# Impact of Organizational Structure on Distributed Requirements Engineering Processes: Lessons Learned

Brian Berenbach
Siemens Corporate Research, Inc.
755 College Road East
Princeton, New Jersey 08820
+1 609-734-6500
brian.berenbach@siemens.com

## ABSTRACT

The requirements engineering program at Siemens Corporate Research has been involved with process improvement, training and project execution across many of the Siemens operating companies. We have been able to observe and assist with process improvement in mainly global software development efforts. Other researchers have reported extensively on various aspects of distributed requirements engineering, but issues specific to organizational structure have not been well categorized. Our experience has been that organizational and other management issues can overshadow technical problems caused by globalization. This paper describes some of the different organizational structures we have encountered, the problems introduced into requirements engineering processes by these structures, and techniques that were effective in mitigating some of the negative effects of global software development.

## Categories and Subject Descriptors

D.2.1 [**Software Engineering**]:Requirements/Specifications – *elicitation methods, methodology, tools.*

## General Terms

Management, Measurement, Documentation, Verification.

## Keywords

Requirements Engineering, organization, analysis.

## 1. INTRODUCTION

Prior research in distributed requirements engineering has tended to focus on facilitation techniques [0] [2] [3]. Others have researched the nature of asynchronicity in distributed environments [4]. There has also been research into the impact of culture on facilitation and negotiation [5][6]. The use of enabling tools has also been reported in [7][8]. Of necessity, many studies on distributed requirements engineering have used collaborative student projects to obtain information [9]. Siemens Corporate Research, in collaboration with several universities is conducting such a study [10]. However, very little has been reported on organizational issues and their impact on the effectiveness of projects in the large, and specifically distributed requirements engineering efforts.

The requirements engineering (RE) competency center at Siemens Corporate Research in Princeton has had the unique opportunity to participate in large, global projects with different organizational structures. Each structure brought different challenges, benefits and issues. In the following sections, I will describe some of the structures we encountered, problems caused by the organizational structures, and, finally, I will suggest techniques for mitigating the problems encountered.



## 2. ORGANIZATIONAL STRUCTURES

Organization in this paper has a very specific meaning. It refers to the leadership and/or management of a specific area. If for example, there are analysts at a remote site but they all report to and are managed by one organization, then the requirements elicitation may be distributed but the organization is not. Figure 1 shows some of the more common arrangements in global software development efforts. "A", "D", and "I" represent analysis, high level design, and implementation (including low level design) respectively. Where the boxes touch it means that the processes are co-located (at the same site and organizationally integrated). Note that integration testing has been left out (It is not the focus of this report).

Project management structure also varied on observed projects, and given any of the sample structures shown, management might have been co-located with any of the other development process sites or spread over several. Where management organization is relevant, it will be mentioned in the narrative.

## 3. CO-LOCATED (SINGLE SITE ) STRUCTURE

Even when processes are completely co-located (structure 1 in Figure 1) there can, of course, be problems. Prior to joining Siemens I consulted on a project where all process activities took place in a single facility. However, the project manager was in department A, and resources and information were required from department B (names changed to protect the innocent and guilty). Since the managers of departments A and B had different reporting chains, they had differing views of staffing priority. As a result, the project did not get resources in a timely manner and was eventually canceled. Consequently, the reader should understand that, for this paper, colocation implies a single site AND a single chain of command for all project activities.

