally, Section 4 concludes the paper.

2 Research Overview

In order to answer the research questions mentioned above, an extensive selected literature review was conducted. Research papers were collected from the following venues being regarded as the key publication sources in the area of global software engineering:

- Proceedings of the Inter. Conf. on Global Software Engineering 2006-2009;
- Communications of ACM special issue on GSE – 49(10)/2006;
- SPIP special issues on GSE – 8(4)/2003, 13(3)/2008, 13(6)/2008, 14(5)/2009;
- IST special issue on GSE – 49(9)/2006;
- IEEE Software special issues on GSE – 18(2)/2001, 23(5)/2006.

In total, 261 papers were collected and reviewed. Since global development is quite a broad concept, several criteria were taken into account in order to screen the papers and select the relevant ones. The screening was performed by the first author.

- **Criterion 1.** The paper reports information about distributed software development project. Thus, if the title clearly states that the article is about a comparison of different tools used in distributed development, the paper was rejected. In case of very general titles abstracts were read. Most of the authors present detailed information about the paper in the abstract, so to understand whether this criterion was satisfied, a review of introduction and/or conclusion was necessary only in few situations.
- **Criterion 2.** The paper presents more specific information about the work division methods or process location (detailed description of a process) in a project. Since this criterion is more specific, we could identify it only after reviewing introductions and conclusions.

After reviewing the 261 initially collected papers, only 8 papers were regarded as relevant for further analysis. Since the amount of selected papers was very low, which threatened the validity of conclusions, we continued searching through the reference list of the selected 8 papers. The gathered references resulted in the identification of 97 additional hints. Further collection was based on 4 criteria:

- **Criterion 3.** Availability of reference (6 papers were not found).
- **Criterion 4.** Reference written in English (5 papers failed).
- **Criterion 5.** Conference proceedings or journal paper (40 references failed).
- **Criterion 6.** Unique paper (8 references were duplicates).

Finally, 34 papers were included for a more thorough review of their relevance for the topic under investigation, but they failed after the evaluation using criterion 1 and criterion 2. After applying the next four criteria, only 4 papers were considered for further analysis. A total of 12 papers were carefully read, and the information about the process distribution, locations, work division and associated challenges was mapped for further analysis. The information was captured in a protocol and then broken down to the following topics of discussion (see Fig. 2.1).

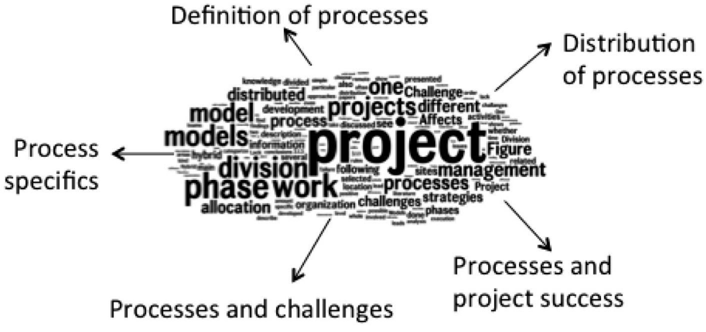

Fig. 2.1 Emerging observations

3 Research findings

Our observations show that the most commonly discussed issues are related to general project information or description of methodology and processes, which are followed. Authors often describe the issues faced in the studied projects and present guidelines how to overcome these challenges. In the majority of the cases, these challenges are related to the whole project, and not to a specific project lifecycle or work division approach. This was also one of the reasons for excluding many studies from the final analysis. Nonetheless a few useful observations were made and are discussed below.

3.1 Definition of processes

While every organization adopts a suitable lifecycle or process model, the notion of following one lifecycle or process model in distributed projects is challenged by collaboration among several organizations often having their own processes, methodologies, and tools. Lack of standards in the activities between distributed teams is regarded as a problematic setting [8]. A lack of coherent development or execution processes is also reported in several other studies [9, 10, 11]. According to Sudershana et al., lack of clear process for project execution leads to increased level of frustration and decreased feeling of ownership, which ultimately results in a very poor acceptance level at the remote location [9]. This finding indicates that a process definition including work division and allocation strategies that clarify roles and responsibilities are essential.

