Causal Analysis of the Requirements Change Process for a Large System

Khaled El Emam
Fraunhofer IESE
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
elemam@iese.fhg.de

Dirk Höltje
School of Computer Science
McGill University
3480 University Street
Montréal, Québec
Canada H3A 2A7
dirk@cs.mcgill.ca

Nazim H. Madhavji
School of Computer Science
McGill University
3480 University Street
Montréal, Québec
Canada H3A 2A7
madhavji@cs.mcgill.ca

International Software Engineering Research Network Technical Report ISERN-97-23

Causal Analysis of the Requirements Change Process for a Large System*

Khaled El Emam
Fraunhofer IESE
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
elemam@iese.fhg.de

Dirk Höltje
School of Computer Science
McGill University
3480 University Street
Montréal, Québec
Canada H3A 2A7
dirk@cs.mcgill.ca

Nazim H. Madhavji
School of Computer Science
McGill University
3480 University Street
Montréal, Québec
Canada H3A 2A7
madhavji@cs.mcgill.ca

Abstract

Implementations of requirements change processes in large system projects face many difficulties. In this paper we present a method for analysing requirements change processes to identify implementation weaknesses and their causes. This method relies on prescriptive process models and tracing actual Change Proposals through the process model. We apply the method in the analysis of the requirements change process for a large real-time system within a Canadian government agency. This allowed us to identify the process implementation problems, and the process, organisational, and people causes of these problems. Based on that experience, we draw general conclusions about the proposed method and its applicability.

1 Introduction

The implementation of good software requirements management practices is believed to be one of the first process improvement steps that an organisation 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) [10]. One analysis of data available at the SEI found that 61% of assessments had findings that map to the requirements management KPA [5]. The above two points indicate that improving software requirements management practices is an industrial priority.

In the related discipline of systems engineering, system requirements management (SRM) practices are also believed to be some of the first that should be implemented to increase systems engineering maturity. For example, in one systems engineering maturity model generalised from the CMM for software, the SRM KPA is at level 2 [2]. In another multi-dimensional systems engineering maturity model, one of the maturity dimensions stipulates that the implementation of a systems engineering paradigm in the early life cycle phases is one of the first signs of maturity [7].

A strong emphasis on implementing SRM practices is not surprising given that implementation weaknesses could lead to problems such as: difficulties in eliciting clear and timely requirements changes, inability to obtain consensus on changes from key stakeholders, inability to maintain the requirements document in a timely and internally consistent manner, and inability to adequately estimate the resources necessary to modify, design, integrate and/or procure systems that meet the new requirements. Such difficulties are accentuated with large systems due to the initial size of the requirements documentation, the larger quantity of changes and the more numerous external organizations that need to be interfaced with.

The implementation of SRM remains a challenging problem, however. This is evidenced by the number of implementation barriers that have been mentioned in the literature, such as [3][9]: achieving buy-in, 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 organisational politics.

In this paper we present a low cost analysis method that can be applied to identify implementation problems in a requirements change process, and to identify the causes of these problems. We apply the method to analyse a process that maintains the requirements for a large real-time system within a Canadian government agency. Once such problems are identified, they can serve as the basis for improving SRM practices.

In the next section we describe the organisation where this study was conducted. We then outline the analysis 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 organisation (henceforth Organisation X) started a systems engineering project. During this project, a Technical Baseline (TB) 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 TB, and is performed by the Systems Engineering Organisation (SEO).

The TB specifies a Canadian national system that operates in geographically distributed locations. The lifetime of this system is expected to be at least a couple of decades. Some parts of the system are already operational, while others are earmarked for development and deployment within the next decade. The program of system development and

* The work reported here was conducted while the authors were with the Centre de recherche informatique de Montréal, Canada.

procurement to make the complete system operational is documented in a Transition Plan, which is also part of the TB. The Transition Plan is intended to be used by senior management for prioritising subsystem development and procurement, allocating human resources and budgets, and monitoring the development efforts.

Organisation X has a central headquarters and a number of largely autonomous regional centres that operate, maintain, as well as develop and procure systems. Within Organisation 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.