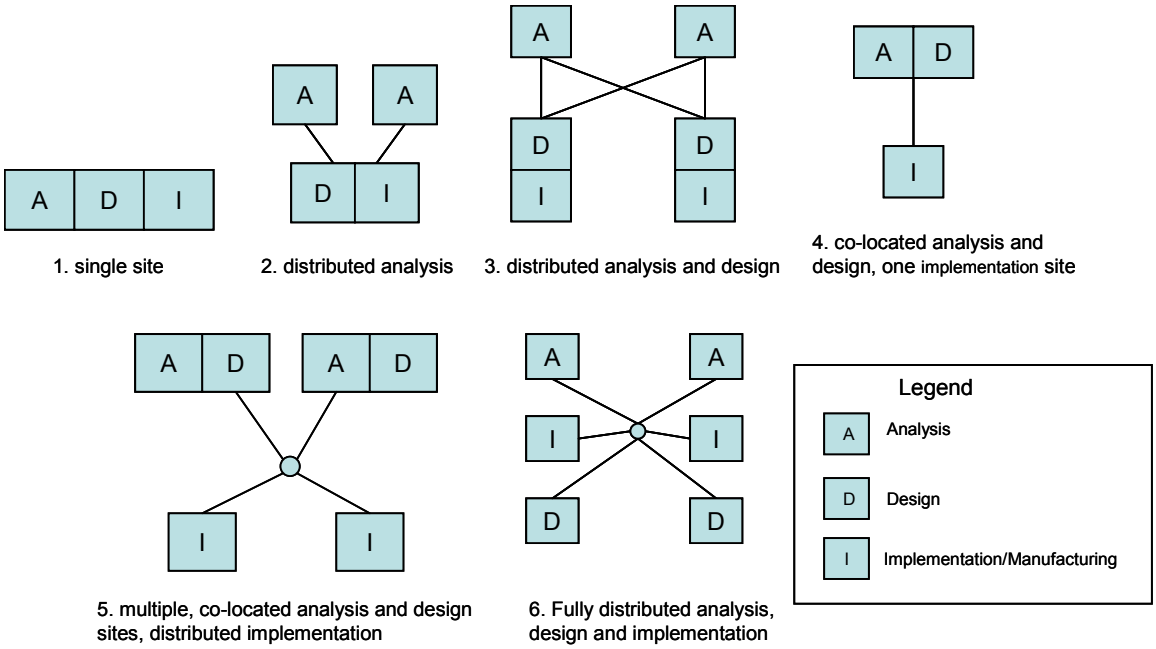


**Figure 1 Example Organizational Structures**

# 4. DISTRIBUTED ANALYSIS

Distributed analysis (structure 2 in Figure 1) carries with it several challenges. Some of them include:

- Technical Management
- Style Variation
- Change Management
- Cross-location review

The requirements development effort is considered distributed when there is more than one requirements analyst (not to be confused with subject matter expert) capturing the requirements, and they are in two or more locations.

## 4.1. Technical Management

One observed issue with technical management on a large project with distributed analysis was the lack of a coherent set of processes and missing overall leadership. Distributed effort implies distributed chain of command and that can cause all kinds of problems (see paragraph 4.4 below).

## 4.2. Style Variation

When the analysis effort is distributed, the analysis techniques may vary. This can result in the analysis work products such as use cases, flow charts, etc. being in different media, which can result in traceability problems ( Figure 2).

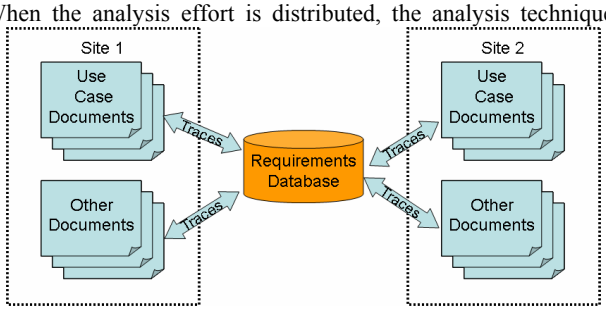


**Figure 2 Multisite Traceability Problems**

## 4.3. Change Management

Change management with multiple customers and analysts and varying sites can be a daunting task without effective policies. On one project we observed analysts bypassing the change control process by continually adding new requirements that were really modifications of existing requirements. At one point, as much as two thirds of the requirements in the shared database had been superseded by new requirements.