3.2 Distribution of processes

It is difficult to decide whether a particular project should be developed by globally dispersed teams – and where it can be better developed, as well as how to divide it across sites [8]. Among the accepted papers there are several describing a certain distribution of phases and also phase allocation strategies. At the same time, several projects include a detailed description of each phase.

The available information was sufficient only to study different variations of phase allocation strategies. In particular, we studied Requirements engineering, Design, Coding and Testing phases and their allocation to different organizations or sites. One of our goals was however to find out what kind of work division approaches were adopted in each project, including different activities and roles each organization played. This is particularly important when a phase is shared by two organizations, since this can be done in a variety of ways. We failed at this point because the presented details about the projects were scarce. Therefore, only conclusions from studying different phase allocation strategies are discussed.

For simplicity reasons, we demonstrate different approaches by depicting phases in a sequential manner distributed between only two sites. This has been done to categorize all the selected models. Three possible alternative divisions would have been available – 1) strict phase separation, 2) joint execution, and 3) hybrid approach (see Fig. 3.2.1)

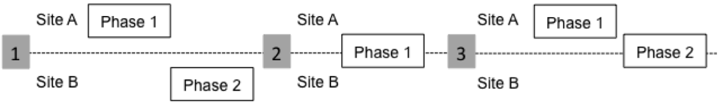

Fig. 3.2.1 Work division approaches.

In the answer to the **RQ1**, we conclude that hybrid division of work [9, 12, 13, 14] is more common than a strict allocation of phases or joint execution of the whole project [10] (see Fig.3.2.2). Possible reasons for the dominance of hybrid models are twofold. We assume that joint execution is an option for site A to mitigate the risks through control and partial participation in the work of site B, and vice versa – involvement of the site B in the activities executed by the site A is an option to mitigate the challenges of information sharing and knowledge transfer in later phases. At the same time, certain key or critical phases, such as requirements engineering, are executed independently due to, e.g., proximity to the customers.

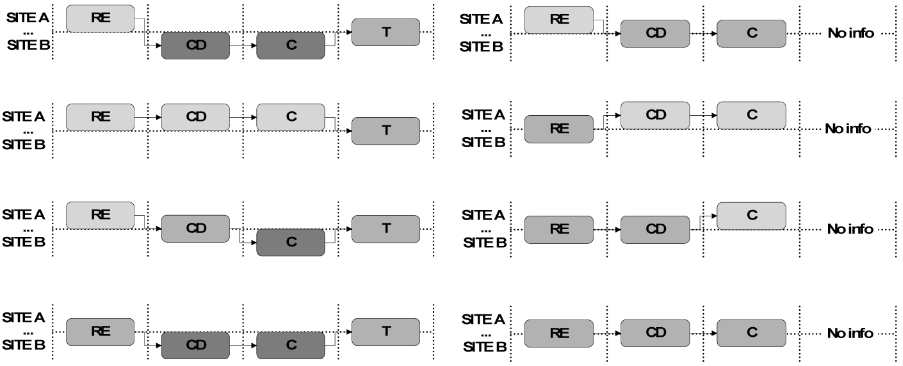

Fig. 3.2.2 Work division models [9,10*,12, 13, 14].
*presents many division models.

While reading project descriptions, several common uncertainties were identified. First of all, one may wonder, whether a phase is performed jointly, are all activities performed jointly too, or do organizations split the work? Secondly, if employees from one site are moved to another site for the entire phase, does that mean that the phase is allocated at one site, or performed jointly? From these questions we conclude that the phase visualization has its limitations and the process description shall allow reflecting answers to all these questions.

3.3 Processes and project success

Unfortunately, the success or failure of the work activities or work division approaches was not explicitly discussed. From the studied research, it was not possible to determine the factors, which have had an influence (positive or negative) on projects. This is yet another case were we have stumbled upon a lack of description of the project lifecycle models, work distribution or even phase allocation, which significantly affected the ability to drive any meaningful conclusions regarding successful or unsuccessful process patterns (**RQ2**).

