Looking for Reasons behind Success in Dealing with Requirements Change

Joost de Wit
University of Twente
The Netherlands
email: joostdewit@gmail.com

María Laura Ponisio
University of Twente
The Netherlands
email: m.l.ponisio@utwente.nl

Abstract

During development, requirements of software systems are subject to change. Unfortunately, managing changing requirements can take a lot of time and effort. Yet some companies show a better management of changes in requirements than others. Why? What is it that makes some projects deal with changing requirements better than others? We pursue the long term goal of understanding the mechanisms used to successfully deal with change in requirements. In this paper we gather knowledge about the state-of-the-art and the state-of-practice. We studied eight software development projects in four different companies –large and small, inclined toward structured and toward agile principles of development–, interviewing their project managers and analyzing their answers. Our findings include a list of practical (rather than theoretical) factors affecting the ability to cope with small changes in requirements. Results suggest a central role of size as a factor determining the flexibility showed either by the organization or by the software development team. We report the research method used and validate our results via expert interviews, who could relate to our findings.

Keywords: requirements change, software development, case study

1 Introduction

Many, if not all, real life software development projects must deal to some extent with changes in requirements, scope, and technology during the project's life cycle. These changes are caused by the intrinsically dynamic (business) environment in which software systems are developed. In fact changing requirements are one of the major causes of software development projects failure [27, 23]. Mistakes made in the requirement elicitation phase are proved to be extremely costly [2] and cause the product's defect density to increase. Despite the difficulties, some companies seem to succeed in handling changing requirements effectively, while others fail. How can this be explained? What is it that makes certain organizations deal with changing requirements better than others? Because changes cannot be eliminated, effectively dealing with it seems the only viable strategy.

This paper presents empirical research that investigated the factors –in theory and practice– affecting the ability to cope with small changes in requirements. Case studies were performed at four completely different companies, located in The Netherlands. We report the results obtained by interviewing project managers of eight different projects from this heterogeneous sample of companies selected. Our findings include a list of factors that, according to the experts, play a significant role in practice to deal with small requirements change. Results suggest a central role of size as a factor determining the flexibility showed either by the organization or by the software development team.

Structure of the paper. Section 2 and Section 3 present factors related to changing requirements and flexibility respectively. Section 4 presents the research method used to execute this research step by step. Section 5 presents the results of the case study. Section 6 elaborates on future work and the paper concludes in Section 7.

2 Requirements Change

Since requirements change and requirements volatility have such a large impact on the success of software development projects a lot of research effort has been put into these topics over the last decade. A broad range of change-related aspects have been studied. The sources of change have been investigated in [12, 22, 17, 13], while a classification of changing (software) requirements have been reported on [3, 18, 22]. The impact of requirements change in general and on project performance and defect density in particular has also been thoroughly investigated by [19, 10, 29, 12, 15, 13]. Other fields of research are requirements traceability [3, 19], requirements management tools [14, 3, 4, 16], the assessment and prediction of change

[13] and means to measure requirements volatility [13, 7]. Moreover several Requirement Change Management Process Models have been developed as presented in [20].

Despite all the research effort on requirements change, there is still the need to improve characteristics of companies to better deal with requirements change. In particular, and to the best of our knowledge, there is empirical research needed in the domain of software development. The literature, however, is a valuable source of factors and aspects that contribute to the extent of requirement volatility. A summary of the factors that we have identified in the literature is listed below:

1. *Internal and external (with client and users) communication and relationships*. Inadequate and poor communication could be one of the reasons causing requirements to change [12, 29, 18, 16, 22].

2. *Means of communication*. Close, face-to-face communication makes that changes in requirements are communicated more clearly [6, 1, 22].

3. *Presence and influence of client and users*. Having the user and/or client present during the project makes communication and the detection of changes easy [26, 16, 22].

4. *Project/product size*. The size of the project (in terms of budget and manpower for example) and product (complexity and KLOC) is an important factor according to [29, 12, 16, 1].

