Abstract

Success Factors for Globally Distributed Projects

Siegfried Zopf
Software Engineering & Qualitätsmanagement Zopf
Siegfried.Zopf@sqz.at

The report presents the essence of experience from hundreds of development projects distributed over locations in eight countries with different languages and culture. One success factor is professional project management based on a well defined development process. The report highlights the topics that have special importance in distributed project. Practical tips will be given for how to deal with soft facts as esteem, team building in global projects and traps in communication in foreign languages and how to avoid them.

Keywords

Outsourcing, global development, distributed development, CMMI, development process, quality assurance, human factors

1 Introduction

1.1 Scope of outsourcing

According to a survey of Cutters Consortium more than 80% of the Fortune 500 companies outsource software development. 86% of them go offshore. [Herr03]

In most cases cost cutting is the reason for outsourcing.

- Cost (44%)
- Capacity (20%)
- Know how (13%)
- Time to market (11%)

As there are differentarchitektur

reasons for outsourcing or globally distributed projects there are different ways to organize those projects. I will distinguish between:

- **Distributed projects** within one company, where the responsibility for the project is in the company and the execution of development takes place in some locations may be with local responsibility for subprojects.
- **Outsourcing to another company** may be offshore or - as it is often the case in Europe - to other countries with lower hourly rates on the same continent. Additionally to the things that concern distributed projects come activities concerning selection of a supplier / partner, contractual topics and customer – supplier relationship.

In both cases there are different situations depending on the extent of outsourcing from “headquarters” to a “remote location” be it in the own company or third party. [Cam05]

Minimal responsibility outsourcing

The headquarters decides about project goals, elaborates the requirements, designs the product and the system architecture and is in charge of all management aspects like project management, quality assurance management and configuration management.

The remote location makes the detailed design (if at all), the coding and component test.

System integration, system test and acceptance is done back home in the headquarters.

With minimal responsibility outsourcing the domain knowledge as well as the software engineering knowledge persists within the headquarters. As coding is only a small part of a development project the advantages are relatively small but also the risk is small and in case of problems it is not so difficult to in-source. Even with minimal responsibility outsourcing the experience with implementation (all those information that is not documented about tools, platforms etc.) does not come back to the designers and the design becomes less realistic over time.

Large scale outsourcing

The headquarters decides about project goals, elaborates the requirements and mostly the design of the product. Also overall project management and quality assurance management rests in the headquarters. The whole technical jobs like system architecture, detailed design, coding, integration and testing are executed at the remote location which is responsible for the development including subproject management. At the end of the project headquarters only execute the acceptance test of the system tested product.

This model contains the big danger for the headquarters of loosing the qualification for the technical job which may lead to dependency on the supplier. Know how is transferred to the supplier. This model should not be applied for strategically important software because the strategic advantage can get lost from one day to the other when the supplier changes his partner.

The risk of loosing technological know how can be reduced when one or more subprojects are still done at the headquarters. In case of skipping the contacts with a supplier there is still a competence for development and maintenance at the headquarters.

I will not discuss the model where you send a requirement spec to a supplier and wait for the product to be delivered in time.

1.2 Role of a Beach head

Even in large scale outsourcing the headquarters have to have beach head functionality for controlling the remote location during the development phases. When it is a large scale outsourcing project some make the error to think: „we have sent all the requirements to our partner and do not need time and capacity for the project any more".

2 Development methods matter

Using a well defined development process always matters, even if you work in one location. But it becomes more important in distributed development, independent of the type of cooperation. [Kain98], [Höhn08]

2.1 Definition of the Product

In large scale outsourcing projects you need a well specified requirements document because the customer or user usually is not on site at the remote location. As we will discuss later there are many reasons why remote developers often do not clarify ambiguous specs or open topics in specs but try to implement something they think is a good solution. In minimal responsibility outsourcing projects the same is true for architectural specifications. Long time ago we saw that the waterfall life cycle model in its purest form did not work. Only documents were handed over from the experts of one phase to the experts of the next phase without additional communication e.g. requirements engineers hand over the requirement spec to the architects. Why should it then work when the developers are on different locations?

Even with the best specifications there will rest ambiguities and open questions. The worst case is that the developers make assumptions: The good case is that they ask questions. But for this they have to have someone who is able to answer. Therefore a "beach head" is necessary.

2.2 Project Management

A clear documented definition of the responsibilities in projects is indispensable. Is everything controlled by the beach head or do you install subprojects. It is nothing new compared with a project with subprojects that you do in one building. But it is more important because you detect problems later, when they have become more expensive. And it concerns not only project planning and control but also technical roles, quality assurance management, configuration management, test etc. [Früh01], [PMI05]

2.3 Risk Management

Especially with new cooperation partners not everything will run as you think. Therefore risk management is more important in distributed projects then in the projects you are used to do.

