On the Identification of Factors that Promote High-Performance Projects in Distributed Development

Preliminary Findings of an Empirical Study of a Fortune 500 IT Multinational Company

Sabrina Marczak[1], Marcelo Perin[1], Rafael Prikladnicki[1], Christiano Ayub[2]

[1]Computer Science School– Pontifícia Universidade Católica do RS (PUCRS)
6681 Ipiranga Ave. – Partenon – 90.619-900 – Porto Alegre – RS – Brazil

[2]Development Center, ORG
Ipiranga Ave. – Partenon – 90.619-900 – Porto Alegre – RS – Brazil

{sabrina.marczak,rafaelp,mperin}@pucrs.br, christiano.ayub@gmail.com

Abstract. *As part of a long-term research on high-performance projects in distributed software development, we sought to investigate what leads a project to meet or exceed its expected performance. In this paper we report on the preliminary findings of our qualitative exploratory study. We conducted 11 semi-structured interviews in a Fortune 500 IT multinational company that develops software in-house to support its business processes. Participants listed 7 factors that promote high-performance in their opinion, including timely attending the organization's business needs. They also mentioned 5 issues related to achieving performance such as having a person mediating the conversation between the business and the IT departments. We present the identified factors and issues, and discuss their implications to the performance of distributed software development projects.*

1. Introduction

Project management discipline aims to guide software teams to plan, implement, and control the development of any software product. Organizations introduce project management to their software projects aiming to deliver their projects on time, on budget, and with quality (Project Management Institute, 2013). However, in today's globalized IT market organizations also have to timely attend their customer's demands in order to remain competitive. Therefore, organizations and managers desire to have their projects attending or exceeding their expected performance goals. We name these *high-performance projects*.

Although there are several studies on project performance in distributed software development (e.g., (Herbsleb and Mockus, 2003)), little is still known about what promotes high-performance in this kind of project in modern times. In the past decade we have seen information and communication technology improving, agile methods being proposed, companies going over major reorganizations to better attend their customer expectations, among other changes that might have affected how software teams perform and deliver software products. Therefore, as part of a long-term research on high-performance projects, in this initial phase we sought to empirically explore what contributes to a distributed software development project to meet or exceed its expected performance. We conducted an exploratory qualitative study based on semi-

structured interviews in a large IT company, named ORG (a fictitious name due to confidentiality restrictions). Our 11 study participants listed 7 factors that promote and 5 issues that make it difficult to achieve high-performance in distributed software projects. In this paper we report on the findings of this initial study and discuss the implications of the identified factors and issues to the performance of distributed software development projects.

2. Research Method

Our empirical study consisted of interviews conducted on-site in a large IT company. Interviews were transcribed and analysis was guided by ground theory procedures (Corbin and Strauss, 2007).

2.1. Company Background

The study was conducted in a large IT multinational company. Software products to support the organizational processes are developed by the IT department. Demands to develop or to update these products come from the business departments. IT development teams are distributed among the headquarters' office located in the US and in Brazil, India, and Malaysia. The IT department follows a matrix structure based on business areas (e.g., sales) and IT functions (e.g., developers). Projects vary from the development of new products to the maintenance of legacy systems. Project teams mainly follow the waterfall model. Some Scrum practices are scarcely adopted to support project management. Processes vary from formal (following CMMI Level 3 practices) to informal (defined by the project members upon their needs).

An annual project roadmap is defined in December based on the requests made by business representatives and recorded by business analysts. Business analyst managers in conjunction with project managers prioritize the requests and define a set of projects to be developed throughout the year. Priorities are defined based on business impact and on development costs. Distributed software teams are then formed to develop the elected projects. Members are assigned to the projects based on their skills and domain knowledge, despite of their physical location. Therefore, a project often has its roles distributed over several locations. By mid February each team receives a business request document. The software team starts working to translate the business into software requirements led by the software requirements analysts. These have to consult with business analysts to clarify business requirements and, when necessary, business representatives are invited to join the discussion. Project managers monitor the project progress based on a set of organizational performance measures that are reported to senior management in a regular basis. Results from these measurements are used to determine whether a project failed, attended, or exceeded its performance goals.

