TUM

INSTITUT FÜR INFORMATIK

Criteria for the Evaluation of Requirements Engineering Methods in a Change Intensive Environment

Martin Fritzsche, Manfred Broy

TUM-I0829
August 08

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-08-I0829-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

Content

Criteria for the Evaluation of Requirements Engineering Methods in a Change Intensive Environment

Martin Fritzsche, Manfred Broy

Institut für Informatik, Technische Universität München, Boltzmannstr. 3,
85748 Garching b. München, Germany
fritzscm@in.tum.de, broy@in.tum.de

Abstract. Unstable requirements are one of the main threats to project success. Dealing with change is therefore an important capability that requirements engineering (RE) methods should have. There are changes that can be prevented by an appropriate method. On the other hand there are always also changes that cannot be prevented. A RE method should provide suitable means to deal with them in a way that minimizes the threat to project success. It is therefore interesting to analyze, how suited existing RE methods are in the context of unstable requirements. In this paper we want to lay the foundation for such an analysis by identifying the critical issues that RE methods have to address in this context. We identify the causes for requirements changes and determine which of these causes can be counteracted. Furthermore we present difficulties arising from changing requirements that have to be overcome. We also discuss the general objectives in RE that a RE method has to achieve.

Keywords: Unstable requirements, project situation assessment for RE process, typical project situations requiring specific methods.

1 Introduction

Unstable requirements increase the difficulty in software development significantly. In several domains, for instance internet and web-based applications, there is a very high rate of requirements changes which can make it almost impossible for software developers to cope with them [3]. A project's ability to deal with changing requirements can prove to be decisive for its success. Several development methods, like for instance agile methods, focus on the problem of dealing with requirements instability. These methods provide guidelines for the development process as a whole. Whether they are able to successfully address RE issues remains unclear. On the other hand there are many methods that focus on RE while not being specifically designed for a change intensive environment. They range from techniques like use cases, scenarios, interviews, requirements workshops, etc. to full grown methods and models covering many aspects of RE [7][14].

Our long term research goal is to analyze how good these methods can cope with requirements related issues in a change intensive environment and to propose our own RE method, based on the findings. In this paper we take a first step towards that goal by identifying all aspects that a RE method has to consider in a change intensive environment.

This paper provides an extensive list of aspects that RE methods have to consider. The list is split up into the following categories:

- Which general aspects of RE have to be covered?
- Which causes for requirements instability have to be considered?
- Which difficulties due to changing requirements have to be dealt with?

These lists can be used to analyze whether a specific method is suited for RE in a change intensive environment or not.

In the following section we give a brief overview over relevant related work. Then we deduce the general objectives of RE, i.e. aspects that always have to be considered by a RE method. In section 3 we identify the causes for requirements instability. Section 4 discusses the problems that are caused by requirements changes. In the final section we give a summary of our contribution.

2 Objectives of Requirements Engineering

In this chapter we will discuss the objectives of RE, i.e. the general aspects which RE methods always have to consider, independent of unstable requirements.

Looking for statements on what the objectives of RE are one finds many vague statements like for instance “to discover what is desired” [6]. While this is definitely an objective of RE, such abstract statements are not very useful for our analysis. There exists no detailed discussion of all the objectives of RE. However, there exists a lot of literature on the topic of which activities RE comprises, how they are best carried out, which documents should be created and what their contents should be [5][6][8][10][12][13][14][16][22]. While these guidelines focus on what to do, the question why the proposed activities have to be carried out is not explicitly answered. For analyzing the whole spectrum of existing methods we need to define what a RE method generally has to achieve instead of proposing what should be done.

The general activities which are proposed are usually elicitation, analysis, specification, validation and management. Wiegers [22] for instance proposes for requirements analysis the following sub steps: draw a context diagram, create prototypes, analyze feasibility, prioritize requirements, develop a requirements model, etc. While these activities offer the right amount of detail they are only suggestions. Not every RE method employs prototypes. Wiegers does not describe objectives pursued by RE but means to achieve them. The main objective behind prototyping is to understand the requirements and to check whether the results of the development match the needs of the customers.

