PROJECT MANAGEMENT PATTERNS TO PREVENT SCHEDULE DELAY CAUSED BY REQUIREMENTS CHANGES
Empirical Study on a Successful Project

Shouzo Hori, Takako Nakatani
Yaskawa Information Systems Corporation, Graduate School of Systems Sciences, University of Tsukuba
1-2-3, Manpuku-ji, Asou-ku, Kawasaki, Kanagawa, 3-29-1, Japan,
Otsuka, Bunkyo-ku, Tokyo, Japan

Keiichi Katamine, Naoyasu Ubayashi, Masaaki Hashimoto
Kyushu Institute of Technology,680-4 Kawazu, Iizuka, Fukuoka, Japan

Keywords: PM (Project Management) patterns, Requirements elicitation, Empirical Study, PM knowledge, Framework of PM patterns.

Abstract: We propose PM (Project Management) patterns in order to prevent schedule delays caused by requirements changes on empirical studies. Changes or late elicitation of requirements during the design, coding and test processes are one of the most serious risks, which may delay the project schedules. However, changes and late elicitation of requirements are sometimes inevitably accepted during the development processes. Therefore, the PM method for preventing schedule delays caused by changes and late elicitation of requirements during the development processes should be studied. In this study, we examined the actual conditions of a project. The project succeeded in preventing schedule delays, though it did accept changes and late elicitation of requirements during the development processes. As a result, we were able to extract various typical PM techniques for preventing schedule delays caused by requirements elicitation. The techniques were also applied to other projects. Thus, we call them “PM patterns”. Moreover, we’ve arranged the patterns on a two-dimensional framework. The first dimension is a set of nine knowledge areas of PM such as scope, time and cost management. The second dimension is a group of PM processes such as planning, executing and controlling processes. We also break down the project goal, in this case, the redevelopment of systems for future modifiability, into issues such as keeping lead time and educating engineers, and arrange them on the framework. Then, we discuss the relationship between the project goal and PM patterns on the framework.

1 INTRODUCTION

In most projects, the software requirements are changed and elicited during the design, coding and test phases. Requirements changes and late elicitations throughout the project can be one of the most serious causes of project schedule delays (Johnson, 1995). However, it is often inevitable that we must accept requirements changes and late elicitations after the requirements analysis phase has been completed in order to achieve customer goals (Davis, A. M, 2005).

To clarify how we cope with problematic requirements changes, we have examined the Project Management (PM) techniques of an actual project which was successfully completed within its schedule parameters, even though the project accepted requirements changes and late elicitations after the requirements analysis phase had been completed (Nakatani et al., 2008) (Hori et al., 2008).

As a result, we derived three types of requirements elicitation processes and seven PM patterns useful in preventing schedule delays caused by the requirements changes.

The purpose of this paper is to propose PM patterns that prevent schedule delays caused by requirements changes. Prior to introducing the patterns, we discuss a framework of PM patterns aimed at

Hori S., Nakatani T., Katamine K., Ubayashi N. and Hashimoto M. (2009).
PROJECT MANAGEMENT PATTERNS TO PREVENT SCHEDULE DELAY CAUSED BY REQUIREMENTS CHANGES - Empirical Study on a Successful Project.
In *Proceedings of the 4th International Conference on Software and Data Technologies*, pages 115-120

solving the problems caused by requirements changes.

In this paper, Section 2 describes the three types of requirements elicitation processes and show the results of our empirical study through the observation of requirements change processes. Section 3 describes the framework of PM patterns and, the patterns themselves. Section 4 discusses our research. Section 5 describes our conclusions.

2 EMPIRICAL STUDY

We first give a brief description of the types of requirements elicitation processes. Secondly, we specify the purposes of our study on PM patterns for managing these types of requirements elicitation processes.