Organisation X is facing severe budget cuts. This created pressure to take some process improvement actions within the SEO.

The main objective of our study was therefore to identify opportunities for improving the change process. Following some initial discussions with members of the SEO, it became apparent that the most important issue at this point in time was the *implementation* of the requirements change process. Implementation in this context means that good SRM practices are actually performed.

3 Analysis Method

The purpose of our method is to identify change process implementation problems and their causes. 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. [1] describe a method for analysing change processes by linking problems faced by the maintenance organisation with specific characteristics of the organisation and its processes. With this method, one first identifies manifestations of process problems (in this case software errors), and then performs a causal analysis.

The general approach of Briand et al. needs to be adapted, however, when analysing a front-end change process. The reason is that some outcomes may not be known for a long time. For example, in our study some subsystems in the TB will not be implemented for another decade, and therefore negative outcomes like software errors or failures from the field are not available at the time of the study to do a causal analysis. To address this difficulty we need another approach for evaluating outcomes that does not require waiting for data on system failures.

The method that we propose can be used to develop a causal model of the general form shown in Figure 1. The final outcome is whether the process achieves its objectives (this serves 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 three types. Type A specify process causes of not achieving the objectives. Organisational, people, and product problems can lead to process problems (relationship type B). Also, organisational, people, and product problems may cause the process not achieving its objectives (type C).

It is not necessary that all relationships in the causal model will exist in a single context. The model of Figure 1 is based on our collective experiences, and therefore may not fit exactly in all application contexts. However, it could serve as a useful starting point.

The steps of the method are presented in Figure 2. The three general techniques that we use in our analyses are called: deviation analysis, comparative analysis, and anchoring. A summary of the relationship types and the techniques used is given in Figure 3. Below we present each of the six steps.

3.1 Define List of Potential Causal Factors (Step 1)

We first need to define a list of potential organisational, product, and people factors that may be related to the process achieving its objectives. This list can be based on common sense and the literature. In Figure 4 we present an initial list that we have used.

3.2 Develop Prescriptive Process Model[1] (Step 2)

For the first step a model of the *desired* process must be identified if one exists, or developed if one does not exist. If one exists, it may need to be formalised for the purpose of the analysis. In our study, an informal model of the desired process existed. This model was developed by the SEO as a manifestation of an *ideal* process that would meet the objectives of the requirements change process. The development of this model utilised inputs from personnel who enact the process, the customers of the process (e.g., project managers who manage projects to develop or procure systems), and national and international standards.

The desired process model that existed was formalised by the research team by translating it into the notations of Statemate (see [4]). This model describes the activities, the documents produced and consumed, and all the roles and committees involved in the change process. The process attribute structure described in [6] was used to elicit information and structure it.

3.3 Select Change Proposals and Trace Them (Step 3)

The engineers of the SEO selected 7 Change Proposals (CPs) which they handled. These were considered to represent a cross-section of typical CPs. We denote this set of CPs as Y.

We then produced an event sequence for each CP. An event is either a performance of an activity or a production of an artifact. This was mostly based on interviews with engineers of the SEO and its contractors, and was augmented with the inspection of documentation that was made available (e.g., original Change Requests and Analysis Plans and Reports).

1.We use the term “prescriptive process model” to refer to the documentation of the desired process.

Figure 1: Causal relationships to be explored using the analysis method.

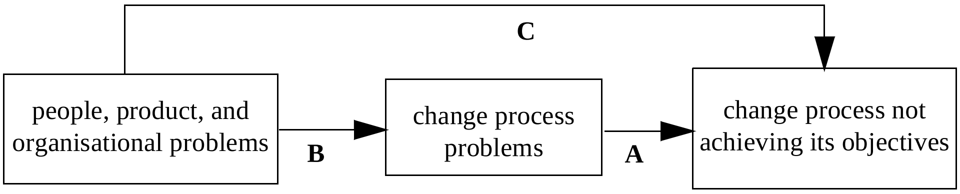


Figure 2: A Summary of the steps of the analysis method.