2.2. Data Collection and Analysis

Our study consisted of 11 interviews conducted on-site with Brazilian team members. Each interview lasted for an average of one hour. We asked the participant to answer to the following taking her working experience with the company into consideration: *"Looking back at the distributed software projects you have participated on, please think of one project that stood out and elaborate on what you think that contributed to this project to attend or exceed its performance goals."*

Study participants were selected based on their experience working with the company and on their role within the IT department. We started by asking the Brazilian IT Director whom we should be talking with. He pointed out 3 Senior Managers who have started in the IT department about 12 years ago when the development center was created. We then asked these 3 Senior Managers to point out more prospective participants. Eight other people were interviewed, totaling 11 participants in our study. We received suggestions of other prospective participants, but as we analyzed the collected data as we were conducting the interviews, we considered this number sufficient by saturation of the responses. Table 1 describes the participants' current roles and job descriptions, and their past roles within the company.

All interviews were transcribed, and transcriptions were prepared for analysis in the ATLAS.ti software. Our subsequent analysis was guided by grounded theory procedures (Corbin and Strauss, 2007). One researcher coded the data using the open-coding procedure. A second researcher coded a smaller sample of the transcriptions to compare the identified codes. The researchers then discussed the code list together, unifying codes into categories when appropriate. The resulting categories represent the factors and the issues presented in Section 3. Both researchers conducted a 3 hours long meeting with the 11 participants to report on the findings, reviewing the results to ensure accurateness and discussing their usefulness to the company as suggested by Creswell (2008).

3. Initial Findings

Participants reported on 7 key factors. We present them below. Note that each participant has reported on more than a factor. Participants' quotes include the study's participant identifier (ID) defined in Table 1.

- **Factor 1:** Enables the business and helps the business to evolve. Four participants mentioned that it is important that the project *"introduces something new to the business that will help it to quickly evolve and better attend the market"* (P6). For instance, one of the senior development managers (P4) discussed *"why would the organization migrate legacy systems that are reliable and have a low cost maintenance to new technologies if the systems would maintain their original features"?* He defends the idea that new functionalities that would help the business to speed up its activities have to be introduced in order to justify such major maintenance migration. The senior manager that was firstly hired in Brazil (P2) recalled that a few projects considered as high-performance were those that had helped the business to answer the question such as *"what can help us improve the way how the [organizational] business is done nowadays?"*
- **Factor 2:** *Delivers what the business needs in a timely manner.* Three participants highlighted the importance of delivering what is requested in a timely manner "*in order to have a software product that supports the business process that is in place*" (P7). The former software architect (P9) argued it is important to "*try to anticipate the estimated deliver date since the faster the new system is in place the more likely it is that it will be in sync with the current business process*". One of the senior development managers (P5) mentioned "*there is no room for delays in this company, if the project is late the process might not be there anymore, we change things to frequently*".

Table 1. Participant's background

ID	Current Role and Job Description	Past Roles
P1	Quality and metrics manager: This is a worldwide position responsible for defining and collecting metrics to measure performance of IT teams as well as defining processes to guide their work.	He started 12 years ago as a Software Reqs Analyst. In this position, he led the worldwide initiative of defining processes to support requirements engineering activities. He also acted as a Project Manager when he proposed, among other things, a subset of the current performance measures.
P2	Career manager: This manager is responsible for planning and help developing the career of the Brazilian developers (it includes developers and team leaders)	He also started 12 years ago. He was the first employee hired by ORG to work in the Brazilian Development Center. He acted as a Development leader and a Project Manager. He is also one of the focal points for the research activities with PUCRS.
P3	Manager of the Project Management function: This is a worldwide position responsible for hiring Project Managers, allocating them to manage IT projects, and following-up on their work.	He is originally from the USA and joined ORG about 20 years ago. He was one of ORG's first salesmen. He joined the IT team 12 years ago as a Project Manager. He was one of the first managers assigned to work with a Brazilian team. Two years later he asked to have this job position transferred to Brazil. He also worked as a Senior Project Manager and a Career Manager.
P4	Senior Development Manager: This position is responsible for assigning developers to projects, and to following-up on their work alongside the Project Manager.	He has been working at ORG for about 10 years and has started as a Developer. He also acted as a Project Manager.
P5	Senior Development Manager: Same job description as P4.	He has also started 10 years ago as a Developer. He has acted as Dev Leader.
P6	Senior Development Manager: Same job description as P4.	He has also started 10 years ago as a Developer, acted as Dev Leader and Software Architect.
P7	Project Manager: This position is responsible for managing the development of the IT projects and for collecting performance measurements to share with Senior Management.	She started about 8 years ago as a Software Requirements Analyst.
P8	Project Manager: Same job description as P7.	She started 10 years ago as a Tester and also acted as Test Leader.
P9	Project Manager: Same job description as P7.	He also started 10 years. He joined ORG as a Developer and acted as a Software Architect.
P10	Senior Development Manager: Same job description as P4.	He started 6 years ago as a Project Manager and acted as the Site Manager for 3 years. As a Site Manager he was responsible for hiring people and controlling the overall IT operations in Brazil.
P11	Senior Development Manager: Same job description as P4.	He joined ORG about 4 years ago as a Project Manager. He managed the development of projects that are critical for the company's operation.