2.1 Types of Requirements Elicitation Processes

There are three types of possible requirements elicitation processes, as shown in Figure 1. The vertical axis indicates the requirement elicitation rate of each software component determined by the software architecture of the system. The rate is defined as follows:

(Requirement elicitation rate) = cumReq / allReq *100

where allReq is the total number of requirements elicited for the software component until the end of the project, and cumReq is a cumulative number of requirements elicited until the target elapse date. Each type is as follows:

- E_type (Early maturation type): It completes the requirements elicitation in the early stage of the development. It appeared in the software components to which the existing system had similar ones or reusable ones.
- L_type (Later period maturation type): It represents a continuous process of requirements elicitation throughout the development. This type is observed in the software components, such as the user interface in order to compete with other companies' products.
- U_type (Unforeseen maturation type): It represents the requirements elicitation process that is caused unexpectedly in the later stage of the development unexpectedly. It is observed in the software components that have an interface connected to the outside components developed by third companies.

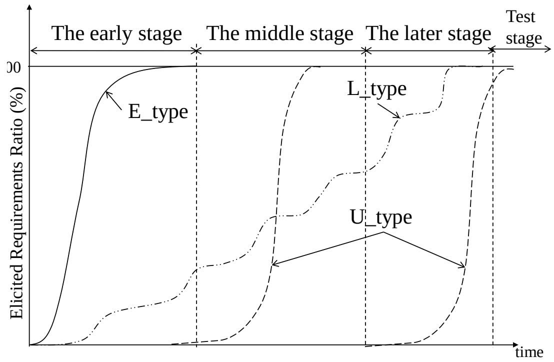

Figure 1: Types f Requirements Elicitation Process.

Of course, the E_type process is desirable, since eliciting all the requirements of the system in the early stage of the development is advantageous. However, the process may become time consuming if there are few skilful engineers. Moreover, the quality of the requirements is doubtful. Therefore, the processes of L_type and U_type were accepted.

We examined the development history of an actual project which reengineered a restaurant ordering system in order to improve its maintainability (Nakatani et al., 2008) (Hori et al., 2008). Figure 2 hows the results of our examination. The horizontal axis of the figure represents the elapsed days of the project.

According to the study, the studied requirements elicitation processes could be categorized into the three types of Figure 1. Then, how to manage these types in real projects?

2.2 Goals of Our Research

The above described E_type requirements elicitation process is desirable. And, although L_type and U_type may pose a risk with regard to causing project delays and disruption of the planned schedule, it is effective to clarify the methods of PM to which the L_type and U_type are applied. We set our goal to extract those PM methods as PM patterns. Concerning the concept of patterns, Alexander shows that although every building is unique, each may be created by following a collection of general patterns (Gamma et al., 1995) (Buschmann et al., 1996). In other words, a pattern is a general solution to a common problem or issue, one from which a specific solution may be derived (Buschmann et al., 1996).