5. *Organization size*. The size of the organization (in terms of number of employees and annual turnover) [12].

6. *Development methodology*. Some development methodologies inhibit flexibility while others advocate it [29].

7. *Outsourcing*. Whether (part of) the project is outsourced can influence the way change is managed and communicated [14].

8. *Project Management* (formalization of documents, tools present, etc.) [16, 14]

9. *Project team* Flexible, self-organizing, small teams with a flat hierarchy and experienced team members tend to cope well with requirements change [6].

3 Flexibility

Flexibility is quite an elusive and vague concept. When does one call something flexible? What properties does a flexible company have? Flexible with respect to what? The Oxford English Dictionary gives the following definition: ”Susceptibility of modification or alteration; capacity for ready adaptation to various purposes or conditions; freedom from stiffness or rigidity”, while [6] defined flexibility as ”the ability of an entity to proactively, reactively or inherently embrace change in a timely manner, through its internal components and its relationships with its environment”. The literature does not provide metrics to measure the flexibility of a company to deal with requirements change, but literature on Agile software development [11, 5, 25, 1, 21] issued some suggestions. Aspects taken in one way or another from literature that are related to flexibility were:

1. *Autonomy of team members*. Autonomous team members who can determine which tasks have to be done are more flexible than team members that are micromanaged [1, 21].

2. *Close customer partnerships and continuous user involvement*. A close relationship with customer and end-users makes it easy to adapt to change since changes are easily observed [11, 21, 5].

3. *Team proximity and amicability*. A team that is located in close proximity and gets on very well together is likely to be more flexible [11, 5].

4. *Length of feedback loop with customers and management*. A long feedback loop inhibits a team's agility [11].

5. *Emphasis on people factors*. According to [5] the most important implication to managers working in the agile manner is that it places more emphasis on people factors in the project: amicability, talent, skill, and communication.

6. *Intense interaction and communication between team members and, developers and users*. (Development) teams that work closely together and interact often tend to be very flexible [5, 11].

7. *Small, organic and dynamic teams*. Small teams that fit together naturally and allow people to join or leave whenever necessary are more flexible according to [1, 25].

These aspects largely cohere to a dominant idea in agile development, assuming that a team can be more effective in responding to change if it can reduce the cost of moving information between people, and reduce the elapsed time between making a decision to seeing the consequences of that decision.

4 Research Method

We use an exploratory case study [24] as basis of our research. First the scope of the research was determined by formulating a research question. Next a literature study was performed to learn about related research and the state-of-the-art in the field. Drawing from this knowledge a reasoning framework was developed, serving us to cover the collection of related properties discovered before by multiple investigators. The reasoning framework contained means to reason about the constructs that were investigated, such as flexibility, success and proximity. It formed the basis of the interview framework, which listed the questions that should be asked during the interviews in order to get the right data from which a sound conclusion could be drawn.

Drawing from established processes of investigation aimed at the discovery of facts [8, 28], the research was structured in the following way:

1. Define the research question
2. Perform a literature study
3. Develop an interview and a conceptual framework
4. Select cases
5. Perform the interviews
6. Evaluate and analyze the data
7. Draw conclusions and report them

The following subsections elaborate on these steps.

4.1 Research Question

It is important to determine the boundaries of the research to prevent an information overload and to be able to focus on what has to be achieved. Therefore a clear definition of the research question is mandatory. Having in mind that ability to deal successfully with small requirements change brings projects closer to success, the research question that has to be answered by this research is:

What is it that makes some projects deal with changing requirements better than others?

All case studies were performed with this question in mind. The expectations were that large companies would be less flexible and that less flexible organizations would be inferior with respect to the ability to cope with changing requirements.

4.2 Perform a literature study

A literature study was performed to learn about related research and the state-of-the-art in the field. The results of this step have been presented in Section 2 and Section 3.

