Empirical Software Engineering, 2, 143–207 (1997)

Qualitative Analysis of a Requirements Change Process[1]

KHALED EL EMAM elemam@iese.fhg.de
Fraunhofer—Institute for Experimental Software Engineering, Germany

DIRK HOELTJE
Positron Inc., Canada

1. Introduction

The implementation of good software requirements management practices is believed to be one of the first process improvement steps that an organization should take. This is clear, for example, in the implied staging of practices in the CMM for software where requirements management is a level 2 Key Process Area (KPA) (Software Engineering Institute, 1995). One analysis of data available at the SEI found that 61% of assessments had findings that map to the requirements management KPA (Kitson and Masters, 1993). This indicates that there is an industrial priority to improve software requirements management practices.

In the related discipline of systems engineering, good systems requirements management (SRM) practices are believed to be some of the first that should be addressed to increase systems engineering maturity. For example, in one systems engineering maturity model generalized from the CMM for software, the system requirements management KPA is at level 2 (Brill and Brammer, 1993). In another systems engineering maturity model, one of the maturity dimensions stipulates that the implementation of systems engineering in the early life cycle phases is one of the first signs of maturity (Mar, 1993).

The implementation of system requirements management remains a challenging problem. This is evidenced by the number of implementation barriers that have been mentioned in the literature, such as (Edwards and White, 1994; Simon, 1994): achieving buy-in, maintaining process fidelity, performing appropriate automated tool selection, achieving tool acceptance, use of natural language in requirements documents that result in imprecise requirements, different systems engineering disciplines using different terminologies, and organizational politics.

In this paper we present a characterization method that can be applied during the implementation of a requirements change process for large systems. In general, the method is useful in identifying problems while implementing new change processes. We apply the method to characterize a process that maintains the requirements for a large real-time system.

In the next section we describe the organization were this study was conducted. We then outline the characterization method that we followed in section 3. In section 4 we present the results. We conclude the paper in section 5 with an overall summary and directions for future research.

2. Study Context

In 1989, a government organization (henceforth Organization X) started a systems engineering project. During this project, a Technical Baseline for system requirements (which is essentially a set of requirements specification documents) of a large real-time system was developed. After these documents were baselined in January 1995, a change process was started. This process is intended to allow for the orderly evolution of the Technical Baseline, and is performed by the Systems Engineering Organization (SEO).

The Technical Baseline specifies a Canadian national system that operates in geographically distributed locations. Some parts of the system are already operational, while others are earmarked for development within the next decade. The program of system development and procurement to make the complete system operational is documented in a Transition Plan, which is also part of the Technical Baseline. The Transition Plan is intended to be used by senior management for prioritizing subsystem development and procurement, allocating human resources and budgets, and monitoring the development efforts. The overall objectives of the system requirements are to: enable traceability of individual subsystems to requirements, ensure that subsystem projects adhere to a general vision of where the overall system is going, avoid duplication across projects, enable configuration management across different subsystem development/procurement projects, manage interdependencies amongst development/procurement, and define interfaces between subsystems.

Organization X has a central headquarters and a number of largely autonomous regional centres that operate, maintain, as well as develop and procure systems. Within Organization X there are also a number of units that represent various software and system disciplines whose knowledge is necessary to specify, develop, operate, and maintain the national system.

The objective of the systems engineering effort is to provide an overall specification and management of the national system. It is hoped that this will result in cost savings. In the past, the autonomy of the regions has resulted in regional systems being developed that duplicate functionality across regions, and in subsystems across regions that do not interface well because the scope of their development was limited to the specific regions. At a global level, such an approach is inefficient. Furthermore, regional projects do not always document their designs and implementations. This leads to difficulties in making changes, as well as difficulties in duplicating these subsystems in other regions or scaling them to a national level.

Organization X is facing severe budget cuts. This created pressure to take some process improvement actions within the SEO. These actions are intended to streamline the requirements change process and enhance customer value.

The objective of our study was to identify opportunities for improving the SRM process. Following some initial discussions with members of the systems engineering organization, it became apparent that the most important issue at this point in time was the implementation of the requirements change process.

Table 1. A summary of the steps of the characterization method.