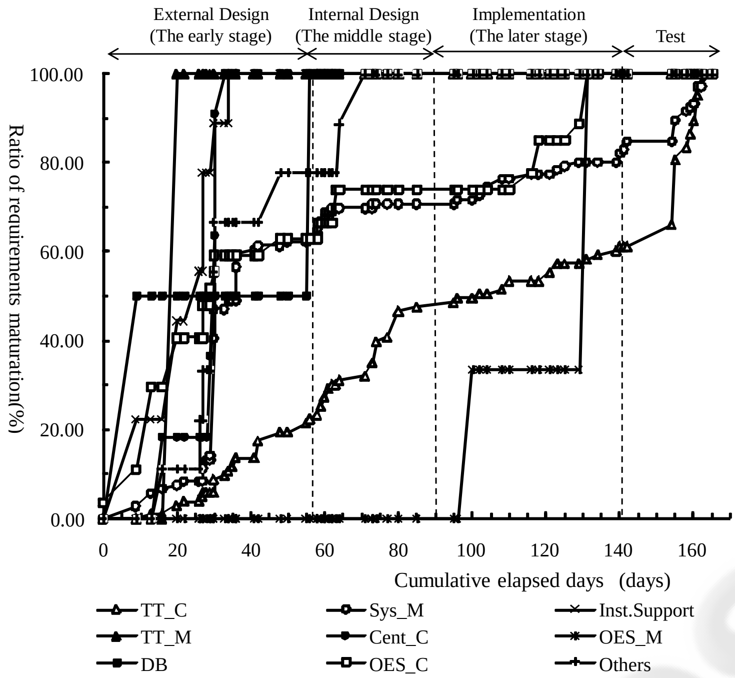

Figure 2: Type of Requirements Elicitation Process.

Before we describe the PM patterns, we show a framework for selecting adequate patterns to apply for a specific situation of the project. In selecting the adequate patterns, the patterns need to be related to the goals of the specific project, such as the reengineering of systems for improving modifiability.

Within the framework, we must define the structure in order to locate any conflict among patterns that are selected and applied to any given project situation. Furthermore, the framework should present a solution for the complications.

3 PM PATTERNS

This section describes a framework of PM patterns. In order to apply the patterns to a project, a manager should decompose their goals. We describe the way to decompose the project goal into the issues in the framework, and then introduce the PM patterns into the framework.

3.1 Framework of PM Patterns

To construct a framework of PM patterns, we propose two dimensions for the framework structure. One dimension is the nine areas of PM knowledge (PMBOK Guide., 2004). The areas are specified in the leftmost column of Table 1. However, in this paper, the Project Procurement Management Area is not specified because the actual example project didn't need procurement. The other dimension is the five groups of PM processes (PMBOK Guide., 2004). The five groups are specified in the right part of the top row of Table 1. However, the Initiating and Closing Processes are not specified because they are not important in this paper.

In order to relate a project goal and the PM patterns with each other, we decompose the project goal into various issues to be solved. We arrange the issues in the nine knowledge areas as specified in the second leftmost column in Table 1.

We also decompose a PM pattern into the solutions intended by the pattern (Buschmann et al., 1996). Further, we also arrange the solutions on the cells of the matrix, which is determined by the two dimensions. The matrix [(specifications are) is specified] in the three columns on the right-side of Table 1. The individual numbers following the solutions in the cells indicate the individual PM patterns described in Subsection 3.3. If the solutions can solve the issues in each of the nine areas, The PM pattern is applicable to the project.

Next, we confirm that there are no implementations contradictions among the PM patterns selected for a project. If two or more PM patterns are related to a solution, we must examine whether the

Table 1: Framework of PM Patterns.

Knowledge Area	Project Issue	Project Management Process Group		
		B) Planning Process Group	C) Executing Process Group	D) Monitoring & Controlling Process Group
I) Project Integration Management				– Integrated Change Management: 2), 3)
II) Project Scope Management	– Variety of function and external interface	– Scope planning & definition: 1), 2), 3) – WBS definition: 1), 2), 3), 5)		– Scope verification & control: 1), 2), 3)
III) Project Time Management	– Short development period	– Project scheduling: 2), 3), 4)		– Schedule control: 2), 3), 4)
IV) Project Cost Management	(– Fund margin of training team)			– Cost control: 2), 3), 4)
V) Project Quality Management	– No detailed system specifications	– Quality planning: 2), 3), 4)	– Quality assurance: 2), 3), 4)	– Quality control: 2), 3), 4)
VI) Project Human Resource Management	– No experience of similar systems – Training maintenance engineer	– Human Resource Planning: 2), 3), 7)	– Project Team Formation: 2), 3), 7) – Project Team Training: 7)	– Project Team Management: 2), 3), 7)
VII) Project Communications Management	– Inefficient communications among subcontractors – No efficient communications path to external interface specifications	– Communication planning: 2), 4), 5), 6)	– Information Distribution: 4), 6)	– Performance Reporting: 4), 5), 6) – Stakeholder Management: 2), 4)
VIII) Project Risk Management	– Keeping delivery date	– Risk response planning: 2), 3), 4)		– Risk monitering & control: 2), 3), 4)

WBS: Work Breakdown Structure

solution or not. If no contradictions are found, all the PM patterns selected are applicable to the project.

