

Why a CMMI Level 5 Company Fails to Meet the Deadlines?

Darja Smite, Cigdem Gencel

Blekinge Institute of Technology, Ronneby, Sweden

{Darja.Smite | Cigdem.Gencel}@bth.se

Abstract. Reliable effort and cost estimation remains to be a challenging issue even for mature software organizations. Although, these organizations collect historical data to base their future estimates, changes in circumstances (such as application type, development platform, etc.) prevent their successful utilization. As a result, companies often suffer from underestimated and unrealistic schedules. Managing software projects that involve a large number of globally distributed stakeholders makes estimation and planning even more challenging. Related studies show that even knowledgeable project managers often underestimate hidden costs and sources of delay associated with distributed development. Therefore, management activities such as estimation of development effort, planning and control require special attention. In this paper we discuss experiences gained from a highly distributed software project, which aimed at development of a product based on a new platform and architectural solution. The project was conducted in a CMMI Level 5 company and still failed to meet initial plan constraints. We thus provide an overview of management decisions in the light of their consequences, and discuss potential areas of improvement.

Keywords: Software Project Management, Effort Estimation, Distributed Software Development, Global Software Engineering

1 Introduction

Considerable effort has been put forth by the software engineering community to define and improve the software engineering process as well as its proper management. Project management and software engineering frameworks such as [1], [2] accumulated considerable amount of knowledge to facilitate project managers. As appreciated by other engineering disciplines, the benefits of process improvement on projects success have also started to be realized by the software engineering organizations.

Moreover, unique tools and techniques were developed to address challenges related to management of software development projects [3], [4], [5]. Many investments have been especially directed to support project upfront planning activities, in particular to find reliable duration, effort and cost estimation models for

2 Darja Smite, Cigdem Gencel

project outcome prediction. Unfortunately, although different models are reported to be successfully used by different groups and for particular domains, they do not have unanimous acceptance by the software community as being not performing well enough.

Therefore, the mature and wise companies have started to collect historical data and use it for future project predictions. However, when a team is introduced to new tasks, technologies, engineering methods or settings, historical data can be of little use. Moreover, diverse and changing nature of application types, which a company develops, limits the usage of the collected historical data for reliable estimation.

Most of software nowadays is developed by global software teams. Effort estimation is recognized as one of the top problems in globally distributed software projects and two thirds of these projects were reported to suffer from faulty effort estimates [6]. Distributed work is relatively new and is recognized as considerably more complex than even the most difficult collocated projects [7]. Thus many problems associated with geographic, temporal and cultural distance take inexperienced project managers by surprise. As a consequence, a large number of project failures plague the global software industry [8].

In this paper we discuss a software engineering project that failed to meet the deadlines due to underestimated scope and unforeseen consequences of corrective actions. The studied project team experienced new tasks, new technologies, new engineering methods and new settings of a global highly distributed software project.

The paper is organized as follows. Section 2 provides case organization and project description. In section 3, we illustrate measurements gathered during the project, analyze project management challenges and actions taken by the software organization, and discuss reasons of failure. Finally, section 4 concludes the study.

2 Case Study

2.1. Methodology

This study is a single-case study [9] and the object of our investigation is a recently finished software development project that experienced huge problems with underestimated effort and duration. The study is based on multiple data sources: interviews and project documentation and is exploratory in nature. This means that the researchers did not have a preconceived theory in mind, but rather focused on understanding the reasons behind project events and their influence on the project performance, and let conclusions evolve through data analysis.

During this study, we used multiple data sources in our analysis. We have had continues discussions with the team leader who was responsible for software development that were held in person, through electronic means (Skype) and through email communication. We have also had access to various sources of project information, such as project plans, measures, and post-mortem analysis data. At the end, our conclusions were reviewed and approved by the project team leader and project manager from the case organization.

In our case study, we have addressed the threats to construct validity by involving two researchers in the data analysis. Internal validity was addressed by approving our observations with the project managers from the studied company. A possible threat to internal validity is the limited number of team members involved in the investigation. However, we believe that reliable project documentation was the key source of information for our research questions. We thus generally believe that there is no speculation or subjective judgment in our conclusions.