An unintended byproduct of distributed analysis can be poor communication and lack of clear project control. This was illustrated by the makeup of the change control board (CCB) on an IT project. I had asked the project manager how many people there were on the change control board. She replied that there were two, the customer representative (actually the major stakeholder representing the customers) and one of the technical vice presidents (vp). During a meeting at which this vp participated, he found out for the first time that he had been on the CCB for a year. Later, the CCB was revised to include the appropriate roles, resulting in improved project execution.

## 4.4. Cross-location Requirement Reviews

There are many problems that can occur when conducting requirement reviews, and these can be exacerbated with globalization. Reviews are difficult enough as it is:

- Reviewers often do not have time to read the material
- Requirements may not be at the right level
- Poor traceability may make it difficult to check references
- Writing style can be variable
- Quality attributes may be missing


**Figure 3 Progressive Communication Issues**

With distributed analysis, other problems arise. They include:

- Failure to conduct global reviews
- Communication issues, especially time zone problems
- Cultural issues
- Underestimating the time it takes to conduct reviews

When analysis personnel are not co-located, they may disassociate from work done at another location and not be willing participants in cross-functional reviews. Also, if reviewers must be up at 3 am to conduct a Telco review, they may not be at their physical best, and might lack some enthusiasm.

We have also noticed cultural problems when outsourcing to some countries. Although the analysts and subject matter experts may be skilled, the lack of physical proximity can lead to a certain timidity and passiveness during the review process.

## 5. DISTRIBUTED ANALYSIS AND DESIGN

Several projects we have worked on involve outsourcing to Siemens organizations in India, China, and Eastern Europe. A typical outsourced project might have the subject matter experts and analysts co-located, with the design and implementation outsourced to a Siemens overseas location (structure 3 in Figure 1). This structure may increase the risk of a negative outcome since:

- Analysts creating specifications for in-house projects may not have the requisite skills for creating sufficiently detailed specifications for outsourcing (e.g. too many assumptions, lack of completeness)
- The designers and developers most likely have no knowledge of the domain, especially if this is the first collaboration.
- Time zone issues can interfere with communication.
- Cultural issues may also arise [5][6].

On one medical software project, it was necessary for the project manager (acting as technical lead) to spend considerable time at site being proactive in insuring that the development team understood the requirements. Just conducting remote meetings or periodic reviews turned out to be insufficient to insure a positive outcome.

On a major outsourced project with distributed responsibility, there were personality clashes between the local and outsourced architects, resulting in communication lapses that impacted productivity and product quality.

Another issue that turned out to be a problem was the assumption of project and technical management responsibility by product managers. Unfortunately, knowing the domain and/or the product really well did not always translate into effective project management. Because of the organization structure, the product manager was at a higher management level than the project or technical managers, and they were unable to mitigate the consequences. As a result, in each case where this happened, the product manager eventually was replaced, not because they had done a poor job as product manager, but because they had not been effective in the other roles that they had taken on.

Organizational structure can have a big impact on the availability of resources, and, consequently, project schedule. A medical project was using staff from another company. The availability of the resources was assured based on a verbal agreement between the Siemens product manager and the other company vice president. Unfortunately, the minute the other company had an internal crisis our resources disappeared.

In general, problems with distributed analysis and design tend to stem from communication difficulties and leadership lapses. With distributed design there is the "assumption" that design at other sites is going well. The lack of a strong, skilled centralized architecture role will totally kill the project.

## 6. DISTRIBUTED ANALYSIS, DESIGN AND IMPLEMENTATION

Unfortunately, the probability that a fully distributed project will have a positive outcome is low. If $P_f$ is the probability of failure, then every increase in organizational complexity increases the probability of a negative outcome. That is:

$$P_f = f(P_a, P_d, P_i)$$

**Equation 1 Probability of failure**

where Pa, Pd, and Pi represent the probability of failure because of distributed analysis, design and implementation (or manufacturing) respectively. Having an organizational structure