The problem can be further understood when looking at eXtreme Programming. The method has several shortcomings concerning requirements specification. Requirements are documented in the form of story cards and acceptance tests. Details of the requirements have to be clarified in discussions with the customer, which are not documented. In this way direct communication remedies the problems arising from the lack of documentation. Using Sommerville’s [16] or Wieger’s [22] model in our analysis we would not consider this alternative strategy. Our approach however is based on the objectives behind the proposed activities. In this case we will check whether and how the method manages to communicate the requirements to all project participants. This is the main objective behind specifying requirements, however not the only one.

A typical RE activity may be an objective as well. Just finding a higher level objective pursued by an activity does not mean that it does not pose an objective itself. There is a hierarchy of objectives where each element is a means to achieve its superordinate objective. The criterion by which we discern whether an activity is an objective is, whether it describes a general aspect, all RE methods have to consider.

2.1 The Traditional Approach in Requirements Engineering

Here we want to present an overview of the typical high level RE activities, which can be used as a starting point for the deduction of the objectives of RE. We compiled a list of activities suggested in the relevant literature [5][6][8][10][12][13][14][16][22]. Detailed descriptions of the activities can be found there. The list should not be regarded as a standard approach to RE. In fact there exists no standard approach. Most applied approaches do not carry out all of the listed activities. However since it will be our basis for determining all of the objectives of RE we tried to compile a very extensive list of activities.

Develop requirements			
	Elicit requirements	Gather information	
		Elicit goals	Elicit business goals
			Elicit customer goals
		Develop system vision	
		Define system scope	
		Identify general conditions	
		Gather requirements	Gather business requirements
			Gather customer requirements
		Refine requirements	
		Identify rationale for requirements	
		Discover the real requirements	
	Analyze requirements	Understand requirements	
		Identify dependencies between requirements	Structure requirements
		Understand the requirements' impact on business processes	
		Determine return on investment	
		Define priorities	
		Elicit risks	
		Assure completeness	
		Assure necessity	
		Assure correctness	
		Assure unambiguousness	
		Assure feasibility	Assure consistency
		Assure verifiability	
		Decide which requirements will be realized	
	Specify requirements	Provide a specification, which can be used in the design and the implementation	
		Make the requirements comprehensive	
		Assure completeness	
		Assure modifiability	

		Assure traceability
		Assure consistency
	Validate requirements	Check requirements documents
		Test requirements
		Define acceptance criteria
	Manage requirements	Define process for changing requirements
		Analyze impact of changing requirements
		Control changes of the requirements specification
		Monitor status of requirements
		Track requirements
		Manage risks associated with requirements
		Manage efforts associated with requirements
Integrate all relevant stakeholders		
	Consider all interests	
	Identify user classes	
	Communicate with stakeholders	
	Facilitate communication between stakeholders and developers	
	Dispel false expectations	Communicate realization constraints
		Communicate limits of the requirements
	Communicate risks	
Use engineering approach		
	Use a planned, structured approach	
	Ensure comprehensibility and repeatability of the process	
	Provide notations and methods	

Table 1: Typical high level RE activities

2.2 Deduction of the Objectives of Requirements Engineering

To every previously identified activity we will now identify the primary objectives pursued with it. Activities can have multiple objectives. For instance the primary aim of requirements specification is to communicate the requirements. A secondary effect is that the understanding of the requirements is increased. We limit our analysis on the primary objectives behind the activities. In our conception secondary effects of individual activities are already covered by primary objectives of other activities. We will order the discovered objectives in a hierarchy. The derivation of the RE objectives is based on our own considerations and our discussions with industrial partners.