Frequent sources of risks are: communication, langue difficulties, cultural difference, tacit assumptions, knowledge of application domain, understanding the common process (if the remote developers have to use the process of the beach head). This comes in addition to the usual risks. [DeMa03]

2.4 Quality Assurance

Quality assurance is another one of the topics that would be important in projects but are neglected very often. In distributed projects quality assurance and the role of a quality assurance manager (QAM) becomes crucial. [Wall01] Depending on the type of cooperation you have either only one Quality Assurance Manager (QAM) in the beach head or you have an additional local QAM in each remote location. In the first case it will be very difficult for the QAM to get reach insights how the developers do their job, how they stick to the processes – in reality, not only the formalisms. A local QAM has the chance to see early indicators of problems. It is also easier to organize quality assurance measures like reviews and observe how they are done.

When you have one QAM for all locations he/she can call in the documentation of quality assurance measures and get an impression on how the project runs. Therefore he/she has to study the documents and not only collect them. Review Report: Are peer reviews done professionally? Was the review object o.k.? Test plan: How are the tests planned? Test report: Have the planed test cases been executed and how many defects were detected etc.

2.5 Configuration Management

I was very surprised when I attended a presentation at an international conference in Düsseldorf 2006 about optimisation of distributed projects, an experience report. The big thing they did was introducing Configuration Management (CM) in the project. And it really helped and solved a lot of problems they had before. I was surprised because I thought CM is standard for 20 years but it showed again that the engineering discipline is not yet basic knowledge in software development organisations. [Berl92] What is important for us is that a distributed project without CM runs into troubles. You do not need risk management, you have a problem.

3 Is CMMI Maturity Level 5 the Silver Bullet?

The Capability Maturity Model Integration (CMMI) is a reference model for the maturity of software development organisations with five maturity levels. It was elaborated at the Software Engineering Institute (SEI) at the Carnegie Mellon University [SEI06], [Hump89]. CMMI is a quai standard all over the world. In appraisals the capability of a supplier can be assessed.

When the development process is so important many contractors have the idea: let's look at the CMMI maturity level of the supplier and select the one with the best ML.

The Idea is good but there are some constraints. Does the partner really work on the promoted maturity level (ML)? Has the beach head the maturity to cooperate with a high level organisation?

3.1 Maturity Level of a development organization

Be careful when you get the informal information “We are on ML5”. What part of the organization was appraised? The department that you work with or another department that may be only 10% or less of the organization and therefore the ML is not typical for all projects. Another question is: who did the appraisal? An appraiser accredited by SEI or someone else may be even the own organization in a self appraisal. Therefore it makes sense to look at the certificate and collect additional information about the partner. Best is to contact clients of the supplier.

When you have an understanding of CMMI you will see in the first days of the project what level the partner has.

3.2 Maturity Level of the beach head

Employing an organization with ML5 only makes sense when the beach head itself has at least a good understanding of ML3. Otherwise the partner will not be able to proceed as required by CMMI ML5. In project cooperation the leading part has to understand and act on ML3. It does not work when the beach head says: ”We do not have the money for CMMI here in Europe. We do our part as we have done all the time but the others shall develop the outsourced part on ML5. They have cheap developers and have the time and capacity to do everything required by CMMI.”

4 People are People in all Corners of the World

Software development is a people business not a machine business. People are the most important “resources” in a development project. We know that the success of development projects depends more on motivation, team work and communication than on hardware, programming languages and development processes. The development department Siemens PSE, where I spent my professional carrier until recently, has 20 development locations in 8 countries of the world. Globally distributed projects are the rule, not the exemption. That is why big effort was spent to find out what makes distributed projects successful. [Acke00]

Motivation, team work and communication are impaired strongest in distributed projects. Hardware and software tools are nearly not influenced.

To be motivated you have to know the goals, have to be included in decision making and have to get recognition by the team and by the management. Motivation and team building may arise sometimes even without much systematic effort and investment of money when the team is in one place. But in distributed teams it never arises without systematic effort and it costs money. A kick off meeting produces additional travelling costs. When you think you can safe this money try it. But measure the failure cost of the project. Analyse the date and the next time you will invest in team building.

Cultural aspects play an important role. Therefore inform yourself about the differences. E.g. in some cultures a person usually will not say “no” but will use other ways to express his disagreement: e.g. “there is another possibility to do this”. Therefore “he did not say no” is not an agreement. Listen to what he says and think about it.