This study serves as an industrial experience report and lessons learned.

2.2. Case Organization Description

The context for this study is one of the top 100 IT services companies in the U.S. that evolved through acquisition and is spread across several locations including software development centers in Eastern and Northern Europe (for confidentiality reasons we do not disclose the true name of the company). The studied company offers outsourcing services for customers around the world and is marketed as a leader in distributed agile development. While it extended its operation in global markets, quality certification has been given a high priority and the processes have been both CMMI Level 5 and ISO-9001 certified for stability, efficiency and maturity. At the present time the company has around 1500 employees and offers such services as global software development, application and architecture reviews, component-based development, enterprise application integration and migration, as well as maintenance.

2.3. Case Project Description

During this study we have investigated a project that was particularly interesting due to a unique combination of new approaches applied. This was a distributed agile project that delivered web-based software application for a call centre built on a new technology platform and architectural solutions. For more detail see Table 1.

Table 1. Project description

Project Characteristic	Description
Application type	Web-based software application for a call centre
Application size	60,000 SLOC PHP & JavaScript; 1,000 SLOC Java; 57, 000 SLOC Java code including copied third part code (SLOC includes comments and empty lines)
Technology platform	Java, PHP, CSS, JavaScript
Development tools	Eclipse IDE
Development methodology	Incremental development with application of Scrum

The studied project at times was highly distributed across six offshore development locations all subsidiaries of the studied company and involved a distributed customer (Fig. 1). The project team consisted of a Project Management Team, a Business

4 Darja Smite, Cigdem Gencel

Analyst Team, a Quality Assurance Team and Development Teams that involved experts from several locations.

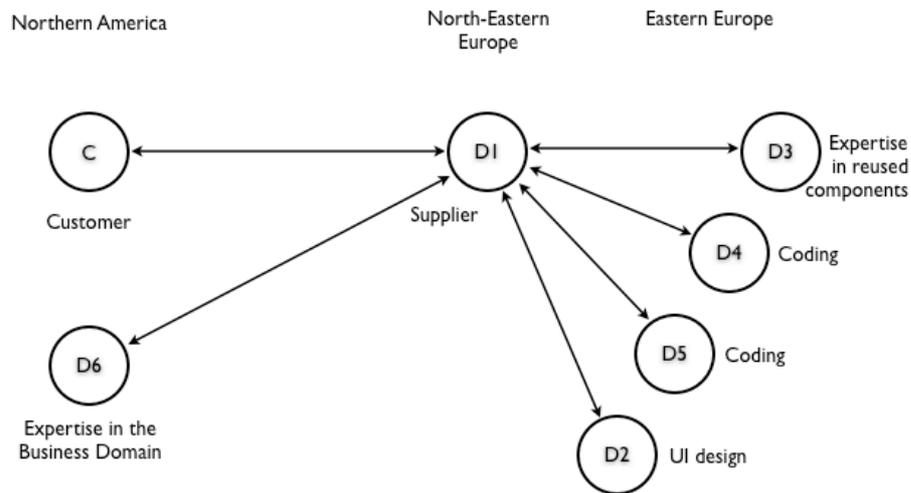


Fig. 1. Sites involved in the project

2.4. Project Management

As the studied company is CMMI Level 5 and ISO 9001:2000 certified, the studied project initially followed a set of best practices. Project manager and a software development team leader were involved in management activities. When the team lacked experience with the development platform, architecture and programming language experts from remote locations were involved at time to fill the gaps in the necessary expertise. As a part of the strategy of delivering quick and qualitative solution, the software product was to be built on reusable components that were developed by the offshore locations of the company.

Since the company was branding itself as the leader in distributed agile development, the project initially chose to follow Scrum practices for project management. Project tasks and activities were scheduled in a project plan that was regularly updated. In illustration, by the end of the project after 10 month of development, the plan was updated 35 times. Project management was supported by a variety of tools that collect measures, e.g. status reports on task progress. The measures collected during the project to ensure quality are given in Table 2.