ORGANIZATION SIZE
Number of employees
Number of countries
Annual turnover (in millions)
PROJECT
Product complexity
Number of levels in team hierarchy
Project's budget
Project duration
Outsourcing
Formalization of documents
Tools
Emphasis on people
Influence of opinion of the customer
TEAM ORGANIZATION
Number of team members
Autonomy
Project manager - Developers relationship
Team proximity (geographical location)
Team interaction and communication
Self organizing team
Hierarchy levels within the team
DEVELOPMENT METHODOLOGY
Length of feed back loop with customers
communication level with customers

Table 1. Some factors affecting the ability to cope with small changes.

4.3 Develop an interview and a conceptual framework

Drawing from the knowledge gathered from our survey of the state-of-the-art, we developed a list of key concepts that served as a framework to reason about flexibility and requirements change. The reasoning framework evolved during the process of interviewing representatives from industry until the list of key concepts reached a point where it became stable. In other words, the list of key elements comprised our reasoning framework when experts, recognizing the elements described in the framework, agreed with all of them.

Our reasoning framework organized the concepts into levels (*e.g.,* organization size, project, team organization and development methodology) and specified concrete factors related to that level. As a consequence, rather that referring simply as "Communication issues", our reasoning framework allowed us to quickly and consistently focus in a concrete aspect, *e.g.,* "Project manager - Developers relationship" (*i.e.,* relationship between project manager and developers (close vs distant) at the team organization level.

The reasoning framework was made up of the aspects identified in sections 2 and 3. Table 1 depicts our reasoning framework and the elements deemed relevant after reviewing the state-of-the-art and confronting with the use in practice as described by the experts.

Having at our service this kind of structure, we expected it to help us to reason about the elements that were investigated. We developed an interview protocol based on our reasoning framework. The interview protocol listed the questions that should be asked during the interviews in order to get the right data from which a sound conclusion could be drawn.

4.4 Select Cases

As pointed out by Eisenhardt [8] it is often desirable to study extreme cases rather than typical ones because the phenomena to be investigated are more evident. Or, as stated by Flyvbjerg [9], "A typical or extreme cases often reveal more information because they activate more actors and more basic mechanisms in the situation studied".

Therefore, four companies were selected based on the dimensions flexibility and size. Two of the selected cases were *expected* to be each others opposite. The other two were *expected* to be more typical. In total eight different projects were investigated. At some companies several people, involved in different projects, were interviewed to get a good overview of the situation at the company.

Table 2 shows the size-related aspects on both the company and project level. The company-level figures are an approximation of the actual numbers while the project level ratings are on a scale from 1 (very small/short/low) to 5 (very large/long/high).

4.5 Performing the Interviews

The reasoning framework made up of the aspects identified in sections 2 and 3 formed the bases of our interview protocol.

The structured interviews were all conducted on site and involved primarily project managers. Participants were guaranteed anonymity and all information that was published had been carefully sanitized so that no person, project or company could be identified. Most interviews were recorded with knowledge of the interviewee, but the interviewee was offered the possibility to (temporarily) turn off the recording device so they could speak freely.

4.6 Evaluate and analyze data

After all the interviews were performed and worked out the data was analyzed and evaluated by the authors. To make this analysis sound, the reasoning framework was applied again on each of the answers.

This is an exploratory case study and our objective was simply to gather knowledge. Rather than focusing on the influence of a variable over another, we summarize the data collected and put it in a format that is easy to digest. Table 3 summarizes the answers obtained in the interviews.

Again we recur to our reasoning framework, this time to read the results. Table 3 lists the factors together with a rate given by the authors. A scale from 1 to 5 summarizes the answers, indicating the relevance of the factors in each company. The value 1 is a pragmatic way of indicating that a flexible view is observed, while 5 indicates that the strict view applies in that company.

For instance, at the level of communication, one entry in Table 3 is "Project manager - Developers relationship is close vs distant". This reads as follows:

> In company A (2-3) the everyday contact between the project manager and developers is neither too close nor too distant.
>
> However, the value 4-5 in the next column shows that in company B the relationship between project manager and developers is distant.
>
> Moreover, in company C (the cell next to the right, containing value 1) the relationship between project manager and developers is perceived as very close.
>
> Finally, in company D, that relationship tends to be perceived as close.