Esteem is important for people in all corners of the world. To esteem your partner is the precondition for successful collaboration. It has the same importance in a local project. But to esteem someone you know, who has the same lifestyle, the same education, loves music and actors you also know comes sometimes automatically and is easier than to esteem someone who is different in many aspects. In this case it does not come automatically and you have to strive for understanding. Care to get information about the interests and achievement of someone and also the circumstances of his life. Be careful not to awake bad feelings because of the different levels of standard of living. But the esteem has to be real. Nobody can play it like an actor. Body language, accent and the like will betray you. But esteem is the basis for successful communication.

5 Communication

In the flow of information through the project as well as in team building language is in the centre. One or sometimes both partners have to communicate in a foreign language. Usually the ability to express thoughts and feelings in a foreign language is lower than in the native language. Also the understanding of thoughts and other messages is lower. Therefore in a globally distributed project attention has to be spent on communication. It is an important task of the management to look if the formal communication runs well. Every team member has to be aware of this challenge and has to be alert in everyday communication written or verbal.

In many cases written communication is better than verbal. The partner can analyse the text with a dictionary what he cannot do after a conversation. But there will be conversations and the question is how to check if the message did arrive. “Did you understand?” is a stupid question. Nearly everybody will say yes. Better is mirroring. “What did you understand?” and than the partner will explain with his own words what he understood. Then you know if he understood or not. It sounds a little bit like Imago Therapy but it works. I do not object to conversations and it is good practice to write down the result of the talk in file note. In cross cultural projects it is useful to follow this practice, write a file note and additionally send your not to your partner.

To be sure observe what the partner does shortly afterwards. Do not wait 3 month if he will bring back the right thing but check after one week if he is on the right way.

6 Summary

In a nut shell the success factors for globally distributed projects are not rocket science:

The most important success factor is professional project management based on a well defined development process.

The architecture of the system and distribution of tasks to different locations should be mapping as much as reasonable possible.

Optimizations have to treat always the global project not the local parts. Watch for local optimizations at the cost of other sites and turn it around.

Plan higher effort (beach head, laborious communication…) than you would for a local project. Higher effort does not mean higher costs. But planning a distributed project with the same effort that was estimated for execution in one location will generate problems and higher costs.

Be always aware of cultural differences and put more emphasis on systematic measures and activities that cultivate human factors and never forget esteem.

Literature

[Acke00] *Ackermann Gerhard / Lutz Benedikt, Cross-Regional Cooperation in PSE Projects, Best Practice Report, Siemens internal paper, 2000*

[Berl92] *Berlack H.R., Software Configuration Management, John Wiley &Sons,1992*

[Carm05] *Carmel Erran, Offshoring Information Technology, Cambridge University Press, 2005*

[DeMa03] *DeMarco Tom / Lister Timothy, Waltzing with Bears, Dorset House, 2003*

[Früh01] *Frühauf Karol / Ludewig Jochen / Sandmayr Helmut, Software-Projektmanagement und -Qualitätssicherung, vdf Hochschulverlag, 2001*

[Herr03] *Herron David, The Impact of CMM on Outsourced Software Development, Cutter Consortium, Executive Report Vol.4, No.5, 2003*

[Höhn08] *Höhn Reinhard / Höppner Stephan, Das V-Modell XT, Springer, 2008*

[Hump89] *Humphrey Watts S., Managing the Software Process, Addison_Wesley,1989*

[IEEE04] *SWEBOK Guide to the Software Engineering Body of Knowledge, IEEE, 2004*

[Kain98] *Kaindl Hermann / Lutz Benedikt / Tippold Peter, Methodik der Softwareentwicklung, Vieweg, 1998*

[PMI05] *Project Management Institute, A Guide to the Project Management body of Knowledge, PMI Books, 2005*

[SEI06] *Software Engineering Institue, CMMI for Development, Version 1.2, CMU/SEI 2006-TR-008 ESC-TR-2006-008*

[Wall01] *Wallmüller Ernest, Software Qualitätsmangement in der Praxis, Hanser, 2001*

Author CVs

Dipl. Ing. Siegfried Zopf

Mr. Siegfried Zopf is self employed consultant for software engineering and quality management. He studied technical physics at the Vienna University of Technology in and started his professional carrier as software developer with Siemens Program and System Engineering (PSE) in 1977. In 1984 he changed to the central quality management department of PSE where he participated in the elaboration of the system development method SEM. He was responsible for the implementation of SEM and the advancement of software engineering and quality management in PSE (ISO 9001 certification, CMMI and EFQM Assessments) at 20 locations in 8 countries.
Besides he is lector at the Vienna University of Technology and at universities of applied science. Since 2005 he is president of the Austrian Association for Software Quality