Table 2. Collected Measures for the Project

Collected Measures	Values
Number of defects /number of features	10
Number of defects found by the Customer	87
Number of defects found by the Quality Assurance	299
Number of defects found by the Development team	279
Number of Test Plans	9
Number of Test Cases	479

The studied project failed to meet the initial deadlines and effort estimates, which served as a motivation for our investigation. There were significant deviations in effort and duration figures (see Table 3).

Table 3. Project Estimate Deviations

Measures	Estimate	Actual	Deviation
Total estimated effort (person/months)	37,5	66,5	177%
Calendar duration estimate (days)	50	222	404%

The reasons why the studied company failed this project are elaborated in the following section.

3 Findings and Discussion

3.1. Effort Underestimation

Although XYZ has some historical data, which includes past productivity figures, these could not be used to make a good estimate for this case because of the following reasons:

- Development of a new application type with no prior experience;
- Application of a new development platform and architectural solution;
- Underestimated complexity of the product;
- Unforeseen limitations of the reusable components;
- Lack of experience with the chosen engineering methods.

First, the application to be developed was a new kind of application for the development organization. They did not have any previous experience in developing such applications.

6 Darja Smite, Cigdem Gencel

Second, the development organization planned to use a new development platform, for which the developers had no experience. The team also used a new software architectural solution. This decision was based on the necessity to gain first customer reference for the company's future marketing activities.

Third, the product to be developed appeared to be more complex than expected. Poorly described set of initial requirements (20 features - each described by a couple of sentences) resulted in the lack of understanding of the customer needs. Considerable amount of time was spent in discussions between the quality assurance team and the development team about interpretations of requirements. Business analysis, initially planned 15 working days, achieved actual duration of 165 working days. Similarly, several initially unplanned development activities resulted in a huge delay for the product delivery. E.g. an unplanned module to be implemented was planned to require 19 working days, but actually took 96 working days; two other unplanned activities resulted in 167 and 191 working days respectively. In addition, the chosen software components at the end did not match the expectations.

Fourth, a number of components that were developed in other geographically distributed locations of XYZ were assumed to be ready for reuse. Accordingly, no effort was planned for any tailoring activities. However, these assumptions appeared to be faulty and the team spent more than 5 person-months modifying them to fit the needs of the customer.

Finally, the project team had no previous experience with the chosen software engineering method. In particular, Scrum practices were relatively new for the development team and the project management. Despite the potential benefits of the methodology, deviations from the agile principles prevented early reaction to the project challenges. The project was formally organized around 6 sprints with a demo meeting at the end of each increment. The main emphasis in each sprint was put on the new functionality; however the results were not delivered to the customer. Therefore, limited feedback was received. Thus, when the product was finally delivered, it led to significant changes due to unmet needs of the customer. This is reflected in the new versions of the project plan through new development activities.

The project team leader described the project by stating that it contained "buzzword-oriented architecture, assumption-oriented design and excuse-oriented execution". As a result, the required effort and duration of the project increased dramatically. Poorly planned activities further caused user acceptance testing to continue much longer than estimated. It was planned to be 45 working days, whereas it actually took 121 working days.

The management continued to be overly optimistic in planning and the project kept failing to meet the new estimated deadlines. A set of corrective actions did not bring expected results and additional rework was necessary, which resulted in almost exponential growth of necessary time for completion of the project (see Fig. 2).