3.2 Project Goal Decomposition

In order to select suitable PM patterns for a project, the goal of the project and its specific environment are broken down into the issues to be solved with regard to the project management. We enumerate the goal and specific environment of the actual example project, and its issues in the following (Hori et al., 2008):

Goal: An existing restaurant ordering system which was developed by another company is redeveloped in a short period in order to guarantee the delivery date and, to ensure its maintainability. This goal is broken down into the following issues:

a. Guaranteeing the delivery date
b. No detailed system specifications
c. Short development period
d. Variety of function and external interface

Environmental Constraints: The redevelopment environment constrained the project with the following issues:

e. No experience with similar systems
f. No efficient communications path to external interface specifications
g. Inefficient communications among subcontractors
h. Training maintenance engineer

Environment Margin: The redevelopment environment has the following margin:

a. Fund margin of training team

The above-mentioned issues are described in Column "Project Issue" of Table 1.

3.3 PM Patterns

Seven PM patterns are extracted from the actual example project. The patterns are described as follows:

1) Early Elicitation of Requirements
   - Context: The existing system is redeveloped.
   - Problem: There are a lot of functions to be redeveloped. In addition, the delivery date and cost projection must be guaranteed.
   - Solution: The requirements of the existing components are elicited in the early phase.

– New Problem: If the qualities of the existing components are not good, a problem may occur in the last phase of development.

2) Phased Elicitation of Requirements
- Context: There are a lot of functions to be redeveloped. However, the project members do not have the domain knowledge.
- Problem: If all the requirements are elicited in the early phase, a risk of decreased quality and schedule delay arises.
- Solution: The requirements are elicited gradually. The customers become involved in the elicitation process (Camel et al., 1993) (Beck et al).
- New Problem: If the process of requirements elicitation is not monitored and controlled adequately, a risk of decreased quality and schedule delay arises.

3) Late Elicitation of Requirements
- Context: The system satisfies the customer needs in the market.
- Problem: Changing the requirements in the last phase of development may not be able to guarantee the delivery date.
- Solution: In the last phase of development, skilful engineers analyse the demands of changing requirements, and determine whether they should be accepted or left for the next version.
- New Problem: If the process of requirements elicitation is not monitored and controlled adequately, a risk of decreased quality and schedule delay arises.

4) Rapid Elicitation of Requirements
- Context: The development time is short.
- Problem: The usual requirement elicitation may cause schedule delay.
- Solution: The communications among stakeholders are planned for rapid elicitation of requirements.
- New Problem: If the process of requirements elicitation is not monitored and controlled adequately, a risk of decreased quality and schedule delay arises.

5) Elicitation of External Interface Specifications
- Context: The system is connected to a product developed by another company.
- Problem: There is no information of the product specifications among the development team. Moreover, there is no efficient communications path to obtain the specifications.
- Solution: The development team participates early in open meetings concerning the product.
- New Problem: If the specifications obtained in the open meetings are insufficient, a problem of interface mismatching arises in the integration test.

6) Communications among Subcontractors
- Context: Two or more subcontractors work together to develop the system.
- Problem: The communications among the subcontractors are inefficient. Therefore, they may cause a schedule delay.
- Solution: The subcontractors gather information in a repository.
- New Problem: A security problem of the information repository may occur.

7) Training of Software Maintenance Engineers
- Context: Software maintenance engineers are needed to extend the system often in a short period of time.
- Problem: There are no software engineers who possess the knowledge of the system requirements.
- Solution: Software maintenance engineers are trained from the early phase.
- New Problem: The cost of training is incurred.