4.7 Draw conclusions and report them

Projects are managed in different styles. Nevertheless, study of the answers revealed that companies were consistent in their view and management of the projects regarding their dealings with small requirements change: company B choses a rigid style and this shows in every factor recognized by our framework. Table 3 shows consistently high values in the column corresponding to company B. For example, company B choses to use "static documents" rather than letting documents change; and tends to impose the same standard for deliverables no matter the kind of project, rather than to let them vary according to the project needs.

Moreover, company B was consistent with this behavior in every level. The same structured choices are revealed at the levels of development processes, team organization, individual employee and communication level, whereas representatives of company A, company C and company D showed the opposite behavior – a more flexible one– also consistently.

Finally, interviewees from company B showed concern on the ability of the projects to deal with small requirements

	Company			
	A	B	C	D
Company level				
Number of employees	150	100,000+	35	4500
Number of countries	1	130+	2	1
Annual turnover (in millions)	?	300,000+	3.5	570
Project level				
Product complexity	2	3	3-4	4
Number of team members	1-2	4	2	2
Number of levels in team hierarchy	1-2	4	1	1
Project's budget	2	2-4	2	2-3
Project duration	1-2	2-3	N/A	2-4

Table 2. Size of the companies and projects studied in our exploratory case study.

	Company			
	A	B	C	D
Document level				
Documents are likely to change vs. static documents	1	3-4	1	2
Deliverables vary per project vs. deliverables are imposed	2	5	2	2
Development process level				
Adapted to fit the client vs. imposed upon the client	1	5	1	1
Agile development process vs. static (waterfall) process	3	4	3	2-3
Team organization level				
Team is located close to each other vs. team is scattered	1	5	1	2
Flat and informal hierarchy vs. formal and rigid hierarchy	2	4	1	2
Individual employee level				
Project manager has a lot of authority vs. has no authority	2	3	1	1
Project manager has a lot of autonomy vs. has no autonomy	2	3	1	1
Developers have a lot of authority vs. have no authority	2	4	1	2
Developers have a lot of autonomy vs. have no autonomy	1	4	1	1
Employees have to be multidisciplinary vs. do not have to be multidisciplinary	1	4	1-2	2
Communication level				
Company - Client relationship is close vs. distant	1-2	4	2	1-2
Company - Client communication is close (face-to-face) vs. distant (telephone, e-mail,)	2	3-4	3	2
Project manager - Developers relationship is close vs. distant	2-3	4-5	1	2
Project manager - Developers communication is close (face-to-face) vs. distant (telephone, e-mail,)	1	3-5	1	1
Developer - Developer relationship is close vs. distant	1	3	1	1-2
Developer - Developer communication is close (face-to-face) vs. distant (telephone, e-mail,)	1	4-5	1	1

Table 3. Summary of the results of the interviews performed in company A, company B, company C and company D. The scale from 1 to 5 shows the perceived status of the left factor (*e.g.,* relationship between project leader and manager, signifying 1 = close, 5 = distant). The values are assigned by the authors after performing the interviews and according to the answers obtained in them.

change. Rather, the interviewees of company A, company C and company D showed that they simply accepted small changes in requirements as part of life and trusted on the flexible way in which their projects were organized and on the creativity of the other employees to deal successfully with it.

5 Results

The reasoning framework proved to be of help to analyze the results of the interviews, by establishing a frame that guided us through a methodological analysis. Indeed, the structure of the framework made us to be precise in the interview and, more important, it helped us to manage the concepts (factors) in a consistent way through out the different steps of our method.

Validity of our results was checked via interviews with experts. They found that the factors showed in our reasoning framework were a suitable list of factors to analyze requirements changes under our focus on flexibility view. Experts suggested that more factors could be added to the list in our framework, and they found our findings coherent and realistic, and could relate to them.