First we will analyze activities associated with requirements elicitation. Gathering information is a very general activity, which can hardly be associated with a single objective. Depending on the type of information gathered, different objectives are pursued. In the context of eliciting requirements two specific objectives can be identified, that is to discover the stakeholders' needs and to gain a broad understanding of the domain, the organization and the business processes. The former is also pursued by the elicitation of business and customer goals. The Development of a system vision also comprises several aspects. The primary aim is to define central goals and communicate them to all relevant stakeholders. The communication aspect includes multiple objectives which we will identify in the context of requirements specification. With the definition of the system's scope it is possible to identify the system's interfaces to users and other systems. The respective general objective is to discover the goals which are pursued by developing the system. The focus of this activity is to make clear which goals can be covered by the system and which goals are beyond the scope of the system and therefore have to be reached by other means. We will include the definition of the system scope as a separate objective. Another typical RE activity is to identify general conditions. Hereby the objective of gaining a broad understanding of the domain, the organization and the business processes is pursued. The activity of gathering requirements pursues the already identified general objective of discovering the stakeholders' needs, but we will include this activity as a more detailed objective: define requirements which meet the stakeholders' needs. By refining requirements one follows the same objective as with gathering requirements since in our understanding the term "requirement" encompasses requirements of all levels of detail. Identifying the rationale for requirements follows also the objective to discover the stakeholders' needs but we will include it also as a separate objective. With the activity of discovering the real requirements one pursues the objective to define requirements which meet the stakeholders' needs. It is important to notice that requirements are only included if they serve the stakeholders' needs. In addition we mean the real needs of the stakeholders and not those they may misleadingly think they have. We use the word "discover" in the objective "discover the stakeholders' needs" to make clear, that the stakeholders' needs are not clear ad initio. It requires a certain amount of effort to determine them.

Now we will discuss activities associated with requirements analysis. Understanding requirements serves besides the already identified objectives to identify the rationale for requirements and to discover the stakeholders' needs also the objective to understand the system's impact on business processes. Identifying dependencies between requirements is necessary to discover and resolve conflicts between requirements. Understanding the requirements' impact on business processes can be generalized to the objective to understand the system's impact on business processes. Determining the return on investment is necessary to assure the system's profitability. To do this the profit of the system and of individual requirements has to be determined. The cost aspect also has to be considered later by checking the budget throughout the project. The definition of priorities is important for resolving conflicts between requirements. Requirements of lesser priority are omitted in favour of high priority requirements. We have to extend the previously defined objective: Resolve conflicts between requirements favouring high-priority requirements. In addition, priorities are important for deciding which requirements should be realized. Often not all goals can be achieved because of limited time and budget. Therefore it has to be determined which requirements take precedence: achieve the most important goals which are pursued by developing the system, understand which requirements cannot be realized and select the necessary requirements to achieve the most important goals. Apart from the mere eliciting of risks an important objective is to provide preventive actions. Therefore another objective is to identify development risks and provide preventive actions and contingency plans. Assuring completeness means that all stakeholders' needs have to be discovered. Assuring necessity implies that only requirements necessary to meet the stakeholders' needs should be defined. Assuring correctness and unambiguousness, i.e. defining the requirements correctly is a separate objective. We will also include assuring feasibility as a separate objective: consider only viable requirements. Assuring verifiability means that it is necessary to check whether the realization matches the requirements: assure compliance with requirements. To decide which requirements will be realized is a separate objective: select the necessary requirements to achieve the most important goals.