1. Make a list of organisational, product, and people factors that may potentially have an impact on the process achieving its objectives.
2. Develop a prescriptive process model that reflects the engineers' concepts of an ideal process that is to be enacted.
3. Select several past change proposals for analysis and produce an event sequence.
4. Compare the event sequence of the actual handling of the change proposal to that expected in the prescriptive process model to identify process problems (deviation analysis).
5. For each change proposal, perform a case-oriented comparative analysis (see [8]) to identify organisational causes of process problems.
6. For each change proposal, identify organisation, people and product causes of implementation problems.

Figure 3: Mapping between analysis techniques and relationship type.

Technique	Relationship Type
Deviation Analysis (using prescriptive process modeling)	A
Comparative Analysis (using the case-oriented method in [8])	B
Anchoring (using structured interviews)	C

Figure 4: Example of an initial list of non-process causes of implementation problems.

Organisational Problems: weak senior management commitment (e.g., lack of resources, lack of support for making decisions about changes), resistance to change by users and project managers (e.g., users not compensated financially for the time they spend reviewing requirements documents, perceptions of a bureaucratic change process by users, project managers do not see the benefits of a formalized change process), lack of control over the process by the SEO, inexistence of formal information channels for gathering information about subsystems, and difficulty to obtain information about major planned changes to systems in the field.

People-related Problems: lack of expertise of participants in the process, lack of knowledge about the requirements documents by the domain experts who are consulted, turnover of competent staff, and ability to identify and schedule competent staff to review the requirements documents.

Product-related Problems: inappropriateness of level of detail for various review committees, terminology differences amongst the various disciplines and between them and the SEO, links are not defined amongst the components of the document(s) to facilitate impact analysis, and the changed requirements documents are not appropriate for the purposes of the users (e.g., for use by project managers for procurement of systems).

The type of information we collected for each CP in order to do the tracing were: the technical contents of the CP, a description of the steps that were actually carried out, and the artifacts that were produced and their description.

During the tracing interviews, we also collect information on non-process factors and the characteristics of the change that are relevant to each CP. For the non-process factors, we use the original list described above as a starting point. Characteristics of the change can be, for example, the type of change, the amount of changes made to the requirements documentation, and existing consensus in Organisation X on how to handle the issue brought up by the change. This kind of information was used in a subsequent analysis.

3.4 Perform a Deviation Analysis (Step 4)

One objective of the event sequence was to identify deviations of the real process from the prescriptive process model. This is achieved by mapping the event sequence to the model.

We use deviations from the prescriptive process model to identify CPs where the objectives of the change process are not achieved. These deviations represent process problems (relationship type A). This is based on the premise that if adherence to the prescriptive model is low, then the requirements change process would not achieve the objectives of the system requirements document.

The identification of deviations is of course not mechanical and an element of common sense has to be applied. In general, we looked for incomplete intermediate products, activities not executed but should be, and prescribed activities that were substituted by other activities. We denote this subset of CPs as N. The subset of CPs Y/N is therefore those where the objectives of the change process were achieved.

3.5 Perform a Comparative Analysis (Step 5)

We follow a case-oriented comparative analysis technique as described in [8] to identify relationships of type B. In particular, we follow an adaptation of the *indirect method of difference*. With this method, each CP is treated as a case. The sequence of steps of this technique are described below.

Identify Invariance in N Cases

We wish to identify the non-process problems that cause process problems. By collecting information on non-process problems during the tracing, we characterise each of the above cases by their non-process problems. We then search for the subset of non-process problems that are invariant amongst all of the N cases. This subset of non-process problems, denoted T, are considered to be potential causes of the process problems.

Identify Invariance in Y/N Cases

If our potential set of causes (T) of process problems is correct, then we would expect the lack of T would be associated with a lack of process problems. We therefore check that the identified non-process problems are not true for the Y/N cases. If they are true then there is increased doubt whether the set T are causes of the process problems. In such a situation, identify a subset of T that are not true for the Y/N cases. These then become the causal factors.

