# A Case Study on Optimization of Resource Distribution to cope with Unanticipated Changes in Requirements

Joost Noppen
*Trese Group*
*Dept. of Computer Science*
*University of Twente*
*P.O. Box 217, 7500 AE Enschede*
*The Netherlands*
*noppen@cs.utwente.nl*

Mehmet Aksit
*Trese Group*
*Dept. of Computer Science*
*University of Twente*
*P.O. Box 217, 7500 AE Enschede*
*The Netherlands*
*aksit@cs.utwente.nl*

## Abstract

*It is a known fact that requirements change continuously, and as a consequence, it may be necessary to reschedule development activities so that the new requirements can be addressed in a cost-effective manner. Unfortunately, changes in requirements cannot be specified precisely. Moreover, current software development methods do not provide explicit means to adapt development processes with respect to unanticipated changes in requirements. This article first proposes a method based on Markov Decision Theory, which determines the estimated optimal development schedule with respect to probabilistic product demands and resource constraints. Second, a tool is described that is built to support the method. Finally, some experimental results are presented on the applicability of the proposed method.*

## 1. Introduction

Requirements evolve continuously during the lifetime of a software system. Unfortunately, several case studies have shown that a major portion of costs of software systems is devoted to changes in requirements [1]. One may deal with changes in requirements, for example, by designing adaptable software architecture using the concepts of object-oriented application frameworks. Although developing adaptable software systems may help to a certain degree, it may not be sufficient; if requirements change during the course of a software development activity, it may be necessary to reschedule development activities so that the new requirements can be addressed in a timely manner.

This article addresses this particular problem and as such proposes an approach for optimum distribution of human resources when facing non-deterministic changes in requirements. The approach is based on the Markov Decision Model, which constructs probabilistic state-transition-diagrams that represent market change scenarios. By estimating changes in market demands, available human resources can be distributed optimally at certain points in time.

The remainder of this article is organized as follows. Section 2 describes the problems that are addressed in this article. Section 3 explains the proposed method by applying it to the example case. Section 4 describes the tool that supports this method. Section 5 explains the case study that we have carried out for evaluating the method. Finally, section 6 and 7 conclude the article and describe the ongoing work.

## 2. Problem Statement

In this paper we assume that to prepare for the changes in requirements at most care is taken for the software design. Currently, application framework technology provides one of the most flexible architectures. As a case study we have selected the design of software for insurance systems. In the following, we will present two problems when dealing with evolution of the insurance software systems demands.

**Difficulty in prioritizing requirements**

Assume that a large set of components have to be implemented in a time span of 2 to 4 years. These components will be used to create several different versions of insurance systems and they must be ready for delivery whenever they are demanded. The components and systems share dependency relations among each other. Therefore, it is crucial to deliver the essential components first, before the customers demand systems that incorporate specific features. This requires giving higher priorities to the components that are demanded first.

In the literature, various techniques have been proposed for prioritizing requirements [2]. These techniques, however, do not provide explicit means for determining the priority values with respect to the changes in demands.

Several publications and tools are proposed for supporting decision-making processes. Most of the approaches are based on the Analytic Hierarchy Process (AHP) method [3] [4] [5]. However, the proposed approaches assume a fixed set of requirements and therefore they are not directly suitable for dealing with software evolution.

**Configuring software development processes for delivering products in time**

After prioritising the components, the project managers should be able to configure the implementation trajectory accordingly. Since there may be many options for the implementation effort, preferably, ideally the project managers should be able to compare all the relevant options and select the best configuration that fulfils the timing and resource constraints. For the example case, this means that the expected optimal planning of resources for a given time span should be determined to maximise profit in selling software systems.

Configuring software processes with respect to available resources is not new [6]. However, in contrast to existing methods, also the demands for future systems and changes in requirements must be considered.

## 3. Approach

The options in scheduling software development processes depend on the possible software system demands. The following three steps are used to determine the most profitable option:

1. Determine probabilistic estimations for market, cost and profit changes;
2. Specify the possible market scenarios that can occur;
3. Evaluate all possible schedules with respect to the scenarios and select the most cost effective schedule.

This method should be flexible enough to cope with changes in the estimations at any given moment in time. To achieve the objectives stated in the previous section, the problem needs to be formalized mathematically. In our approach the optimization problem is formalized using Markov Decision Theory [7].