Now we will analyze activities associated with requirements specification. Providing a specification, which can be used in the design and the implementation, has the objective of communicating the requirements to the developers. Usually this happens through documents. A super ordinate objective is to realize the necessary requirements. The developers have to understand the requirements correctly. Therefore they have to be communicated in a correct and comprehensible way. This is covered by the following objectives: assure that the requirements are comprehensible for the developers and define the requirements correctly. Assuring completeness means that all of the stakeholders' needs have to be discovered and that all requirements necessary to meet these needs have to be defined. Requirements can change. Therefore they have to be modifiable. On the one hand changes can be introduced because of wrong requirements which are not suited to cover the stakeholders' needs. On the other hand they may happen because of changing needs of stakeholders. The associated objectives are to define the requirements necessary to meet the stakeholders' needs and to consider changes to the stakeholders' needs. By assuring traceability one wants to assure that the stakeholders' needs are covered by the requirements and that the realization complies with the requirements. The other way round the rationale for each requirement should be established. Another typical RE activity is to assure consistency. The associated objective is again to define the requirements necessary to meet the stakeholders' needs. Apart from that, consistency in the specification poses a quality feature. The associated objective is to assure quality in the process.

Next we will analyze activities associated with requirements validation. By checking requirements documents one tries to assure that requirements are correctly defined. Testing requirements has the purpose to improve the comprehension of the defined requirements. The objective is to discover the real needs of the stakeholders and to test whether they are completely covered by the defined requirements. Acceptance criteria serve the purpose to check whether the stakeholders' needs are sufficiently satisfied by the realization. The associated objective is to assure compliance with requirements.

Now we will analyze requirements management activities. The activity to define a process for changing requirements has the same objectives as the modifiability of the specification: define requirements necessary to meet the stakeholders' needs and consider changes to the stakeholders' needs. One wants to analyze the impact of changing requirements in order to know whether changes to requirements are feasible.

In addition one wants to check whether changes have sufficient priority to justify their costs and the possible discarding of conflicting requirements. The associated objectives are to determine which requirements take precedence in the development and to understand which requirements cannot be realized. Changes to the requirements specification should be controlled. Thereby one hopes to communicate changes to requirements to the developers. Monitoring of the status of requirements and tracking of requirements are carried out to assure compliance of the system with the requirements. Managing risks associated with requirements comprises several objectives. Development risks have to be identified and communicated to the stakeholders. Furthermore preventive actions and contingency plans should be provided. Another typical RE activity is to manage efforts associated with requirements. It is important to assure, that the requirements can be realized within budget and schedule. The associated objectives are to check the budget and to check the schedule.

The next topic of our analysis is the integration of relevant stakeholders. One tries to consider all interests and identify user classes in order to discover all needs of all stakeholders. The objectives pursued by communicating with the stakeholders are twofold. On the one hand communication with the stakeholders is necessary to elicit their needs. On the other hand it is important to give them feedback from development. They have to be informed about risks and the feasibility of requirements. As an important side effect the stakeholders' understanding of their requirements is improved. Often stakeholders have false perceptions of the feasibility of their requirements or do not know what their real needs are. By dispelling these false perceptions one tries to communicate the feasibility of the requirements to the stakeholders and to discover the real needs of the stakeholders. The activity of communicating risks to stakeholders is a separate objective that has already been identified above.

Finally we will analyze what objectives are associated with applying an engineering approach. With the use of a planned, structured approach one tries to assure quality in the process. The objective of focusing on process quality is to enhance the quality of the product. The same objective is pursued with the activities of ensuring comprehensibility and repeatability of the process and of providing notations and methods.

By integrating all the identified objectives we arrive at the following hierarchy of objectives.