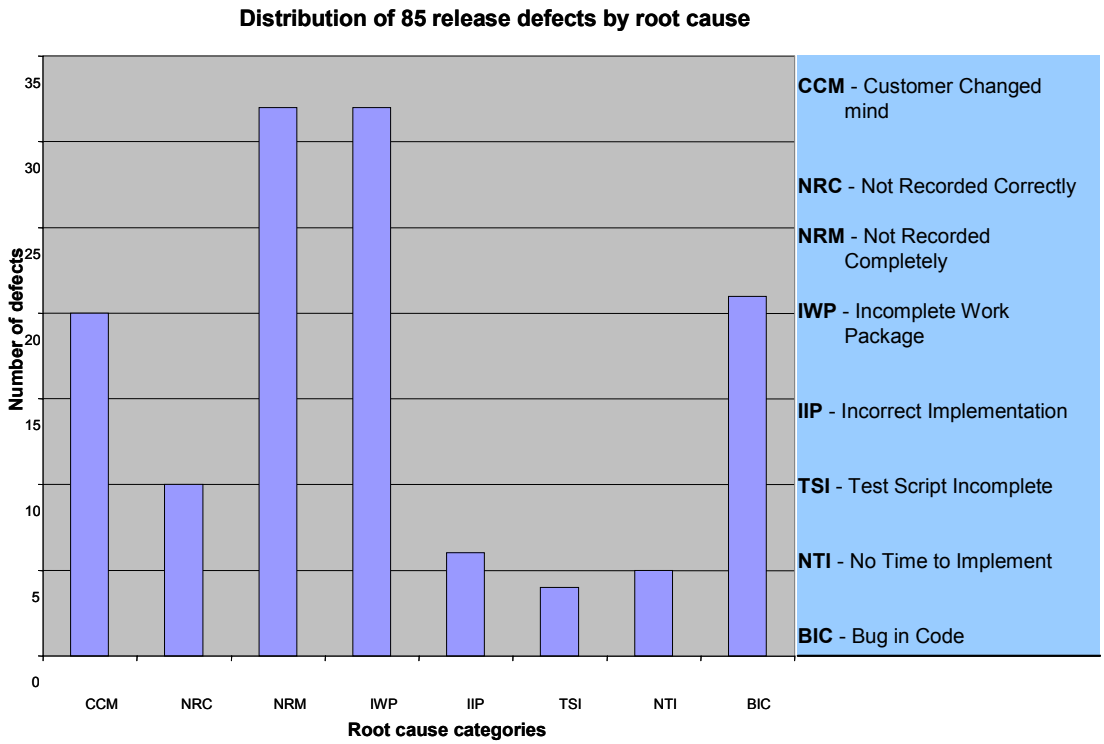


**Figure 4 Lifecycle Problems Caused by Distributed Communication**

(such as that shown in Figure 1, structures 5 or 6) may carry with it significant risks.

Communication problems between analysis, design, and implementation staff can be exacerbated by poor traceability. One of the biggest implementation issues found to date is that of the implementer having to resolve requirements issues not resolved during analysis or design (Figure 3).

A real-world example of communications problems arising from project distribution (Figure 1 structure 6 ) can be seen in Figure 4. SCR had been asked to conduct a review at a Siemens organization that seemed to be having difficulty. Analysis, design, implementation and testing were done at a variety of locations. Initially, the defects shown were reported as "software" bugs. After conducting a root cause analysis it became clear that a large percentage of the problems were requirements related. There were a surprising number of defects caused by tests being done improperly as requirements had not been correctly understood by quality assurance creating test scripts. The staff in all areas was highly qualified, and each organization fully understood its technology domain. The problems came about because of organizational missteps and a missing layer of leadership.

## 7. SUMMARY AND CONCLUSIONS

In every distributed project in which we participated (over 10) most of the issues could be traced back to problems with organizational structure and/or management. There were, of course, issues with elicitation and cultural misunderstandings, but, surprisingly, they were completely overshadowed by organization and management problems.

