Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Computer Science 8 (2012) 279 – 284
Procedia Computer Science

New Challenges in Systems Engineering and Architecting
Conference on Systems Engineering Research (CSER)
2012 – St. Louis, MO
Cihan H. Dagli, Editor in Chief
Organized by Missouri University of Science and Technology

An empirical study of refactoring decisions in embedded software and systems

Sara Dersten[a,b,*], Jakob Axelsson[a], Joakim Fröberg[a,b]

[a]*Mälardalen University, SE-72123 Västerås, Sweden*
[b]*Volvo CE, SE-63185 Eskilstuna, Sweden*

Abstract

This paper describes an empirical study of decision-making when changing the architecture in embedded systems. A refactoring of the system architecture often gives effects on both system properties and functions in the company organization, and there is a lack of efficient analysis methods for decision support in the system architecture process. This study investigates the information needed to make a decision about a system refactoring. Scenario-based interviews have been conducted with managers and system architects from companies developing embedded systems. The results show that the companies investigate similar issues regardless of their industry sector. The most wanted information prior to a decision is also presented.

Keywords: System refactoring; system evolution; system architecture; software architecture; decision-making; empirical study

1. Introduction

The main drivers behind the evolution of distributed embedded systems include new customer functionality, technology shifts, legislation, and cost reductions. Sometimes an effort is required to prepare the software or system to be able to fulfill these drivers. The system complexity might have to be reduced or the system might have to be more flexible to meet future demands. We call such changes, which are not directly related to the visible functionality of the system, *system refactoring*.

* Corresponding author. Tel.: +46-16-541-42-08 ; fax: +46-161-529-67 .
E-mail address: sara.dersten.2@volvo.com .

1877-0509 © 2012 Published by Elsevier B.V. Open access under CC BY-NC-ND license.
doi:10.1016/j.procs.2012.01.058

The benefits of system changes related to new customer functionality are relatively easy to calculate and motivate for management. For system refactoring it is harder to know if it is worth to invest in a new technology because the benefits come at a later stage and are uncertain. Earlier studies of system refactoring [1, 2] show that changes in system architecture not only impact the system and its properties but also the processes related to the entire life cycle of the system, such as development, maintenance, verification, supplier cooperation, product planning and sale of the product on the market. This paper presents a study of what information decision-makers want when deciding about a proposed system refactoring. Scenario-based interviews with persons who make such decisions, i.e. system architects and system managers, at companies producing embedded systems were conducted. With an improved understanding of the drivers behind system refactoring decisions, we hope to lay a foundation for a guideline to be used in the system refactoring process. The remainder of the paper is organized as follows. Section 2 gives an overview of the related work in the area. The study methodology is explained in Section 3 and an analysis of the collected data is presented in Section 4. Section 5 discusses the results and Section 6 concludes the paper.

2. Related work

There already exist several trade-off analysis methods for evaluation of system architecture alternatives, such as ATAM [3], CBAM [4] and Pugh's matrix [5], but none of these methods explicitly deals with effects on the entire company. Saliu and Ruhe [6] have evaluated several release planning methods and point out that there is a lack of system focus in most such methods, which are necessities when planning system evolution. Therefore, we hope that this study will answer if the decision-makers investigate the current system and the impacts on it prior to a decision or if there is a lack of awareness of this issue amongst them. A systematic literature review on strategic release planning models was conducted by Svahnberg [7] et al. Their results showed that all models used so called "hard factors", such as technical, budget, resource, and time constraints as decision factors. 58% of the models used "soft factors" as decision factors. These are the factors that are hard to estimate, like values and risks. An interesting study by Wohlin and Aurum [8] presents criteria that are important when making decisions on which requirements to include or not in software projects. Decision-makers at companies producing software-intensive system were asked to rank the most important pre-defined criteria for the decisions. The result indicated that business-oriented criteria were the most important and the technical-oriented criteria less important. All these studies concern decisions to add new features or to plan new software releases. A system refactoring is not feature-driven and therefore it is more difficult to estimate the value of its benefits. The contribution of our study is to increase the understanding of the drivers that lead to system refactoring and to aid in the identification of the information that is required to make a decision about system refactoring.