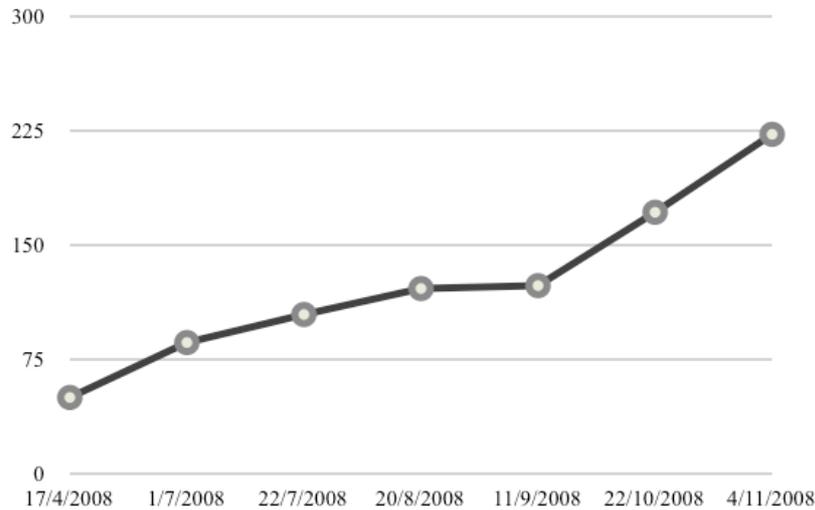


Fig. 2. Evolution of the Estimates with Respect to Plan Update Dates

3.2. Unforeseen Effects of the Corrective Actions

To overcome the underestimated problems such as failure to meet the deadlines and the lack of on-site resources and expertise, the project management team made a decision to involve three other remote locations of XYZ, which actually caused the project inevitably to become a failure.

Initial plan included the tasks to be distributed across three locations of the company. Different activities by type were to be conducted by the most experienced team members. This means that one site had to work on integration of components; another on business analysis and the main site performed the rest of the work. Though development activities were not initially planned to be distributed, project problems forced the project manager to distribute coding tasks across four locations in total. Similar to the mistakes made by other companies discussed in related studies [10], the case project underestimated the communication, coordination, trust and commitment challenges as well. Thus, poorly controlled dependencies resulted in late deliveries and parts of remotely developed pieces of software had to be re-built. As a consequence, once initiated for good reasons, collaboration with these remote locations was not further prolonged.

Although empirical studies show that an agile delivery strategy is recognized to have a positive influence on the scope, timelines and cost of the project [11] and at the same time is found to be useful for reducing communication, coordination, and control problems that have been associated with global software development [12], the project management failed to use these benefits due to deviations from the chosen methodology. In particular, the project started as an agile project and intended to follow Scrum practices, however failed to strictly maintain the practices throughout the project. Close collaboration with the remote locations was challenged by temporal

and geographic distance. The project maintained product and sprint backlogs with requirements and user stories. However, the team leader (Scrum master) individually coordinated tasks for each developer. Lack of experience with Scrum and the pressure of deadlines prevented establishment of a cohesive agile team and application of self-management. In addition, coordination by mutual adjustment across locations was challenged by a lack of previous experience of working together, lack of trust and commitment, and "us versus them" attitude. Finally, daily Scrum meetings across multiple locations were challenged by the limitations of computer-mediated communication and were thus often withdrawn.

Configuration management is another challenging task in globally distributed projects [13]. Due to unplanned involvement of remote team members into development activities, the project infrastructure was not prepared to facilitate distributed software development. Thus, the team experienced significant difficulties in relation to configuration management and required additional effort for integration of the pieces developed by remote locations. Accordingly, the configuration management activity initially planned 5 working days appeared in the newer version of the plan to take 76 working-days and at the end took 154 working-days.

4 Conclusions

In this paper, we discussed a case project conducted in a CMMI Level 5 company that failed to meet initial plan constraints. We elaborated the possible causes for a high-maturity level organization to fail a project and observed that no company is immune to fail addressing unforeseen problems.

The basic reason for the project to become a candidate for a failure was initially underestimated effort and unrealistic schedules. This was unavoidable since the company cannot utilize the historical data they collected in the past years. Moreover, unfortunately, there exists no effort estimation model, which is accepted to successfully address all possible circumstances in software engineering projects, And especially, in distributed work. Therefore, we emphasize a need to collect project related data in benchmarking datasets that can be utilized by the organizations worldwide. Otherwise, only with the local efforts put by the organizations, it seems that these failure stories continue to be told. During the last 10 years, such effort has been put on forming publicly available benchmarking datasets such as the one by the International Software Benchmarking Standards Group (ISBSG) [14] to enable organizations share and use the others' experiences gained. However, without the commitment of the software organizations to provide data to these datasets, improvement in effort estimation area will stay minimal.