If Y/N is empty then this may indicate that the CPs were not selected properly or that there is a general problem that has an impact on *all* of the CPs. Judgment should be applied in interpreting the results in these circumstances.

3.6 Anchoring (Structured Interviews) - (Step 6)

Organisation, people, and product factors may have a direct impact on the achievement of objectives (relationship type C). In particular, non-process problems do not all have to be causes of process problems. Furthermore, non-process problems may be general and therefore not easily associated to a particular CP, which makes the application of the comparative technique inappropriate.

Since we do not have a direct measure of the achievement of process objectives, we use expert opinion to identify those causal factors. Experts are those who enact or interface with the process. Based on the premise that expert opinion is a sound basis for identifying causal relationships, the issue now is how to elicit this opinion in a structured manner.

Using the events in addressing each of the CPs as anchors, identify organisation, people and product problems at each step of processing each CP. This is achieved through structured interviews with those who enact the change process by asking if any organisational, process and/or product difficulties were experienced for each CP. The basic set of organisational, process and product problems identified earlier are used as a starting point for the questions.

4 Results

In the results, we focus on only one phase of the requirements change process because that is the one where the most critical implementation problems were witnessed, namely, Preliminary Analysis.

4.1 Prescriptive Process Model

The change process consists of four major phases. Inputs are the 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 TB and an Analysis Report documenting the rationale for 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.

For this paper, the most relevant committee is the Configuration Control Board (CCB). The CCB oversees the analysis of the comments and solicits help from Advisory Groups (consisting of domain experts) that are summoned for each CP.

The four phases 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 phase is to validate the comments and enter them as Change Requests (CRs) into a database. If a CR addresses a problem that is within the scope of the TB and that has not been addressed before, a CP 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 subactivities. The first activity, 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 CCB, the second activity, Carry Out Analysis Plan, can be initiated. The objective of this activity is to develop potential solutions and, with the CCB's approval, to select the solution to be implemented. This activity involves usually several Advisory Group meetings in which the problem is analysed and in which the potential solutions are discussed. The last activity, Get Approval in Principle, involves the preparation of a preliminary Analysis Report which is then presented to the appropriate board to obtain its Approval in Principle.

3. Detailed Change Analysis

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

4. Implementation

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

4.2 CP's Analysed

In the following we give an overview of the technical background of each CP that we analysed:

- **CP 5:** This addresses a change of a high level policy. Thus, it has an impact on all documents comprising the TB. This CP went through the 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 14:** This CP involves the procurement of a new subsystem which is going to be installed in several sites across the country. It refers to the last phase in which the last 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.
- **CP 16:** This CP is related to a new project within Organisation X. In this project, Organisation X will incorporate a complete subsystem into the TB. This subsystem is part of a system developed and to be operated by an external organisation. Organisation X will only purchase and install additional equipment.
- **CP 18:** This CP only studies the description of certain functionality in the TB and how it can be integrated in the description of other subsystems.
- **CP 30:** 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 33) 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 phases of the change process, but to translate the output of the Sugar Shack Task Force directly into TB terms.
- **CP 32:** This CP removes a system from the TB 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.
- **CP 33:** Organisation X has a multi-million dollar project, Project A, to modernise 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 33 is to re-package the functions that are being dropped from Project A in the Transition Plan. In the case of CP 33, the SEO is not the driving force of the events, and so the SEO has to rely strongly on informal contacts for up-to-date information on directions and decisions within Project A.

4.3 Results of Tracing

In Figure 5 and Figure 6 we illustrate how the seven selected CPs were being processed in a graphical way for all of the Preliminary Analysis phase and for its first activity respectively. The bold lines in the activity charts 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*.

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 obtaining sufficient resources when required to keep the TB up-to-date. This increases the danger of an inconsistent requirements document. For example, an Analysis Plan was not produced for CP16 and CP32 (see Figure 5).

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 organisation X, e.g., an analysis report was not produced for CP18 (see Figure 5).

Figure 5: Preliminary Analysis: Flow of events.