In Table 1, each number of the above-mentioned PM patterns is specified on the side of the PM process to which the implementation of the PM pattern is related. Using the framework specified in Table 1, we can conclude that the problematic issues of the project are solved by the PM patterns, and the PM patterns have no implementation contradictions among them.

4 DISCUSSION

The PM patterns described in Subsection 3.3 were found in projects other than the actual example project. Some of the projects were successfully completed. Therefore, the PM patterns are useful, and will be applied in the future. However, some of the projects to which the PM patterns were applied, such as the Phased and Late Elicitation of Requirements, ultimately failed. One of the main causes of failure was an unanticipated large quantity of requirements to be obtained by the phased and late elicitation. These kinds of failures can be found in projects developing embedded software. In such projects, the hardware and software are often concurrently developed. If it is decided that a problem within the hardware is to be solved through the software, an unanticipated large quantity of software requirements arises. This may delay the projects projected schedule. To prevent such schedule delay, the condition in which we apply the PM patterns should be made clear and, proven to be satisfactory before starting the project.

Although the PM patterns are useful, non-experts of PM can not apply the patterns to actual projects by themselves. The non-experts cannot conceive the whole picture of actual project management. Therefore, it would be difficult for them to apply the PM patterns to actual projects. The PM patterns would be useful for experts of PM in order to prevent the mistakes of PM. Moreover, they would be useful for the experts to teach and lead non-experts.

In this paper, a kind of whole picture of actual PM is given by the framework described in Subsection 3.1. After PM patterns are selected in order that they may be applied to a project, the consistency among them can be approved within each of the PM processes specified in the framework. Moreover, the consistency between the PM patterns and other parts of PM can also be approved in the same way. Thus, the consistency of approving is restricted to each of the PM processes in order to make it practical.

In the future, PM patterns other than the ones described in this paper, which are aimed at preventing schedule delay, which are caused by requirements elicitation, should be extracted. Furthermore, the PM patterns should be systematized. The framework of PM patterns is one of the most important issues in the technology of PM patterns. Therefore, the framework should be studied in order to make the application of PM patterns more practical.

5 CONCLUSIONS

In this paper, we proposed seven PM patterns in order to prevent schedule delay caused by requirements elicitation. Moreover, we proposed a framework for approving project goal implementation through the PM patterns, and non implementation contradiction among the PM patterns.

In the future, other PM patterns should be extracted and systematized. Moreover, the framework should be studied in order to make the application of PM patterns more practical.

ACKNOWLEDGEMENTS

This work was funded by the Joint Forum for Strategic Software Research (SSR). We would like to thank our colleagues at SSR. We would also like to thank Mr. Masao Shimoji and his colleagues at the Yaskawa Information Systems Corporation who cooperated on our study as interviewees.

REFERENCES

Nakatani, T., Hori, S., Ubayashi, N., Katamine, K., and Hashimoto, M. (2008). A case study: Requirements icitation processes throughout a project. In *Proc. of the 16th International Requirements Engineering Conference (RE'08)*, pages 241–246. IEEE.

Hori, S., Nakatani, T., Ubayashi, N., Katamine. K., and Hashimoto, M.(2008). Towards risks avoidance by eficient requirements elicitation. In *Proc. of the Society of Project Management (in Japanese).* pages 470-475.

Johnson, J.(1995). Chaos: the dollar drain of IT project failures. *Application Development Trends*, 2(1), pages 41-47.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal, M. (1996).; *Pattern-Oriented Software Architecture. A System of Patterns*, John Wiley & Sons.

Carnel, E.,Whitaker, R.,and George, J. (1993).: PD and joint application design. a transatlantic comparison, *CACM*, 36(4), pages 40-47. ACM.

Project Management Institute. (2004). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) Third Edition*, PMI .

Davis, A. M. (2005): *Just Enough Requirements Management : Where Software Development Meets Marketing*, Dorset House.

Beck, K., et al. (2001). Manifesto for agile software development, http://agilemanifesto.org/.