## 4. Tool Support

The optimisation process is very labour intensive and therefore automatic support is necessary. In addition, tool support is necessary for modelling artefacts, market, and available resources. The architecture of our tool is shown in Figure 5. Here, the models and processes are represented as rectangles and ellipses, respectively.

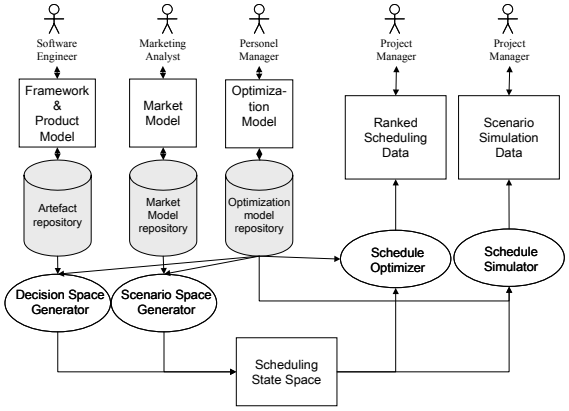


*Figure 1. The tool architecture*

Artefacts (framework parts and software systems), software system demand scenarios and resources are modeled by the software engineer, marketing analyst and personnel manager, respectively. The Personnel Manager also defines the goals of the optimization problem, such as minimization of cost, maximization of profit, etc.

The process *Decision Space Generator* retrieves the necessary information from the artefact repository and the process *Scenario Space Generator* retrieves the necessary information from the Market Model Repository. Together they generate the data *Scheduling State Space*. The next step is the determination of scheduling advice. The process *Schedule Optimizer* takes *Scheduling State* space as

input and generates the data *Ranked Scheduling Advice* as output. This data can be presented to the project manager in various formats, for example, as a table shown in Table 3. In addition the project manager can run simulations by using the *Schedule Simulator, which* results in the data *Scenario Simulation Data*.

## 5. Case Study

The success of our approach depends on the, estimations of the cost of resources, the flexibility of the software design process in adapting to the changes in requirements and accurate estimations of the market demands. We assume that it is possible to make reasonable estimations of the cost of resources. The flexibility of the software design process can be achieved by adopting agile software development methods. In addition, we used object and component based application frameworks for creating highly tailorable software architecture. To determine how successful we are in estimating the market demands, we have carried out a case study.

This case study consists of four steps: the first step is the implementation of the tool described in section 4. Second is the identification of market estimations at a predetermined point in time, which will be used for the optimization. The third step is verification of the results of the simulation with the actual data from the years for which predictions have been made. Finally the effectiveness of the approach is evaluated with the acquired results. This case study is aimed at clarifying the following issues: can we extract realistic market expectations from statistical data and is the approach suitable for realistic cases.

After the tool had been implemented, a group of students was asked to make market estimations as if they were in the year 1994. To understand the application domain, the students have also received a feature diagram which represents the domain of insurance products, as shown in Figure 2:

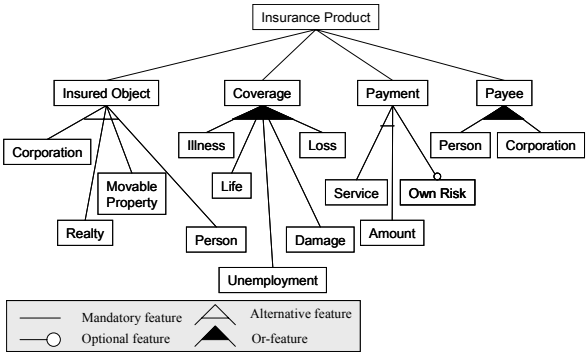


*Figure 2. Insurance Framework Feature Diagram*

A feature diagram represents both the commonality and variability of a product [8]. In figure 2, four different parts of insurance products are shown: *Insured Object*, *Coverage*, *Payment* and *Payee*. Each of these parts can be represented by several entities. For example, an insured object can be a *Person* or a *Corporation*. The symbols that are depicted in the legend are used as restrictions on the variability. For instance, an insured object cannot be a corporation and a person at the same time.

The results of the market estimations were analysed so that the framework architecture could be adjusted accordingly. After thorough analysis, three scenarios were selected as important business requirements, and such should be considered during the implementation of the framework.

***Scenario 1: Damage Insurance Increase***

After weeks of rain and early rises in temperature in the Alps, the Dutch rivers have exceeded their capacity. The result is a flooding of a large number of cities, leading to increased damages. As a result many people want to upgrade their damage insurance policy to cover flooding damages. The probability of this scenario occurring is estimated at 5 %.