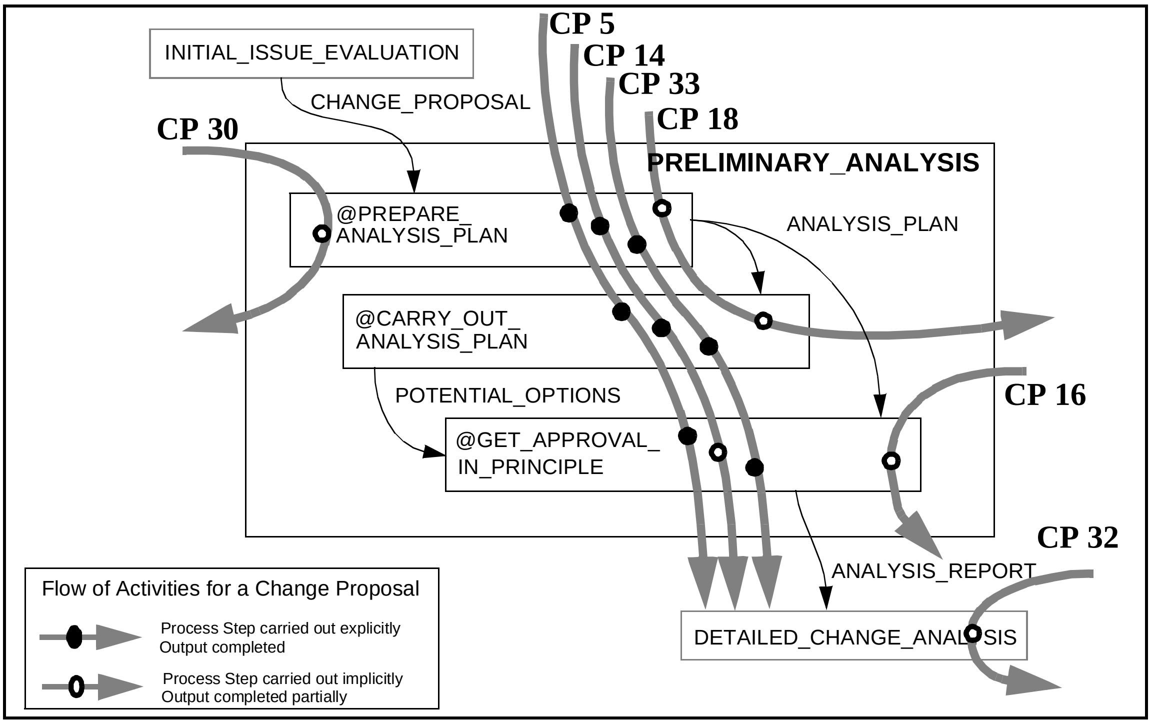
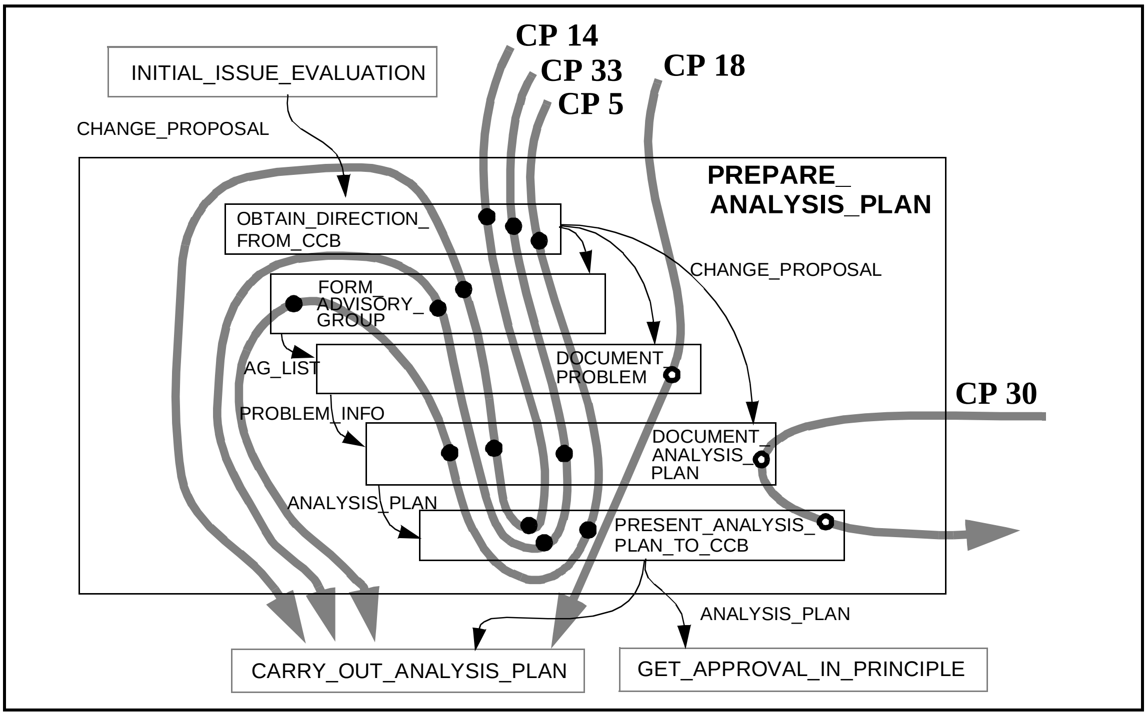