	Method for Characterizing a Change Process
1.	Make a list of organizational, product, and people factors that may potentially have an impact on the process achieving its objectives.
2.	Identify/develop/formalize a prescriptive process model that reflects the engineers' concepts of an ideal process that is to be implemented.
3.	Select one or several past change proposals for analysis.
4.	Trace the handling of each change proposal through the process (i.e., what activities were performed and what artifacts were produced).
5.	For each change proposal, perform a case-oriented comparative analysis to identify process risks.
6.	For each change proposal, identify organization, people and product problems that were encountered.

3. Characterization Method

The purpose of the method that we use is to characterize a requirements change process to identify the *causes* of implementation problems in the process. One general approach where this purpose can be achieved is to identify situations in the process where implementation problems occurred (i.e., negative outcomes) and trace back through the process to identify the causes.

Briand et al. (1994) describe a method for characterizing a maintenance process by identifying links between problems faced by the maintenance organization and specific characteristics of the organization and its processes. With this method, one first identifies manifestations of process problems (in this case software errors), and then performs a causal analysis.

When the process is a front-end requirements change process for a large system, as is the case in our study, following the above approach is not immediately obvious. The reason is that some outcomes may not be known for a long time. For example, in our study some subsystems in the Technical Baseline will not be implemented for another decade, and therefore negative outcomes like software errors are not available at the time of characterization to do a causal analysis.

There are two paths to addressing this difficulty: identify other ways of evaluating outcomes (and hence identifying negative outcomes) that do not require waiting for data on software failures, and rely on expert opinion of potential causal factors. Our method adopts both approaches.

The method we propose can be used to develop a causal model of the form shown in Figure 1. It does so by tracing Change Proposals (CPs) through the change process. The final outcome is whether the process achieves its objectives (this is used as a surrogate measure of process implementation). If the process does not achieve its objectives then this is an implementation problem. We identify causal relationships of type A that specify process causes of not achieving the objectives. Organisational causes can lead to process problems (relationship type B). Also, organisational, people, and product problems may cause the process not achieving its objectives (type C). The steps of this method are presented in Table 1.

The three general techniques that we use in our analyses are called: deviation analysis,

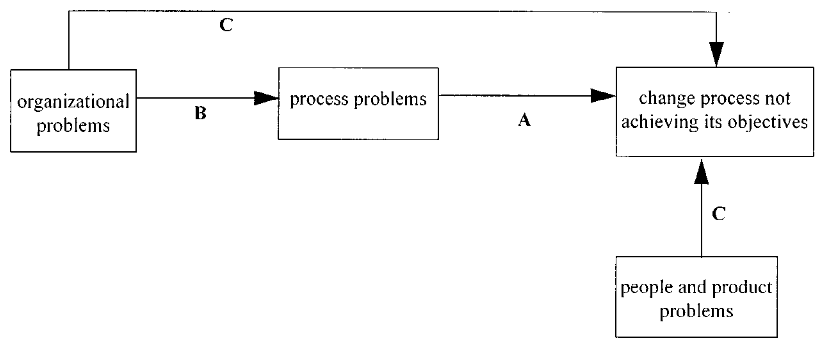

Figure 1. Causal model to be developed using the characterization method.

Table 2. Mapping between analysis technique and relationship type.

Technique	Relationship Type
Deviation Analysis (using prescriptive process models)	A
Comparative Analysis (using the method in (Ragin, 1987))	B
Anchoring (using structured interviews)	C

case-oriented comparative analysis, and anchoring. A summary of the relationship types and the techniques used is given in Table 2.

4. Results

The results we present here provide sufficient information to facilitate a general understanding of the characterization method that was used and its findings. We focus on only one phase of the requirements change process, namely Preliminary Analysis.

4.1. Prescriptive Process Model

The prescriptive process model was built shortly after the Technical Baseline was put under configuration control and when only the first few Change Requests were being received. The change process consists of four major phases. Inputs are the Technical Baseline (TB) and comments that can be provided by the units in the head office, by the regional offices or by the end-users of the system. Outputs are a changed Baseline and an Analysis Report documenting the rational of the changes. This process is a logical one that reflects the general need to exercise control over changes to a large document (about 7000 pages) and also the need for different levels of approval by interdisciplinary committees.

The four steps are outlined below. We describe the Preliminary Analysis phase in more detail since it is the focus of our results.

1. Initial Issue Evaluation

   The purpose of this step is to validate the comments and enter them as a Change Request (CR) into a database. If the CR addresses a problem that is within the scope of the Technical Baseline and that has not been addressed before, a Change Proposal will be generated.