<table>
<tr><td colspan="7"><b>Discover the goals which are pursued by developing the system</b></td></tr>
<tr><td rowspan="7"></td><td colspan="3" rowspan="3">Discover all stakeholders’ needs</td><td colspan="3">Define requirements necessary to meet the stakeholders’ needs</td></tr>
<tr><td colspan="3">Identify rationale for requirements</td></tr>
<tr><td colspan="3">Consider changes to the stakeholders’ needs</td></tr>
<tr><td colspan="6">Gain a broad understanding of the domain, the organization and the business processes</td></tr>
<tr><td colspan="6">Understand the system’s impact on business processes</td></tr>
<tr><td colspan="3">Assure the system’s profitability</td><td colspan="3">Determine return on investment of the system and individual requirements</td></tr>
<tr><td colspan="6">Define system scope</td></tr>
<tr><td colspan="7"></td></tr>
<tr><td colspan="7"><b>Achieve the most important goals which are pursued by developing the system</b></td></tr>
<tr><td rowspan="9"></td><td colspan="6">Select the necessary requirements to achieve the most important goals</td></tr>
<tr><td rowspan="6">Realize the necessary requirements</td><td rowspan="4">Consider only viable requirements</td><td colspan="2" rowspan="3">Understand which requirements cannot be realized</td><td>Resolve conflicts between requirements favouring high-priority requirements</td><td>Determine which requirements take precedence in the development</td></tr>
<tr><td colspan="2">Check budget</td></tr>
<tr><td colspan="2">Check schedule</td></tr>
<tr><td colspan="4">Communicate the feasibility of the requirements to the stakeholders</td></tr>
<tr><td rowspan="2">Communicate requirements to the developers</td><td colspan="4">Assure that the requirements are comprehensible for the developers</td></tr>
<tr><td colspan="4">Define the requirements correctly</td></tr>
<tr><td rowspan="2">Assure compliance with requirements</td><td colspan="5">Identify development risks and provide preventive actions and contingency plans</td></tr>
<tr><td colspan="5">Communicate possible problems and risks in the development to the stakeholders</td></tr>
<tr><td colspan="7"></td></tr>
<tr><td colspan="7"><b>Enhance the quality of the product</b></td></tr>
<tr><td></td><td colspan="6">Assure quality in the process</td></tr>
</table>

Table 2: Objectives of RE

3 Causes for Requirements Instability

In this chapter we will identify and classify the causes for requirements changes.

The Chaos Report [17] has shown that unstable requirements are a widespread problem. How projects deal with this problem is a critical success factor. However, there is no work in which all aspects of changing requirements are thoroughly discussed. Though there exist a lot of works discussing single aspects, for instance by Ambler [1], Davis [5], Jones [9], Larman [11], Strens & Sugden [18] and Wan-Kadir & Loucopoulos [21]. Obtaining empirical data on the problem of changing requirements is difficult since the causes for unstable requirements differ greatly depending on the domain, the organization and the project. We want to contribute to the solution of the problem qualitatively by providing an extensive categorized list of probable causes for requirements instability. The statements in this chapter are based on general knowledge acquired from the relevant literature and our discussions with industrial partners.

As a general categorization we differentiate between factors that can be influenced, i.e. which can be mediated by an appropriate approach to RE and those that cannot.

First we will discuss causes for requirements changes that can be influenced.

Often some requirements are not completely understood in the early phases of the project. This can mean that certain requirements are not known or overlooked but also that stakeholders may not be aware of their real needs or do not really comprehend the requirements they defined and their ramifications. During development they gain a better understanding. This causes changes to requirements. Jones [9] states, that the primary cause for requirements changes is, that the domain where the system under development will be deployed is not completely understood. During the course of the project this vagueness is reduced. Therefore Jones regards changing requirements as a "technical necessity" in many projects. Furthermore comprehension problems can arise due to communication issues. It is possible that requirements get defined incorrectly or ambiguous so that other stakeholders and developers misinterpret them. Strens and Sugden [18] state that specifications that are too vague and lack detail cause misunderstandings and therefore additional requirements changes.

Incorrect estimates of the costs of requirements at the beginning of the project are another cause for requirements changes. In most projects it is very difficult to make precise estimates. During development it can become clear, that within the budget and the schedule more or less requirements can be realized than previously assumed.

Requirements can also change because priorities change. For instance some requirements could have been omitted because of their low priority or adjusted because they were in conflict with more important requirements. If they later get a higher priority they may be reintroduced or the changes to them may get reverted. The reason for changing priorities usually is that they were ill-defined at the beginning of the project. It can also be the case that the priorities change because of external reasons that cannot be influenced, e.g. due to organizational changes or changes in the environment of the system.