- **Factor 3:** *Has an alignment between the business needs and the software requirements*. Seven out of 11 participants discussed how dynamic the organization is and how challenging it is to have the software products attending the business needs. For instance, the company has gone over two major reorganization structures in the last 18 months, changing job functions, business goals, among other changes. Therefore, they highlighted that a key factor to achieve high-performance is *"to be flexible and fast in perceiving changes and adjusting to them in order to keep the software aligned to business needs"* (P1).
- **Factor 4**: *Finds the balance between what the customer 'wants' and what the customer 'really needs'*. Three participants cited how critical it is that the IT team is able "*to distinguish between what the customer wants and what is really necessary to support the business processes*" (P3). In a distributed IT organization, where the customers and end users are located in different time zones, often over 2 or 3 continents, it is "*challenging to get to know the business processes in details to be able to question the customer about her requests*" (P2). The Quality and Metrics manager (P1) recognized the importance of a "*highly-skilled software requirements analyst*" to mediate the discussion with the business department in such scenario.
- **Factor 5:** *Has a requirements engineering process that efficiently and effectively defines what has to be done.* Four participants recounted the relevance of a clear, well-defined, and disseminated requirements engineering process. Former tester (P8) recalled when the entire IT organization followed a single and unified development process based on the CMMI Level 3. She mentioned "*it was easier and faster to communicate with everyone, and to perform our tasks. Now we need to waste time negotiating with people from other sites how to do things and we never always reach an agreement at first*".
- **Factor 6:** *Has an adequate and qualified team.* Five participants mentioned the impact of "*the right team*" (P10). They were unisonous in recognizing that the organization "*has skilled employees that know how to provide the expected technical solutions*" (P2). They referred to the fact of having "*people who know how to approach the business team*" (P2), "*how to establish critical connections with them*" (P10), and "*to identify whether the proper stakeholders are the ones originating the demands to the IT team*" (P11).
- **Factor 7:** *Delivers on time, on budget, and with quality.* Although 6 of the participants mentioned that attending "*traditional project management targets (e.g., being on time) is expected from any manager*" (P2), only one participant described as a key element "*to deliver on time, on budget, and with quality"* (P5). He said "*we cannot forget these are basic goals to any project and that sometimes we go easy on them jeopardizing the quality of the final product*".

Although it was not part of our initial goal, during the interviews participants also cited issues that had led projects to fail or to not completely attend their performance goals. Given how much emphasis the respondents put on these issues, we report them below as follows:

- **Issue 1:** *To have a mediator between the business department and the IT team.* Four participants recalled the difficulties faced by software teams to timely clarify the requirements. The former software requirements analyst (P7) mentioned "*we know that this is the structure that the organization imposes on us but they need to realize*

how much it delays reaching out those who really know how the process should be run". The f critical applications' former manager (P11) added "*We need to be able to reach the business representatives without depending on the business analysts that quite frequently work in a high level that leaves important details out for the design of the software solution*".

- **Issue 2:** *To validate requirements too late in the development lifecycle.* Four participants pointed out as a serious drawback the fact that software requirements are validated only after the development is over. One of the project managers (P7) reported "*too many business rules are identified as missing or misunderstood by the software team in this point*". "*It is also here that we learn that business processes have changed and that the software has not been deployed yet is already obsolete*" (P9). One of the senior development managers (P11) suggested that "*traditional validation techniques like prototyping could be adopted to avoid such late findings*".
- **Issue 3:** *To have poorly written software requirements.* Although the participants mentioned that they perceive the software professionals as well skilled, 3 of them indicated that software requirements are still poorly written. A senior manager (P2) said "*because the projects' roadmap is defined based on non-standardized written business needs and requests, sometimes we inherit poorly defined business requirements that are translated into poorly written software requirements. It would be best if we could be closer to the business representatives*".
- **Issue 4:** *To work based on assumptions.* Associated to the perception that the distance between business and IT teams is prejudicial, 3 participants mentioned that "*assuming certain knowledge about the business processes is a common practice of the software teams that often results in disaster situations*" (P11). A senior development manager recalled that "*the organization is so dynamic that even within a single project it is risky to assume that the [business] processes have not changed, mind in between projects*" (P6). They wish the team members *"would double-check more often business rules and other important definitions used in the specification of the software requirements"* (P9).
- **Issue 5:** *To implement improvements that were not requested.* Three participants commented on the fact that "*software teams often add small improvements to the applications without discussing them with the business analyst, resulting sometimes in a positive feedback from the customers but quite often in waste of time and rework*" (P7). "*This initiative is seen as pro-active behavior by software members but perceived as 'noisy' by the business team*" said a manager (P1).