***Scenario 2: Lease car damages***

In 1995 the amount of lease cars will increase dramatically. Since a lease car is no longer private property, lease cars will not be treated with the same caution. This will lead to a substantial increase in damages in this specific group of car users. It is expected that insurance companies will design a tailor made insurance for lease cars. The probability for this scenario is estimated at 40 %.

***Scenario 3: Medical Facilities***

Due to the increasing average age in Holland the costs for usage of medical facilities will increase. In the context of economic growth and the slight grow in population a slight increase in Medical Facilities Insurances is expected. The probability for this scenario is estimated at 55 %.

These scenarios, together with the feature diagram, have been entered into the tool to determine the optimal implementation policy. In addition, additional information was provided, such as availability of resources and profit calculation functions. This has resulted in development policies that focus on the parts of the architecture which will most likely maximize profit. Currently we are in the process of verifying the results of the case study with the actual data. In the

near future, the approach will be evaluated with these results.

## 6. Evaluation

In section 2, prioritizing changing requirements and configuring software development process accordingly were identified as two important problems. In the following, we will evaluate our approach with respect to the following concerns:

***Identification of Market Scenarios***

The identification of relevant market scenarios proved to be a very difficult task. The available statistical information provided valuable input, however it was seldom available in a readily usable form. The information needed to be transformed in order to be usable for the tool. These transformations are quite complex since they involve explicit interpretation of statistical information. For this particular activity the knowledge of domain experts, combined with statistical experts, could greatly help in maintaining the accuracy of the statistical data.

***Scalability of the approach through managing state explosions***

To determine the optimal result for Markov Decision Models, all relevant states are needed. This can cause very large state spaces, also known as the *Curse of Dimensionality* [9]. This was also experienced during the case study. To address this problem, within the field of Markov Decision Theory, we are currently experimenting on various state-space reduction mechanisms.

***Human Resource Model***

In our approach it is assumed that every employee can perform any task, if the time permitting. A more realistic approach would be to categorize human resources with respect to their skills. This adds an extra level of dependencies that needs to be considered when scheduling human resources. We are currently working on a more realistic human resource model, which improves the model properties based on the experiences obtained in the past.

## 7. Conclusion

Conventional software development methods and tools do not provide support in minimizing the cost of rescheduling of software development activities due to unanticipated changes in requirements. To overcome this problem, we have proposed a method and a tool, which represents the possible development activities as a state space using Markov decision theory. The developed tool supports the modeling, the selection and the evaluation of the changes to requirements and the allocation of human resources. The approach and the related tool explicitly enable the designers to think about the priorities and risks of decision making with respect to changes in requirements.

We have set up a case study in order to validate our approach. By determining market estimations for a period that is set in the past, we hope to gain a better insight to verify the applicability of our approach. However, this is still an ongoing activity. At this point the market estimations have been made, and the optimizations have been determined. Future work will be the evaluation of the approach by using actual data of the estimated period.

## 8. References


[1] Kniesel, G. Noppen, J. Mens, T. Buckley J. *The first international workshop on unanticipated software evolution* Workshop report for the first int. workshop on unanticipated software evolution at ECOOP 2002

[2] Jacobson, I. Booch, G. Rumbaugh, J. *The Unified Software Development Process* Addison Wesley, 1999, ISBN: 0-201-57169-2

[3] Karlsson, J. Ryan, K. *Prioritizing Requirements Using a Cost-Value Approach* IEEE Software September/October issue, pp. 67-74.

[4] Salo, A. Hämäläinen, R. *On the measurement of preferences in the analytic hierarchy process* Journal of multi-criteria decision analysis, vol. 6, 309-319, 1997

[5] Forman, E. Sally, M. *Decision by Objectives* World Scientific, 2001, ISBN: 981-02-4142-9

[6] Podorozhny, R. Staudt Lerner, B. Osterweil, L. *A rigorous approach to resource management in activity coordination* University of Massachusetts, Amherst MA 01003, USA, 1999

[7] Puterman, Martin, *Markov decision processes, discrete stochastic dynamic programming*, 1994, Wiley-Interscience, ISBN: 0-471-61977-9

[8] Kang, K. Cohen, S. Hess, J. Novak, W. Peterson, A. *Feature Oriented Domain Analisys (FODA) Feasability Study* CMU/SEI-90-TR-21 ESD-90-TR-222, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, November 1990

[9] Bellman, R. *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, 1961.