Problems during development are another reason for changing requirements. It may be the case that requirements cannot be implemented. Either they are not realizable at all or just not within the given schedule and budget. The reason for these changes usually lies in the misjudgment of the requirements' feasibility.

Conflicts between requirements that are discovered too late are another cause for requirement changes. Conflicts have to be resolved. This is possible by discarding single requirements favoring others or by finding compromises. If this is not done in the early phases of the project, changes have to be made later.

Changes may also occur when new stakeholders are introduced into the project. This happens when one failed to identify and assemble all stakeholders at the beginning of the project.

Now we will look into factors that cannot be influenced.

Sometimes business processes change [21]. They can change because of the introduction of the system. If these changes were not anticipated and the identified requirements do not account for the changed processes then this results in additional requirements changes. Sometimes anticipating these changes can prove to be very difficult. It is advisable to try to anticipate changes which can be detected without too much effort. Business processes can also change independently from the introduction of the system because of external factors which cannot be anticipated. Furthermore interfaces to other systems can change. This factor is dependent on the frequency of changes in neighboring systems.

Apart from changes in the immediate environment of the systems also external factors have to be considered. The introduction of new technologies in the market is one of the root causes for changes. In certain domains they can be very frequent. They result in additional functionality or offer different possibilities to solve the problem.

Because of economical changes or new competitors the business strategy and priorities of the organisation which wants to employ the system can change. This can lead to requirements changes. This is only a minor factor since this situation occurs only seldom. Another organizational reason for changes to requirements is when the schedule or the budget gets adjusted. The schedule may get shortened because of market pressure. The budget on the other hand may get increased in order to be able to realize more requirements.

Furthermore changes in laws, regulations and standards concerning the system can cause requirements changes.

We have identified the following causes for requirements instability:

Factors that can be influenced		
	Factors concerning the understanding of the requirements	Requirements were not understood in the early phases of the project
		Stakeholders develop a better understanding of the desired system in the course of the project
		Requirements were misinterpreted due to communication problems
	Estimates about the costs of requirements were wrong	Budget does not match the number of requirements that shall be realized
		Schedule does not match the number of requirements that shall be realized
	Priorities of requirements change	
	Feasibility of requirements was misjudged	Requirements cannot be realized due to technical problems
		Requirements cannot be realized within the given schedule and budget
	Conflicts between requirements are found	
	Not all relevant stakeholders were integrated	
Factors that cannot be influenced		
	Changes of business processes in the immediate environment of the system	Business processes in the system environment change
		Interfaces to neighboring systems change
	Introduction of new technologies	New technologies which allow a different solution of the problem
		New technologies which shall be integrated into the product
	Factors concerning the organization	Business strategy or priorities change
		Budget changes
		Schedule changes
	Laws, regulations or standards change	

Table 3: Causes for requirements instability

It is advantageous for a RE method to consider the difference between changes that can and those that cannot be influenced by the approach used in the project. Factors that can be mitigated are especially important for the analysis of RE methods.

The usual approach to minimize requirements changes is to perform a long and intensive RE phase at the start of the project. This, however, often conflicts with time constraints. Many requirements changes are caused by a lack of effort put into RE because of time constraints. But time pressure can not always be avoided. Therefore strategies have to be found which accommodate for requirements changes and time pressure at the same time.

4 Difficulties with Changing Requirements

On the topic of problems caused by changing requirements some papers have to be mentioned. Tomayko [20] and Thayer & Dorfman [19] focus on the difficulty of cost estimation in projects that are exposed to requirements instability. Boehm [2] and Brooks [3] provide valuable insights into the costs associated with rework. In our paper we do not want to restrict ourselves to single aspects but present an extensive list of all difficulties that arise because of changing requirements. The difficulties shall be presented in form of problems which have to be solved and resulting activities that have to be carried out.