2. Preliminary Analysis

   The objective of this phase is to develop a conceptual solution to the problem outlined in a CP and to get an Approval in Principle from an appropriate board. The Preliminary Analysis involves three substeps (see Figure 2). The first step, Prepare Analysis Plan, involves the formation of an Advisory Group and the development of a detailed description of the problem. If the Analysis Plan has been approved by the Configuration Control Board (CCB), the second step, Carry Out Analysis Plan, can be initiated. The objective of this step is to develop potential solutions and, with the CCB's approval, to select the solution to be implemented. This step involves usually several Advisory Group meetings in which the problem is being analyzed and in which the potential solutions are being discussed. The last step, Get Approval in Principle, involves the preparation of a preliminary Analysis Report which is then presented to the appropriate board (e.g., the CCB) to obtain its Approval in Principle.

3. Detailed Change Analysis

   When the preliminary Analysis Report has been approved, all documents constituting the Technical Baseline have to be analysed to identify the necessary changes.

4. Implementation

   This final step in the process contains activities that deal with the publication of a new release of the Technical Baseline and the closing of the initial Change Request(s).

4.2. CP's Analyzed

There are three types of CPs: design, rationalization, and procurement. Design CPs involve the design of new subsystems that would be included in the Technical Baseline. Procurement CPs involve updating the Technical Baseline to incorporate subsystems that will be purchased, not designed. Rationalisation CPs are those requiring a clean-up of the Technical Baseline, such as a deletion of subsystems or quantities of installed subsystems, and maintaining the consistency amongst the subsystems.

In the following we give an overview of the technical background of five of the seven CPs that we analyzed:

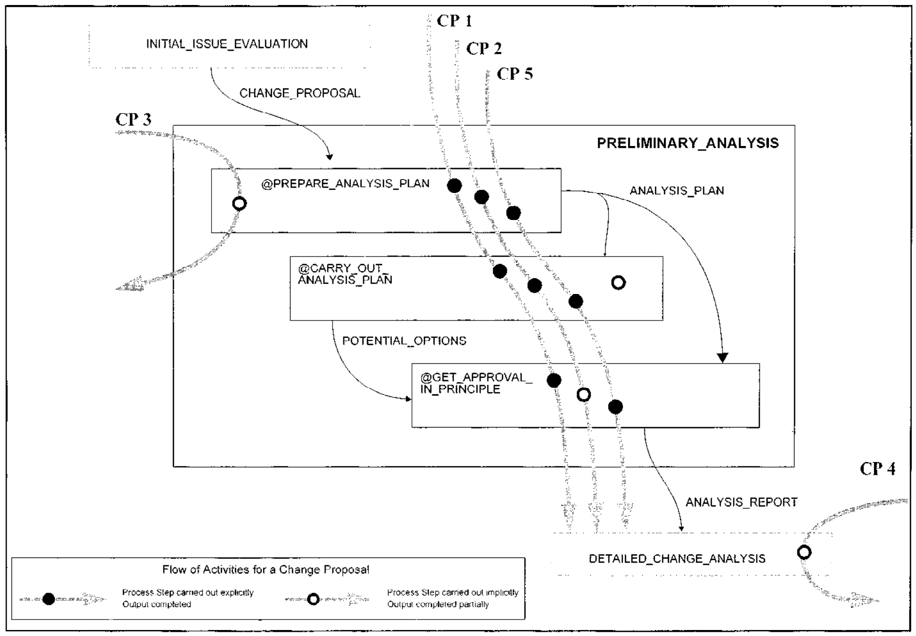

Figure 2. Preliminary analysis: flow of events.

- **CP 1**: This addresses a change of a high level policy. Thus, it has an impact on all documents comprising the Technical Baseline. This CP went through the TB Change Process in the way it was intended, because the solution became clear very early in the process and the details being discussed in the Advisory Group meetings were very specific. In addition, this CP was backed by a group that communicated well with the SEO.

- **CP 2**: This CP involves the procurement of a new subsystem which is going to be installed in several sites across the country. This procurement has been started in 1987 and until today, two phases have been completed. CP 2 refers to the third phase in which the last four sites will be equipped. Thus, this CP merely deals with updating the TB to reflect the new status at the four sites receiving the new equipment. Since, there are no detailed technical problems to be solved in this CP, it is handled very informally. The Analysis Plan has only been drafted, and the Analysis Report will not be written.