Figure 6: Prepare Analysis Plan: Flow of events.



4.5 Relationships of Type B

For the comparative analysis, a table such as that shown below is constructed. This shows the boolean values for the potential non-process problems. In three cases an Analysis Plan was produced and followed (CP5, CP14, and CP33). For the remaining four 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 Organisation X (see column *Advisory Group Formed?*).

Change Proposal	Origin of Change Proposal is SEO ?	Advisory Group formed ?	Analysis Plan Produced?	Analysis Plan Followed?
CP5	Y	Y	Y	Y
CP14	Y	Y	Y	Y
CP16	Y	N	N	N
CP18	Y	N	N	N
CP30	Y	N	N	N
CP32	Y	N	N	N
CP33	Y	Y	Y	Y

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.

In all three cases where decision making was internal to the SEO a plan was produced and followed (see the CP5, CP14, and CP33 flows in Figure 6). In all cases where decision making was external to the SEO, a plan was not produced.

Another potential causal explanation for the planning difficulty process problem is the origin 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 CP5 this was the case where there were no planning difficulties, but was also the case for CP30 where planning difficulties were faced. This eliminates *origin of the CP* as a causal factor.

For the design rationale process problems (table not shown), in six out of the 7 CPs an Analysis Report was not produced. One design CP (CP5) 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 involved the design and development of new subsystems, 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 TB that it would have been a monumental task to document all of these minor changes. The existence of resource constraints is a possible supporting explanation for the above finding.

4.6 Relationships of Type C

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

1. The CPs we analysed were originally generated by the SEO. This is an indication of an implementation problem because it seems that the users of the TB are not providing proactive feedback to keep the TB documents updated.
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. This issue was raised during the analysis of CP14 where new staff were hired on the contractor side during its analysis.
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 requirements documentation rather than discussing potential impacts on the national system, resolving conflicts and coming to consensus. This issue was raised for CP5.

4.7 Discussion: Appraisal of Causal Analysis Method

Based on our experiences applying the method, we can present an appraisal of its main advantages and disadvantages, both from a practical and a methodological perspective. We also present some general guidance on applying the method.

One point before this appraisal needs to be clarified however. It will be noticed that no product related problems were identified in our analysis. There are a number of possible explanations for this. The obvious one is that we presented the results from 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 the other phases. Other plausible explanations can be provided. First, the method that we followed may not have allowed us to discover product type weaknesses if they did exist. Second, in our study we were able to only interview those individuals whom the SEO decided we could interview. This introduces the risk of positie bias. Third, the TB has been under change for a relatively short time compared to its lifetime, and therefore the structure of the requirements specifications documents may still not have been affected by the continuous changes.