The issues we have encountered, most severe listed first, are:

- Unqualified Project Architect (lacking in leadership and skill).
- Multiple or diffused chain(s) of command.
- Core staff unskilled in handling distributed projects.
- Lack of central, authoritative leadership in requirements engineering, especially in the area of customer management, resulting in poor coordination and cooperation of analysts.
- Failure to follow the documented project processes.

**Table 1 Observed Organizational Problems**

| Organizational Structure | Observed Problems |
|---|---|
| 2 | No cross-location reviews; varying documentation styles; weak configuration management |
| 3 | 2 + communication difficulties between analysis and design organizations. Pushing detailed analysis to remote development organizations. |
| 4 | Late feedback; requirements suitable for in-house development incomplete and confusing to remote sites. |
| 5 | 3 + architectural inconsistencies due to poor communication or weak architectural management. Lack of coordination between development sites; difficulty planning integration testing. |
| 6 | 5 + 4 + lack of overall central management leading to project spiraling out of control |

There has been some success in mitigating the above issues summarized in Table 1) on Siemens projects using the following techniques:

- Strong leadership at the project management and technical lead levels.
- Face-to-face relationship building and training prior to project initiation
- Clear chain of command
- Minimizing the use of non-Siemens staff
- Clear definition of roles and responsibilities
- Well defined, well understood requirements engineering process
- Frequent collaborative reviews using web or network hosting tools.
- The identification of a liason at each site with responsibility for all communication activity.

Any company embarking on global development projects, be they software, hardware or a combination, should initiate the effort fully cognizant of the risks they will face. Strong leadership skills, strong technical skills, a single chain of command and coherent distributed structure, and of course, skilled staff (with prior distributed experience) will definitely improve the probability of a positive project outcome.

# 8. REFERENCES


[1] Damian, D.E.H.; Eberlein, A.; Woodward, B.; Shaw, M.L.G.; Gaines, B.R., "An empirical study of facilitation of computer-mediated distributed requirements negotiations", *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering, 200*1 , vol., no.pp.128-135, 2001

[2] Damian, D.E.; Eberlein, A.; Shaw, M.L.G.; Gaines, B.R., "An exploratory study of facilitation in distributed requirements engineering," *Requirements Engineering* (2003), vol.8, no.1, p.23-41.

[3] Macaulay, L.A., "Seven-layer model of the role of the facilitator in requirements engineering," *Requirements Engineering* (1999) , vol. 4, no.1, p.38-59.

[4] Campbell, C.L.; Van de Walle, B., "Asynchronous requirements engineering: enhancing distributed software development," *Proceedings of the ITRE2003* pp. 133- 136, 11-13 Aug. 2003

[5] Damian, D.E.; Zowghi, D., "An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations*," Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003* , vol. 10 pp.-, 6-9 Jan. 2003

[6] MacGregor, E.;Hsieh, Y. and Kruchten, P., "Cultural Patterns in Software Process Mishaps: Incidents in Global Projects", *Proceedings HSSE'05*, Saint-Louis, May 16th, 2005.

[7] Herlea, D.; Greenberg, S., "Using a groupware space for distributed requirements engineering ," Enabling Technologies: Infrastructure for Collaborative Enterprises, 1998. (WET ICE '98) Proceedings., Seventh IEEE International Workshops on , vol., no.pp.57-62, 17-19 Jun 1998.

[8] Evaristo, R.; Watson-Manheim, M.B.; Audy, J., "e-Collaboration in distributed requirements determination," *International Journal of e-Collaboration (2005)*, vol.1, no.1, p.40-55.

[9] Johansson, C.; Dittrich, Y.; Juustila, A., "Software engineering across boundaries: student project in distributed collaboration," *IEEE Transactions on Professional Communication*, vol.42, no.4pp.286-296, Dec 1999

[10] Mullick, N., Bass, M., Paulish, D. and Sangwan, R., "Global Studio Project: A case study in managing globally distributed software development", *Submitted to IEEE Software*, January 2006.