Despite the different approaches to work division, each of the studied hybrid projects faced many challenges. It shows that there are no universal development model for all projects and organizations. Each organization has to choose the most appropriate model for their situation.

3.4 Processes and challenges

One of the objectives was to understand whether the selected work division model in a project makes an impact on a project management. Project management is the application of knowledge, skills, tools and techniques to project activities in order to meet project requirements [15]. The following key challenges found in the literature can be linked with one or several knowledge areas of project management.

Joint execution with no standards for the interface between the sites.

Usually, every organization has their own standards, defined processes, which they follow, but no standards determine how to join them or how to work together [8]. This has mainly an effect on the project integration management.

Joint execution with limited synchronous interaction.

This is a challenge that is manifested in projects involving sites separated in time. It is more complicated to share information between team members that lack synchronous communication or have limited overlap in working hours [8]. In such cases, tools for sharing information commonly used in software projects are of little help. Hampered communication and coordination delay are also consequences of asynchronous interaction [11]. This affects project coordination, scope and time management as well as communication management significantly.

Joint execution requires more time than co-located work.

Tasks in global software development projects often take much longer than in co-located environments [11]. In order to estimate activity duration or develop a common schedule for project, it is not enough to rely on experience from co-located projects. This affects the project time management as well as the scope management.

3.5 Specifics of processes

From our observations we deduce that the decisions of process model are closely affected by the work allocation to different sites and the chosen level of sharing the processes, phases or activities. In the following figure, we offer our understanding of process-specific factors of concern, based on a class diagram of a software process model defined in [16].

Accordingly, we emphasize that each phase can be allocated to an organization (or its site). However, for the clarity of roles and responsibilities it is also important to specify which activities are going to be performed by each location or joint.

A study depicting the proposed work division details would be sufficient to categorize the projects according to the following four main work division approaches [17]:

- *Phase-based approach* – This is a division of work by phase/process step, when globally dispersed sites engage in different phases of a project in a sequential manner [17].
- *Model-based approach* – Division of work by product structure (product module), when each product module/feature is developed in a single site [17].
- *Distributed approach* – Division of work that minimizes requirements for cross-site communication and synchronization; however, only for particular types of product architectures [17].
- *Customization-based approach* – This is a division of work based on product customization, so that one site develops the product and other sites perform

customization, that is, changes such as adding features and enhancements for specific customers [17].

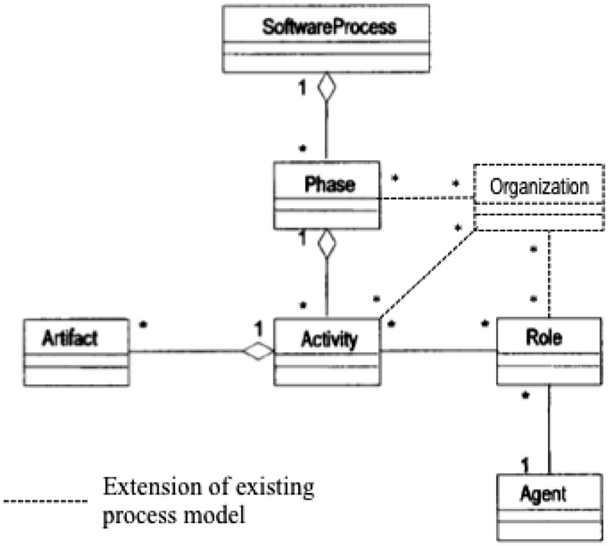

Fig. 3.5.1 Process model (extension of [16])

We consider it to be important that the above-discussed details must be specified by authors of empirical studies of distributed projects who report the success or failure of certain projects, processes, approaches, or practices, in order to further trace the main reasons to the process organization. This could further facilitate determination of successful or unsuccessful process patterns.