Global software development puts new challenges on project managers since geographic separation leads to more difficult effort estimations, project planning and control [15]. Effort required for communication, coordination and integration of the developed pieces of software in the studied case project was underestimated due to over optimistic expectations. These can be explained by lacking experience of working together across locations involved in the project.

Our study also shows that task distribution to remote locations under the pressure of deadlines drove the project to even deeper problems. In the studied case, project management decided to share the pain and stress with remote colleagues. Nonetheless, lack of cohesiveness and commitment from remote colleagues caused failure. However, it is worth mentioning that due to initially underestimated scope, schedule and project staffing, managers are often left with little choice of corrective actions. Therefore, the choice to distribute some of the effort to remote locations seemed natural. And this trend for distributed software development seems to increase as well. Therefore, global software development requires urgent tools and methods that help to overcome the difficulties and enable efficient distributed work.

Unfortunately, the case study does not allow evaluating the suitability and advantages of agile approaches for globally distributed environment, since agile principles and practices at the end were not complied. However, we can conclude, that a lack of previous experience and familiarity with the chosen methodology along with the changing members of the team prevented the project to experience potential benefits. Software community requires more empirical evidence of specific methodologies, such as agile approaches, applied in organizations practicing distributed development.

Acknowledgement

We would like to sincerely thank the team leader from the studied case project, who provided all the data and spent his precious time in discussions that helped us to make this study. This research is conducted within BESQ Research Centre at Blekinge Institute of Technology.

References

- [1] A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Third Edition, Project Management Institute, 2004.
- [2] A Guide to the Software Engineering Body of Knowledge (SWEBOK)”, IEEE Computer Society, 2004.
- [3] Jones, T.C., Estimating Software Costs, McGraw-Hill, 1998
- [4] Boehm, B.W.: Software Engineering Economics, Prentice-Hall, (1981).
- [5] Thayer, H.R., Software Engineering Project Management, Second Edition IEEE CS Press, 2001
- [6] Smite D., “Project Outcome Predictions: Risk Barometer Based on Historical Data”, In proc. of the ICGSE conference, IEEE Computer Society, August 2007, Germany, pp. 103-112
- [7] Karolak D.W., “Global Software Development: Managing Virtual Teams and Environments”, IEEE Computer Society, 1998
- [8] Prikładnicki R., Audy J. L. N., and Evaristo R., "A Reference Model for Global Software Development: Findings from a Case Study", In: Proc. of IEEE Int. Conf. on Global Software Engineering (ICGSE 06), Florianópolis, Brazil, IEEE Computer Society Press, pp. 18-25, 2006

10 **Darja Smite, Cigdem Gencel**

- [9] Yin, R. K., Case Study Research: design and methods, vol. vol. 5, 2 ed. Newbury Park, CA: Sage Publications, 1994
- [10] Ramesh B., Cao L., Mohan K., and Xu P., "Can Distributed Software Development Be Agile?" In: Communications of ACM, 49(10), 2006, pp.41-46
- [11] Chow T., and Cao D.-B. "A Survey Study of Critical Success Factors in Agile Software Projects", The Journal of Systems and Software, 81 (2008), pp. 961-971
- [12] Holmström H., B. Fitzgerald, P.J. Ågerfalk, and E. Ó. Conchúir "Agile Practices Reduce Distance in Global Software Development", In: Information System Management, 23:3, 2006, pp. 7-18
- [13] Battin, R.D.; Crocker, R.; Kreidler, J.; Subramanian, K., "Leveraging resources in global software development", In: IEEE Software, 18(2), 2001, pp. 70-77
- [14] ISBSG Dataset 10, [http:// www.isbsg.org](http://www.isbsg.org), 2007.
- [15] Taxén L., "An integration centric approach for the coordination of distributed software development projects", In: Information and Software Technology No.48, 2006, pp. 767–780