6 Limitations and Future Work

The answers collected and our findings based on them do not reveal surprising practices, but confirm existing theory. However, this is only a first step toward understanding mechanisms used in practice to successfully deal with changing requirements. This suggests the need to perform case studies in more companies and to improve our reasoning framework toward building an ontology.

Measuring the successfulness of the projects that were investigated proved to be problematic. A number of projects investigated was not yet finished and one would not finish at all. Moreover, all of the interviewees of projects that were finished stated that the project in which they were involved was completed successful, but they could not provide any figures indicating how successful the project was. Most of the interviewees simply did not had the information or were not allowed to provide it.

7 Conclusion

Pursuing the long term goal of understanding the mechanisms used to successfully deal with changing requirements, our objective in this paper is to gather knowledge as to which are the characteristics of organizations dealing successfully with small requirements change.

In this paper, presenting an exploratory case study, we report our findings. We studied the state-of-the-art of requirement change and factors or characteristics supposed to influence successful dealing with small requirements change. In particular, we focused on those factors that are suspected to influence flexibility of the software development process, recognized by literature and recognized by experts in practice.

We constructed an initial reasoning framework including factors relevant to flexibility to cope with small requirements change. Moreover, we used our reasoning framework in the development of an interview protocol. Then we selected different cases of companies and projects for study. Eight projects from four different companies where analyzed via interviews with the project managers using our interview protocol. Finally, we analyzed the answers obtained, drew conclusions and reported them.

Results showed that companies were consistent in their view and management of the projects: one followed consistently a rigid view and stressed rigidity in the factors of every level, while three companies were more flexible in their dealings with the factors detected.

Moreover, the company having a more rigid style to manage the factors supposed to influence successful dealing with small requirements change showed concern on the ability of their projects to deal with small changes. Rather, the companies that revealed to have a more flexible style to manage those factors –the ones present in our reasoning framework– accepted small requirements change as part of life and trusted on the creativity of their employees and on the characteristic of their projects of being less strict, to successfully deal with such changes.

ACKNOWLEDGEMENTS

Thanks to Pascal van Eck for his valuable comments and advice on this work. We gratefully acknowledge the financial support of the Dutch Jacquard program for the project "QuadREAD".

References

[1] S. Augustine, B. Payne, F. Sencindiver, and S. Woodcock. Agile project management: steering from the edges. *Commun. ACM*, 48(12):85–89, 2005.

[2] B. W. Boehm. *Software Engineering Economics*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1981.

[3] J. Brier, L. Rapanotti, and J. G. Hall. Problem-based analysis of organisational change: a real-world example. In *IWAAPF '06: Proceedings of the 2006 international workshop on Advances and applications of problem frames*, pages 13–18, New York, NY, USA, 2006. ACM Press.

[4] J. Buckley, T. Mens, M. Zenger, A. Rashid, and G. Kniesel. Towards a taxonomy of software change. *Journal on Software Maintenance and Evolution: Research and Practice*, 17(5):309–332, September-October 2005.

[5] A. Cockburn and J. Highsmith. Agile software development: The people factor. *Computer*, 34(11):131–133, 2001.

[6] K. Conboy and B. Fitzgerald. Toward a conceptual framework of agile methods: a study of agility in different disciplines. In *WISER '04: Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research*, pages 37–44, New York, NY, USA, 2004. ACM Press.

[7] S. Datta and R. van Engelen. Effects of changing requirements: a tracking mechanism for the analysis workflow. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1739–1744, New York, NY, USA, 2006. ACM Press.

[8] K. M. Eisenhardt. Building theories from case study research. *Academy of Management Review*, 14(4):532–550, 1989.

[9] B. Flyvbjerg. Five misunderstandings about case-study research. *Qualitative Inquiry*, 12(2):219–245, April 2006.

[10] L. Goldin and A. Finkelstein. Abstraction-based requirements management. In *ROA '06: Proceedings of the 2006 international workshop on Role of abstraction in software engineering*, pages 3–10, New York, NY, USA, 2006. ACM Press.

[11] J. Highsmith and A. Cockburn. Agile software development: The business of innovation. *Computer*, 34(9):120–122, 2001.