3. Method

To answer our questions, we interviewed seven system architects and seven managers on different levels at development departments from eight companies that develop products containing distributed embedded software and that are used to make decisions about changes in the system architecture. The participating companies were from the automotive (five) and the automation (three) sectors. The interviews started by giving the respondents a start scenario that represented a suggestion of a change to be made in the system of the companies' products. The respondents were then asked to request the information they needed to complete the decision on whether the system change should be performed or not. After the respondent answered, pre-defined additional information related to the requested information was given. The respondents were then asked to request further information needed to complete the decision. This procedure was repeated until the respondents answered that they were able to

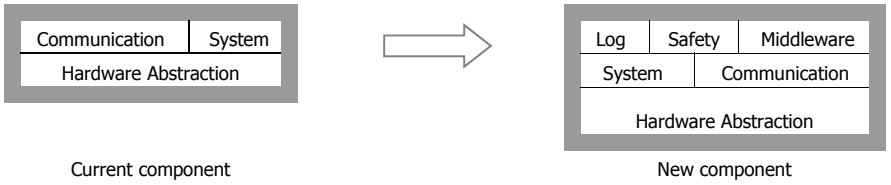

Fig. 1. Illustration of the current and the advanced component that was presented in the start scenario.

complete the decision or at least make a recommendation. During interviews the respondent sometimes asked for additional information that was not pre-defined. When this happened, new information was constructed on site and saved to be used in later interviews if the same information was requested again. In this way we were able to consistently catch information that we otherwise had missed and still have the ability to give the same pre-defined information to all respondents.

The aim of only giving the start scenario in the beginning of the interviews was to capture what information the decision-maker would investigate in a real-life case without any interference from us. Still, we wanted the interviews to be formalized to the extent, that they were analyzable and comparable. The start scenario was designed to fit in several different kinds of embedded distributed systems. In the start scenario the respondents were told that there was a suggestion to replace a current component in their product systems by a more advanced component. In this phase the respondents were not told about the content of the new or current component, but a figure of the component structure was shown to them. Fig 1 shows the illustration of the start scenario that was given to the respondents.

4. Analysis and result

The conducted analyses are based on complete interviews and on the early phases of the interviews when only the start scenario has been given to the respondents. We will also look at to what extent certain areas of the company organization are investigated by the respondents.

4.1. Most important information

The most important information that a decision-maker must have is measured by the extent to which the respondents at all ask for that specific information. We counted the numbers of respondents that requested information from each of twelve identified information areas at any time during all 14 interviews. The frequency of each of the areas is found in Table 1.

Table 1 shows the requested information areas. The frequency indicates how many of the respondents that requested a certain area.

Information area	Costs	Profits	Supplier	Technical details	Future requirements	Current requirements
Frequency	13	12	12	10	8	8
Information area	Enterprise constraints	External constraints	Design alternatives	Risks	Human requirements	Technical management
Frequency	8	7	7	6	3	1

Almost all of the respondents wanted more information related to cost, such as development costs, manufacturing costs and maintenance costs. Also profit information, related to technical advances and new market opportunities, were requested to a large extent. During the interviews, the respondents were

given the information that the component was available from a software supplier and many of the respondents wanted to investigate this supplier further in terms of viability and supportability. A majority of the respondents wanted more information on the technical details and how the components will impact the current system and if new engineering tools will be required. Less interest was given to the technical management of the component and to human requirements.

We also noticed that costs, profits and technical details were requested early during the interviews by the majority of respondents. This might be a further indication of that these areas are the most important. Interesting is that the least requested areas, technical management and human requirements, were requested early by those respondents that did request these areas.