Changing requirements lead to additional effort. First one has to decide whether changes should be accepted or rejected. The people authorized to make these decisions have to be assembled, usually in the so called change control board. To consider the whole decision making competence, all stakeholders would have to be assembled. It could be enough to congregate only the stakeholders whose interests are affected by the requirements changes. Often though only the stakeholders themselves know which changes affect them. In addition single project participants could gain advantages by the absence of others and therefore exclude them even if their interests are touched by the changes. If the effort for assembling all stakeholders is to be avoided, mechanisms have to be found to identify affected stakeholders and enable them to take influence on the decision. If a change gets accepted, affected artefacts have to be adapted. Besides the requirements specification also the design, the code, the tests and other artefacts may be affected depending on how advanced the implementation of the respective requirement is. The first step hereby is to identify affected artefacts. It is helpful when requirements can be traced to all documents which are dependent upon them. It has to be considered whether the benefits justify the effort to provide forward traceability. Depending on the change, artefacts may have to be modified, discarded or added. In order to cut the cost of changes, artefacts should be designed in a way that makes them easily modifiable and extendable. Since this also creates additional effort it has to be determined whether this is justifiable. Modifying software to include new requirements is often complicated and error-prone [3]. Since requirements may be changed or even discarded during the course of the project, the effort to analyze and realize them can be wasted. Therefore it is not always sensible to put much effort in detailed analysis and documentation at the beginning of the project. It has to be checked whether this effort can be justified despite the risk of changes. Since new requirements mean additional effort, there is a reluctance to include them [15]. Rejecting requirements for this reason can mean a decrease in the system's quality.

Another problem with changing requirements is that it is impossible to detect all conflicts between requirements in the early phases of the project, since not all requirements are known. Conflict analysis has to be carried out again when requirements change. Requirements of lesser priority conflicting with new requirements may be discarded. In this case the already spent effort on them is wasted. There is a reluctance to include new requirements if they conflict with existing requirements. This can lead to a decrease in the system’s quality.

Thoughts about requirements during the early phases of the projects can later be forgotten. This is especially problematic when a conflict analysis with new requirements has to be carried out or if the priorities of requirements should be revaluated. The situation can be complicated if relevant stakeholders are not available anymore. Especially important thoughts are the rationale for the requirements’ priorities or for the discarding of requirements. The problem can be mediated by trying to document the thoughts. Although in reality it is difficult to document them completely. Furthermore this means additional effort, which has to be justified.

If requirements change, the project plan usually has to be modified. Cost and risk estimates have to be updated. Contents of releases or iterations may change. Schedule and budget have to be checked.

The project costs pose another problem. Since not all requirements are known in the early phases of the project, it is unclear how much the system will cost [20]. Changes may not be realizable within the budget. Therefore the budget has to be extended or requirements have to be discarded. Requirements found late in the project may be especially expensive. Boehm [2] estimates, that changes to requirements can cost 200 times as much if they are made late in the project rather than at the beginning.

We have identified the following problematic aspects of changing requirements:

Additional effort is generated		
	Stakeholders have to be assembled again in order to decide about the acceptance of changes	
	Artefacts have to be adjusted	Artefacts that are to be adjusted have to be identified
		Artefacts have to be modified, discarded or added
		Artefacts should be extendable und modifiable
		Adjustments are often complicated and error-prone
	Initial effort may be wasted when requirements change	
	Requirements changes may get denied because people shy the additional effort	
Not all conflicts can be detected at project beginning		
	Conflict analysis has to be carried out multiple times during the course of the project	
	Development results get discarded because they are in conflict with new requirements	
	Important requirements get denied because they are in conflict with old requirements	
Not documented thoughts about requirements are no longer present but needed when requirements change		
Project plan has to be adjusted		
	Estimates have to be updated	
	Content of releases and iterations can change	
	Schedule can change	
	Budget can change	
Problems with costs arise		
	System costs are not known at project beginning	
	Changes possibly cannot be realized within the budget	
	Late changes can be particularly costly	

Table 4: Difficulties with changing requirements

5 Conclusions

In this paper we extensively identified the issues that a RE method has to address in change intensive projects. Aspects that have to be considered are general objectives pursued by RE, causes of requirements volatility and difficulties that arise due to changing requirements.