- **CP 3**: The objective of this CP is to integrate and enhance the systems installed in a particular type of site ("Sugar Shack"). It overlaps with Project A (see CP 5) and many other projects that involve systems installed in Sugar Shacks. To coordinate these projects, a very high level manager has set up a Sugar Shack Task Force. Since this task force has more weight than the SEO, the head of the SEO decided not to follow the

TB Change Process, but to translate the output of the Sugar Shack Task Force directly into TB terms.

- **CP 4**: This CP removes a system from the Technical Baseline which has been installed only as a prototype in one very small site. The development of this system has been cancelled, because today a better technology is available. Since the prototype will be kept in operation, this CP results only in a change of quantities and locations in the Transition Plan. Therefore, no Preliminary Analysis will be carried out. Also, an Advisory Group could not be set up because the project office responsible for this system has been discharged.

- **CP 5**: Organization X has a multi-million dollar project, Project A, to modernize its national system to meet the demands of the next century. Project A is highly political, behind schedule and suffers big cost overruns. Since budget cuts are ubiquitous, Project A is being scaled down. The objective of CP 5 is to repackage the functions that are being dropped from Project A in the Transition Plan. In the case of CP 5, the SEO is not the driving force of the events, and because of the political importance of Project A, the engineers have to rely strongly on informal contacts across Organization X and other government departments.

4.3. Tracing Results

In Figure 2 we illustrate how the selected seven Change Proposals were being processed in a graphical way. The bold lines in the activity chart illustrate the flow through the process. A black dot indicates that a process step was explicitly carried out or that an output artifact was completed. White dots, on the other hand, illustrate an informal execution of a process step or a partial completion of an output artifact.

4.4. Relationships of Type A

For the purposes of this paper we focus on two process problems to illustrate the general analysis method. These two are: planning difficulties and documenting rationale for the change.

4.4.1. Planning Difficulties

In some cases an Analysis Plan is not produced and consequently there is no plan to follow. Inadequate planning creates the risk of not having sufficient resources to keep the Technical Baseline up-to-date, and to reflect changes in the subsystems that are to be developed and/or procured. This increases the danger of an inconsistent requirements document.

4.4.2. Documenting Change Rationale

Even though the formal process definition of the SEO specifies that an Analysis Report that documents the decisions made during the analysis is produced at the end of the Preliminary Analysis process, this is not always done. Lack of explicit rationale documentation raises the risk of not capturing the consensus that was developed amongst the various disciplines and groups within Organization X.

4.5. Relationships of Type B

In three cases an Analysis Plan was produced. For the remaining cases an Analysis Plan was not produced. We looked at whether decision making for addressing the CP was being done internally by the SEO through its usual Advisory Group process, or the decision making was carried out outside the SEO by some other unit in Organization X. If decision making is under the control of the SEO then it is possible to make plans and actually follow them. If decision making is not under the control of the SEO then planning is not effective since they have no or little influence on whether the plan that is produced can be followed or whether the plan is realistic.

In all three cases where decision making was internal to the SEO a plan was produced and followed. In all cases where decision making was external to the SEO, a plan was not produced. An alternative explanation is the source of the CP. If the SEO themselves generate the CP by monitoring events external to the SEO, then this may lead to better planning because they would only generate CPs when sufficient information is available and when it is deemed possible to form an Advisory Group. For CP1 this was the case where there were no planning difficulties, but was also the case for CP3 where planning difficulties were faced. This eliminates *source of the CP* as a causal factor.

For the design rationale process problems, in six out of the 7 CPs an Analysis Report was not produced. One design CP that also had been discussed extensively before a change request was generated was the only one for which a report was produced. All of the other CPs did not meet both of the above conditions. The causal mechanism that would explain this is that the extensive prior discussion of the issues related to that CP produced much of the documentation and analysis required for the Analysis Report. Therefore it was an easy task to incorporate that material in an Analysis Report. For the remaining CPs, if they were design CPs then it was a substantial effort to produce an Analysis Report. For the non-design CPs it may have been perceived that the CP requires numerous small changes throughout the Technical Baseline that it would have been a monumental task to document all of these minor changes. The existence of general resource constraints is a possible supporting explanation for the above finding.

Finally, all of the CPs were generated by the SEO, therefore there was general invariance on this factor (i.e., who generates the original change request). However, this is probably an indication of an implementation problem because it seems that the users of the Technical Baseline are not providing proactive feedback to keep the Baseline documents updated.