Advantages

Following our method leads to a clear link between the information collected and the conclusions about process implementation problems that were drawn. Because of this explicit relationship to actual events, it is easier for the organisation to decide what improvements are necessary. Furthermore, identification of causes and the production of a prescriptive process model provide a strong basis for starting improvement actions.

The method we used is particularly suited to the analysis of front-end change processes where software or system failure data are not available yet. The comparisons with a prescriptive model provide the basis for identifying process problems.

In terms of costs, we estimate that, after the prescriptive process model was constructed, our application consumed 40 person hours (excluding travel, report writing and reviews, and administrative overheads, and including data collection and qualitative data analysis). Formalizing the prescriptive process model is estimated to have taken 136 person hours. The costs of developing a prescriptive process model is expected to vary depending on the particular organisational context.

Weaknesses

When the comparative method is applied mechanistically, multiple causation is difficult to identify using the case-oriented comparative method [8]. For example, if E *or* F are causal factors then a case oriented strategy will neither identify E nor F as causal factors.

Also, it is clear that the evidence that can be brought to support the causal relationships of type C is weaker than the evidence that can be used to support relationships of type A and B. Furthermore, the identification of relationships of type C is more dependent on the skills of the interviewer and the analyst.

Application Guidance

Based on our experiences applying the method, in every instance where there was deviation from the prescriptive process model, substantive change process implementation problems were identified. However, it should be emphasised that this conclusion follows directly from having a good prescriptive process model. In general, this can be attained by customising international, national, industry or professional body standards. Some organisations may have customer-stipulated processes that they have to follow, and these can be used as the basis for developing a prescriptive process model. If the prescriptive process model does not reflect a process that is capable of achieving business objectives, then deviations from it would likely not be indicative of implementation problems.

Furthermore, in the case where there are numerous CPs that exhibit similar deviations, this may be an indicator that there is a certain class of CPs that ought to be dealt with by a different process altogether. Therefore, consistent deviations should be examined closely to determine if there is a need for another desired process.

5 Conclusions

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

In terms of future research, it would be of value to further investigate the application of prescriptive process models in for process improvement. Previous research has focused mainly on the application of descriptive process models in maintenance environments (e.g., [1]) for process improvement. In addiotion, research on developing prescriptive models would be of added value given that it took proportionally more effort during our study to build such a model.

Acknowledgements

We wish to thank Martin Verlage for his comments on an earlier version of this paper.

References

[1] L. Briand, V. Basili, Y-M Kim, and D. Squire: “A Change Analysis Process to Characterize Software Maintenance Projects”. In *Proceedings of the International Conference on Software Maintenance*, pages 38-49, 1994.

[2] J. Brill and K. Brammer: “A Framework for Assessing Systems Engineering Capability”. In *Proceedings of the Third Annual International Symposium of the National Council on Systems Engineering*, pages 643-650, 1993.

[3] M. Edwards and S. White: “Re-views: A Requirements Structure and Views”. In *Proceedings of the Fourth Annual International Symposium of the National Council on Systems Engineering*, pages 73-78, 1994.

[4] M. Kellner: “Software Process Modeling: Value and Experience”. In *SEI Technical Review*, pages 23-54, 1989.

[5] D. Kitson and S. Masters: “An Analysis of SEI Software Process Assessment Results: 1987-1991”. In *Proceedings of the 15th International Conference on Software Engineering*, pages 68-77, 1993.

[6] N. Madhavji, D. Höltje, W. Hong, and T. Bruckhaus: “Elicit: A Method for Eliciting Process Models”. In *Proceedings of the Third International Conference on the Software Process*, pages 111-122, 1994.

[7] B. Mar: “Systems Engineering Maturity Model”. In *Proceedings of the Third Annual International Symposium of the National Council on Systems Engineering*, pages 631-635, 1993.

[8] C. Ragin: *The Comparative Method*. University of California Press, 1987.

[9] B. Simon: “Requirements Management Implementation Roadblocks”. In *Proceedings of the Fourth Annual International Symposium of the National Council on Systems Engineering*, pages 79-82, 1994.

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