[12] T. Javed, M. e Maqsood, and Q. S. Durrani. A study to investigate the impact of requirements instability on software defects. *SIGSOFT Softw. Eng. Notes*, 29(3):1–7, 2004.

[13] A. Loconsole and J. Borstler. An industrial case study on requirements volatility measures. In *APSEC '05: Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05)*, pages 249–256, Washington, DC, USA, 2005. IEEE Computer Society.

[14] M. Lormans, H. van Dijk, A. van Deursen, E. Nocker, and A. de Zeeuw. Managing evolving requirements in an outsourcing context: An industrial experience report. In *IWPSE '04: Proceedings of the Principles of Software Evolution, 7th International Workshop on (IWPSE'04)*, pages 149–158, Washington, DC, USA, 2004. IEEE Computer Society.

[15] Y. K. Malaiya and J. Denton. Requirements volatility and defect density. In *ISSRE '99: Proceedings of the 10th International Symposium on Software Reliability Engineering*, page 285, Washington, DC, USA, 1999. IEEE Computer Society.

[16] C. Mao, Y. Lu, and X. Wang. A study on the distribution and cost prediction of requirements changes in the software life-cycle. In *ISPW*, pages 136–150, 2005.

[17] N. Nurmuliani, D. Zowghi, and S. Fowell. Analysis of requirements volatility during software development life cycle. In *ASWEC '04: Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04)*, page 28, Washington, DC, USA, 2004. IEEE Computer Society.

[18] N. Nurmuliani, D. Zowghi, and S. P. Williams. Using card sorting technique to classify requirements change. In *RE '04: Proceedings of the Requirements Engineering Conference, 12th IEEE International (RE'04)*, pages 240–248, Washington, DC, USA, 2004. IEEE Computer Society.

[19] J. S. O'Neal. Analyzing the impact of changing requirements. In *ICSM '01: Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01)*, page 190, Washington, DC, USA, 2001. IEEE Computer Society.

[20] N. Ramzan, S. Ikram. Requirement change management process models: Activities, artifacts and roles. In *INMIC 06: Proceedings of The 10th IEEE International Multi-topic Conference*, pages 219–223, 2006.

[21] K. Reed, E. Damiani, G. Gianini, and A. Colombo. Agile management of uncertain requirements via generalizations: a case study. In *QUTE-SWAP '04: Proceedings of the 2004 workshop on Quantitative techniques for software agile process*, pages 40–45, New York, NY, USA, 2004. ACM Press.

[22] K. E. S. Harker and J. Dobson. The change and evolution of requirements as a challenge to the practice of software engineering. In *Proceedings of IEEE International Symposium Requirements Engineering*, San Diego, CA, USA, 1993.

[23] I. Sommerville. *Software Engineering*. Addison-Wesley, 6th edition, 2001.

[24] W. Tellis. Introduction to case study [68 paragraphs]. the qualitative report [on-line serial]. 3(2), 1997.

[25] W. H. M. Theunissen, A. Boake, and D. G. Kourie. In search of the sweet spot: agile open collaborative corporate software development. In *SAICSIT '05: Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 268–277, , Republic of South Africa, 2005. South African Institute for Computer Scientists and Information Technologists.

[26] J. E. Tomayko. Managing evolving requirements using extreme programming. In *Soft-Ware 2002: Proceedings of the First International Conference on Computing in an Imperfect World*, pages 315–331, London, UK, 2002. Springer-Verlag.

[27] A. van Lamsweerde. Requirements engineering in the year 00: a research perspective. In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 5–19, New York, NY, USA, 2000. ACM Press.

[28] R. K. Yin. *Case Study Research: Design and Methods (Applied Social Research Methods)*. Sage Publications, Inc, December 2002.

[29] D. Zowghi and N. Nurmuliani. A study of the impact of requirements volatility on software project performance. In *APSEC '02: Proceedings of the Ninth Asia-Pacific Software Engineering Conference*, page 3, Washington, DC, USA, 2002. IEEE Computer Society.