4.2. Differences between respondents' answers

One could suspect that managers are interested in economic factors while system architects want to know about the technical details. We started to investigate if there were differences between which areas the two groups had requested information from, by looking at the complete interviews. We also made a separate analysis of what the respondents from each group had requested in the early phase of the interview. This was done by exploring what the respondents requested directly after being given the start scenario. The reason, for further exploring the start scenario, was to see if there was a difference between the groups in the way they chose to start their investigation when a decision is going to be made. The resulting frequencies of areas that each group requested information from is presented in Fig 2.

We could not see any clear difference between which kinds of information that is required to complete a decision, whether the decision-maker is a system architect or manager. Both groups seem to be highly interested in costs, profits and technical details. An explanation could be that both the system architects and managers have long experience within the company or from at least their industry sector. However, the two groups, managers and architects, seem to slightly differ in their starting points. We noticed that managers wanted information about costs and profits at an earlier stage in the investigation process than the system architects, who seem to have a need for first defining the problem that is going to be solved.

An analysis was also made on the consistency of answers within a specific company. At one of the companies that participated in the study, five people were interviewed. This company develops electronics for vehicles and amongst the respondents were two managers and three architects. We wanted to see if this group requested any type of information differently than the rest of the 14 respondents. The same analysis was also conducted to see if there were consistencies within companies from a certain industry sector. From our results we could not say with statistical significance that a certain information entity is always requested within a certain company or a certain industry.

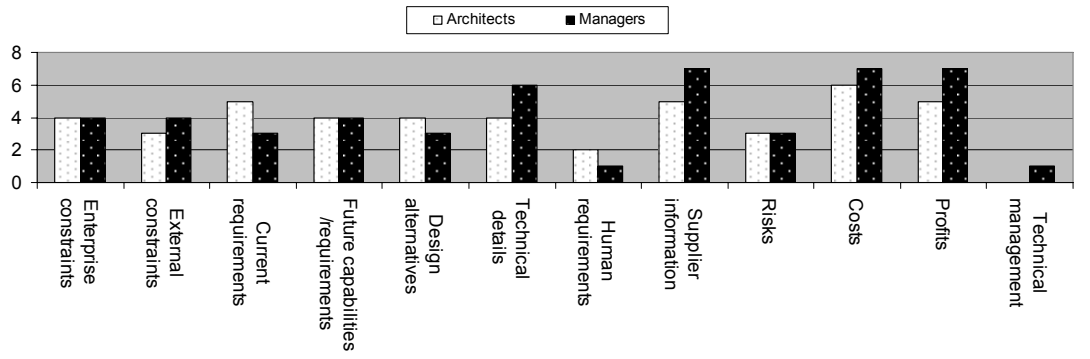

Fig. 2. Differences between the responses from managers and architects.

4.3. Investigated company functions

We have learned from prior studies [1, 2] that large changes in system architecture, such as system refactoring, not only affect the properties of the system but also parts of the company organization that are responsible for a certain task. We therefore mapped each requested information to one or several company functions. The marks in Table 2 show which company functions each respondent has investigated during the interviews.

Table 2 The table shows which company functions each respondent has investigated in the interview.

Company functions	Respondent														
	System Architects (1-7)							*Managers (8-14)*							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	*Sum*
System development	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	14
Manufacturing	*x*		*x*			*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	11
Support/Maintenance	*x*	*x*				*x*	*x*	*x*		*x*	*x*	*x*	*x*	*x*	10
Supplier cooperation		*x*	*x*			*x*		*x*	*x*			*x*	*x*	*x*	8
Sales		*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*		12
Product planning	*x*	*x*	*x*	*x*	*x*		*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	13
Purchase	*x*	*x*	*x*	*x*		*x*		*x*	*x*	*x*	*x*	*x*	*x*	*x*	12

The results indicate that information related to development of the system is the most desired in decision-making. The interest was also large within manufacturing and sales. Issues that concern these company functions are costs, related to hardware and production, and increased profits gained by new market opportunities. System verification is not mentioned at all by any respondent in this study. One explanation might be that system verification is part of system development. To ensure that nothing is left out, a recommendation in a future refactoring guideline is to map expected effects to different company functions or other organization entities.