4.6. Relationships of Type C

During the structured interviews, we identified a number of type C relationships. Some of these are:

1. The detailed impact analysis that is performed depends on the analyst's knowledge of the Technical Baseline documents and the interconnections between their components. Most of this work is currently done by an external contractor. The contract is soon to come to an end. This creates the risk that the SEO will have difficulties, at least during the transition period, in performing an adequate impact analysis.

2. New employees on the contractor side do not have intimate knowledge of how to deal with government agencies, and this sometimes causes friction. Given that there is a strong dependency during the Preliminary Analysis process on personal contact to obtain information, friction-causing incidents can risk alienating informants.

3. In some cases, domain experts recruited to the Advisory Groups do not have good knowledge of the system requirements documents. This tends to lead to more effort spent during valuable meetings explaining to them the contents of the systems requirements documentation rather than discussing potential impacts on the national system and resolving conflicts.

5. Conclusions

In this paper we described a study to analyse a requirements change process with the intention of starting an improvement program based on the findings. We applied a low-cost risk analysis methodology to identify organization, process, and people problems in implementing a change process. The findings from our study serve as input for process improvement within this organization.

It will be noticed that no product related problems were presented. There are a number of possible explanations for this. The obvious one is that we presented the results for the Preliminary Analysis phase only. For the tasks performed during this phase, no specific product problems were identified, but this does not necessarily hold for other phases. There are other plausible alternative explanations. First, the method that we followed may not have allowed us to discover product type problems if they did exist. For example, product problems may be antecedents to organizational problems, and such antecedents were not considered within the framework of the causal model. Second, our study may have been less than ideal in that we were able to interview only those individuals whom the SEO decided we could interview. This means that there may have been a positive bias in our sample of informants. Third, the system has been in maintenance for a relatively short time compared to the life time of the system, and therefore the structure of the requirements specification documents may still have not been affected by the continuous changes.

In terms of future research, it would be of value to further investigate the application of desired or prescriptive process models in the context of improvements to maintenance processes. Previous research has focused mainly on the application of descriptive process models in maintenance contexts (e.g., Briand, et al., 1994) or in general (e.g., Bandinelli, et al., 1995; Kellner and Hansen, 1989) for process improvement. Moreover, research on developing prescriptive process models would be of added value. Given that it took more than three times to develop a prescriptive process model, improvements in that part of the characterization method can reduce its costs even more.

Based on our experience, we have found the case comparative technique to be a useful tool for the analysis of qualitative data. Further applications and refinements of this technique in the context of analyzing change processes is warranted.

Notes

1. The work reported in this paper was done while the authors were at the Centre de recherche informatique de Montreal, Canada.

References

Bandinelli, S., Fuggetta, A, Lavazza, L., Loi, M., and Picco, G. 1995. Modeling and improving an industrial software process. *IEEE Transactions on Software Engineering* 21(5): 440–454.

Briand, L., Basili, V., Kim, Y-M and Squire, D. 1994. A change analysis process to characterize software maintenance projects. *Proceedings of the International Conference on Software Maintenance* 38–49.

Brill, J., and Brammer, K. 1993. A framework for assessing systems engineering capability. *Proceedings of the Third Annual International Symposium of the National Council on Systems Engineering* 643–650.

Edwards, M., and White, S. 1994. Re-views: A requirements structure and views. *Proceedings of the Fourth Annual International Symposium of the National Council on Systems Engineering* 73–78.

Kellner, M., and Hansen, G. 1989. Software process modeling: A case study. *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences*, January.

Kitson, D., and Masters, S. 1993. An analysis of SEI software process assessment results: 1987–1991. *Proceedings of the 15th International Conference on Software Engineering* 68–77.

Mar, B. 1993. Systems engineering maturity model. *Proceedings of the Third Annual International Symposium of the National Council on Systems Engineering* 631–635.

Ragin, C. 1987. *The Comparative Method*. University of California Press.

Simon, B. 1994. Requirements management implementation roadblocks. *Proceedings of the Fourth Annual International Symposium of the National Council on Systems Engineering* 79–82.

Software Engineering Institute 1995. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley.

Evaluating Impact Analysis—A Case Study

MIKAEL LINDVALL mikli@ida.liu.se

Department of Computer and Information Science, Linköping University, 58183 Linköping, Sweden

Introduction

Impact Analysis (IA) is an important release planning activity analyzing how suggested new requirements affect an existing system. IA is the foundation for many other early