4. Discussion and Final Considerations

To deliver a project on time, on budget, and with quality is a key premise in software organizations. However, with the constant changing IT market software solutions need to quickly adjust to business changes and to new customer' requests. Agile methods have been introduced aiming to provide a more flexible approach to software teams (Larman, 2003). Although these methods define practices that promote a more pro-active way of working, there are organizations that still work based on more traditional approaches such as the waterfall model as the investigated organization. Participants claimed that following a more structured development model and a more 'traditional' organization structure in which communication channels between departments are centralized helps

assuring that the project goals will be better achieved and that the customer will be better served. Almost two decades ago Al-Rawas and Easterbrook (1996) have reported that organizational barriers can inhibit efficient communication that leads to requirements issues found later on in the development process. Despite this knowledge, our findings suggest that the issue is still around.

Assuming that this Fortune 500 large IT multinational is not the only company in the world to still follow the waterfall model and to having strictly communication channels defined via the organization structure, our findings contribute to bringing to light that traditional software engineering issues (e.g., issue 3 - poorly written software requirements (Damian and Zowghi, 2003)) are still being faced by distributed software teams. When going agile and following a more loose and dynamic process are not a feasible option due to imposed organizational management decisions, software engineering is still to provide effective and efficient recommendations. In a dynamic market like the current one some practices, processes, and techniques provided in literature might be limited. For instance, techniques to derive software requirements from formally defined business processes (e.g., EKD (Bubenko, Sirna, and Brash, 2001)) to ensure alignment might be over due when considering that organizations might not be able to keep up-to-date written business processes.

We find it interesting that, in general, the factors and issues mentioned by our respondents are not specific to distributed teams, i.e., these factors and issues are not necessarily caused due to distance. The implication is that co-located teams might also face them. Generalization of our findings has to be considered with caution. We have investigated one single organization and interviewed members located in Brazil only. Despite these limitations, we understand that the variety of roles played by the respondents over the years and their large experience within the organization represent a large set of projects, and as such the results are worth being considered by software organizations with similar settings. Our next step in this long-term research is to quantitatively investigate historical project data from ORG to identify which project characteristics promoted high-performance.

Acknowledgement

We would like to thank the PDTI Program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91), and the participants for their collaboration and time.

References

Project Management Institute (2013) “A guide to the project management body of knowledge,” 5th ed., Newtown Square, Project Management Institute, 589p.

Herbsleb, J. and Mockus, A. (2003) An Empirical Study of Speed and Communication in Globally Distributed Software Development. IEEE Trans. on Software Eng., vol. 29, no. 6, June, pp. 481-494.

Corbin, J. and Strauss, A. (2007) “The basics of qualitative research: Techniques and procedures for developing grounded theory,” Sage Publications, Los Angeles, USA, 3rd ed., 400 p.

Creswell, J. (2008) “Research design: Qualitative, quantitative, and mixed methods approaches,” Sage, Thousand Oaks, USA.

C. Larman (2003) “Agile and iterative development: A manager’s guide,” Addison Wesley, New York, USA, 368p.

Al-Rawas, A. and Easterbrook, S. (1996) “Communication problems in requirements engineering: A field study,” In: Proc. of the Conf. on Professional Awareness in SEng, London, pp. 46-60.

Damian, D. and Zowghi, D. (2003) “RE challenges in multi-site software development organisations”, Requirements Engineering, vol. 8, no. 3, pp. 149-160.

Bubenko, J., Sirna, J., and Brash, D. (2001) “EKD user guide,” Dept. of Computer and Systems Sciences, Stockholm: Royal Institute of Technology.