5. Discussion

We have seen that both managers and system architects are interested in technical details prior to a decision. We have also seen that both managers and system architects investigate profits but do not look much into risks. These “soft factors” are difficult to estimate and usually missed in the existing release planning tools. The decision-maker often uses his own experience to estimate added values. During the interviews many respondents emphasized the importance of the organization’s support when realizing large changes in architecture. They meant both the technical management and especially the understanding amongst co-workers and management on why and how the change would be implemented. It was also important to have well-reasoned arguments and facts when forwarding decisions to higher-level management. For this reason, we do not believe that there is primarily a need for e.g. a computer-aided tool that calculates an answer from given factors. Instead we believe a support package should include guidelines on how to investigate impact on the company organization, in terms of required resources, costs, profits gained etc. and the impact on the system, in terms of system properties, fulfillment of requirements, and so on.

If we compare the results in the study conducted by Wohlin and Aurum [8] and ours, we see that both studies imply that development costs and requirements are the two most important factors in decision-making. However, the results differ in that the respondents are not interested in the technical and

maintenance issues to the same extent. Our study participants were mostly line-managers and system architects whereas some participants in the study of Wohlin and Aurum were project managers, whose goals are often to complete their current project in time. The most probable explanation though, is that we studied changes in the system architecture rather than which features to include and exclude.

6. Conclusion

In this study we learned that the most important information is related to *cost*, *profit*, *technical details*, *supplier information* and *requirements*. Less important are *technical management* and *requirements on humans*. Within embedded systems development, we could not see any significant differences in decision-making between system architects and managers, between specific companies, or between specific industry sectors found within this study. Decision-makers examine how changes in architecture can affect the functions across the company but seem to forget about system verification. *System development*, *manufacturing* and *sale* are the most investigated company functions. From these lessons learned, we can conclude that a guideline for system refactoring should provide the same support regardless of whether the users develop systems for different products. It should provide a structure in order to check effects on the entire company organization. We also highlight, added value estimation and risk calculations, as important ingredients in a guideline of systems refactoring. Further, it should provide advise or best practices of how to increase understanding about the proposed architecture changes amongst upper-managers and amongst co-workers in the organization.

References

[1] Dersten S, Froberg J, Axelsson J, Land R. Analysis of the business effects of software architecture refactoring in an automotive development organization. In *Proceedings of the Software Engineering and Advanced Applications (SEAA)*, 2010 36th EUROMICRO (Lille, France, 1-3 Sept. 2010). IEEE.

[2] Dersten S, Axelsson J, Froberg J. Effect analysis of the introduction of AUTOSAR - an automotive system refactoring study. In *Proceedings of the Software Engineering and Advanced Applications (SEAA)*, 2011 37th EUROMICRO (Oulu, Finland, August 29 - September 2, 2011). IEEE.

[3] Kazman R, Klein M, Clements P. ATAM: Method for architecture evaluation. 2000.

[4] Kazman R, Jai A, Klein M. Quantifying the costs and benefits of architectural decisions. 2001.

[5] Pugh S. Concept selection - A method that works. In *Proceedings of the Konstruktionsmethoden in Uebersicht*: Vortraege der Internationalen Konferenz ueber Konstruieren (Rome, Italy, 1981). HEURISTA, Zurich, Switz.

[6] Saliu O, Ruhe G. Supporting Software Release Planning Decisions for Evolving Systems. In *Proceedings of the Software Engineering Workshop*, 2005. 29th Annual IEEE/NASA (Washington DC, 7-7 April, 2005).

[7] Svahnberg M, Gorschek T, Feldt R, Torkar R., Saleem S. B, Shafique M. U. A systematic review on strategic release planning models. *Information and Software Technology*, 52, 3 (2010), 237-248.

[8] Wohlin C, Aurum A. What is important when deciding to include a software requirement in a project or release? In *Proceedings of the Empirical Software Engineering*, 2005. International Symposium on (17-18 Nov., 2005).