4 Conclusion

The efforts directed towards capturing different approaches used in practice and reported in academic literature showed that research on description of process models and work division practices in globally distributed projects is scarce. More often, the studies present the main challenges but do not mention the necessary information about the project context, which confirms the findings from a systematic review in this area [1]. Our findings also show that it is not possible to strictly define or make any conclusions about the best strategy for process allocation and work division. Despite that, we were able to identify several key challenges related to project management that arise from joint execution of task. This paper also presents an extension of an existing process model and emphasizes the necessity for accurate description of distributed projects.

By evaluating the ratio between the amount of selected papers for further analysis and the rest of the papers initially collected from the venues that represent research in the studied field, we can also conclude that there is a need for additional studies in this area, i.e. the understanding of software processes in distributed projects. To foster the progress, we present specifics of processes necessary to be identified in future research. We also encourage reporting complete and comprehensive information about the studied projects.

References

1. Conchúir, E. Ó., Holmström, H., Ågerfalk, P.J., & Fitzgerald, B. "Exploring the assumed benefits of global software development". In P. Fernandes et al. (Ed.), IEEE International Conference on Global Software Engineering. Los Alamitos, CA: IEEE Computer Society, 2006, pp.159-168.
2. Damian D, Zowgui D. "The Impact of stakeholders' geographical distribution on managing requirements in a multi-site organization". Proceedings of International Conference on Requirements Engineering, Monterey,CA, 2002.
3. D. Šmite, C. Wohlin, R. Feldt, T. Gorschek "Empirical Evidence in Global Software Engineering: A Systematic Review", In: Journal of Empirical Software Engineering, Vol. 15, Nr. 1, February 2010, pp. 91-118.
4. Fuggetta A. "Software process: a roadmap". In Proceedings of the Conference on The Future of Software Engineering (ICSE '00). ACM, New York, NY, USA, 2000, pp. 25-34.
5. Royce W. W., "Managing the development of large software systems: Concepts and techniques," Proc. WESCO, 1970.
6. Rodriguez-Martinez L.C., Mora M., Alvarez F.J, "A Descriptive/Comparative Study of the Evolution of Process Models of Software Development Life Cycles (PM-SDLCs)," Mexican International Conference on Computer Science, 2009, pp. 298-303.
7. Karolak, D. **"**Global Software Development: Managing Virtual Teams and Environments". Wiley-IEEE Computer Society Pr; 1st edition, December 27, 1998.
8. Prikladnicki R.,Audy J., Evaristo R. "Global Software Development in Practice Lessons Learned". Software Process: Improvement and Practice.2003, pp. 267-281.
9. Sudershana S., Villca-Roque A., & Baldanza J. "Successful Collaborative Software Projects for Medical Devices in an FDA Regulated Environment: Myth or Reality". Proceedings of International Conference of Global Software Engineering, 2007, pp. 217-224.
10. Berenbach, B., "Impact of Organizational Structure on Distributed Requirements Engineering Processes: Lessons Learned", Workshop on Global Software Development for the Practitioners at ICSE, Shanghai, 2006, pp. 15-19.
11. Peter Faßbinder, Volker Henz, "Improving Global System Development and Collaboration across Functions: Experiences from Industry," Proceedings of International Conference of Global Software Engineering, 2009, pp.262-266.
12. Caprihan G. "Managing Software Performance in the Globally Distributed Software Development Paradigm". Proceedings of International Conference of Global Software Engineering, 2006, pp. 83-91.
13. Burger W. "Offshoring and Outsourcing to INDIA". Proceedings of International Conference of Global Software Engineering, 2007, pp. 173-176.
14. Cusick J., & Prasad A. "A practical Management and Engineering Approach to Offshore Collaboration". IEEE Software, 2009.
15. A Guide to the Project Management Body of Knowledge (PMBOK Guide). American National Standard, 2004.
16. Karolak, D. "Global Software Development: Managing Virtual Teams and Environments". Wiley-IEEE Computer Society Pr; 1st edition, December 27, 1998.
17. Oshri I., Korlarsky J., & Willcocks L.P. "The Handbook of Global Outsourcing and Offshoring". Palgrave Macmillan 2009.