Our catalogue of requirements can serve as a basis for analyzing a RE method's suitability for a change intensive environment. For each of the objectives it has to be analyzed in how far the method is able to achieve it. Concerning the causes of requirements it has to be discussed how a method thwarts them. However, not all causes can be influenced. In addition a balance between the benefits from counteracting changes and the associated effort has to be found. In this respect also certain objectives of RE have to be considered. For instance, changes can occur because the real needs of the stakeholders are discovered. Preventing these changes means discovering the real needs in the early phases of the project - which is often impossible or at least very difficult - and not ignoring a change. Finally it has to be analyzed whether the method considers and mitigates difficulties arising from changing requirements.

Our catalogue of requirements related issues in change intensive projects remains to be evaluated. Existing methods have to be analyzed using our approach. With the feedback gained from the analysis our lists will have to be refined. However, this paper provides a starting point for understanding the problems that methods have to address and can serve as a basis for further discussion.

The next step in our research effort will be to apply our approach by analyzing methods which promise to be advantageous in the context of unstable requirements.

References

1. Ambler, S.W: Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. Wiley, New York (2002)
2. Boehm, B.W.: Software Engineering Economics. Prentice-Hall, Englewood Cliffs (1981)
3. Boehm, B.W.: Requirements that Handle IKIWISI, COTS, and Rapid Change. IEEE Computer 33(7), 99-102 (2000)
4. Brooks, F.P.: The Mythical Man Month. Addison Wesley, Boston (1995)
5. Davis, A.M.: Software Requirements. Prentice-Hall, Englewood cliffs (1990)
6. Gause, D.C., Weinberg, G.M.: Exploring Requirements: Quality before Design. Dorset House, New York (1989)
7. Geisberger, E., Broy, M., Berenbach, B., Kazmeier, j., et al.: Requirements Engineering Reference Model (REM). Technical report, Technische Universität München (2006)
8. Jackson, M.: Software Requirements & Specifications. ACM Press, Wokingham (1995)
9. Jones, C.: Strategies for Managing Requirements Creep. IEEE Computer 29(6), 92-94 (1996)
10. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. Wiley, Chichester (1998)
11. Larman, C.: Agile & Iterative Development. Addison Wesley, Boston (2004)
12. Macaulay, L.A.: Requirements Engineering. Springer, London (1996)
13. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: Proceedings of the Conference on the Future of Software Engineering, 35-46. ACM Press, New York (2000)
14. Robertson, S., Robertson, J.: Mastering the Requirements Process. ACM, Harlow (1999)
15. Schwaber, K.: Agile Processes: Emergence of Essential Systems. The Agile Alliance, http://www.agilealliance.org/articles (2002)
16. Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. Wiley, Chichester (1997)
17. Standish Group: The Chaos Report, http://www.standishgroup.com (1995)
18. Strens, M. R., Sugden, R. C.: Criteria for the assessment of representation methods as vehicles for handling change. In: Proceedings of the International Symposium and Workshop on Systems Engineering of Computer Based Systems, 233-239. (1995)
19. Thayer, R., Dorfman, M.: Software Requirements Engineering. IEEE Computer Society Press, Los Alamitos (1997)
20. Tomayko, J.E.: Engineering of Unstable Requirements Using Agile Methods. In: International Workshop on Time-Constrained Requirements Engineering (TCRE'02). Essen (2002)
21. Wan-Kadir, W.M.N., Loucopoulos, P.: Relating Evolving Business Rules to Software Design. Jorunal of Systems Architecture 50, 367-382 (2004)
22. Wiegers, K.E.: Software Requirements. Microsoft Press, Redmond (2005)