

Defining the Process for Making Software System Modernization Decisions

Jarmo J. Ahonen^{1,*}, Henna Sivula², Jussi Koskinen², Heikki Lintinen², Tero Tilus²,
Irja Kankaanpää², and Päivi Juutilainen²

¹ Department of Computer Science,
University of Kuopio, P.O. Box 1627, FI-70211 Kuopio, Finland
jarmo.ahonen@uku.fi

² Information Technology Research Institute,
P.O. Box 35 (Agora) FI-40014 University of Jyväskylä, Finland
{henna.sivula, jussi.koskinen, heikki.lintinen, tero.tilus,
irja.kankaanpaa, paivi.juutilainen}@titu.jyu.fi

Abstract. This paper outlines a process for software system modernization decisions. The rationale of the process is explained and the process is defined in a way that allows its adaptation for other organizations and situations. The process is a light-weight one and is based on the use of objective data. The procedures for collecting the data are explained. The process has been used to solve a real industrial decision making situation in which the process was successful.

1 Introduction

Many large software systems, whether tailored or not, are nearing or have reached an age which makes it necessary to decide what to do with them. The alternatives are normally to do nothing really new — to continue with normal maintenance as before — reengineer the system, modernize the system, or replace the system. For large software-oriented service provider or user organizations those decisions may be so common that the process of making such decisions should be documented and guidelines for decisions should be provided. The making of those decisions should be implemented as a well-thought part of the collection of software engineering processes of the organization.

The reason why those decisions should be seriously considered is that the decision whether to modernize an old system is a remarkable one in the case of a business critical system. Major modernizations create several risks to the user organization. Those risks include the possible bugginess, potential misunderstandings of the previously implemented business-critical knowledge, delays, and many other issues that may have negative impacts to the business of the user organization. Therefore the economic impacts of the decisions can be remarkably higher than the actual software engineering costs caused by the decision. In the case of information technology infrastructure outsourcing or major long-time agreements those economic impacts should be taken into account by the service provider organizations' software engineering process.

* Corresponding author.

One of the issues that should be taken into account is the fact that the life-cycle costs of an information system tend to have 50–70 % of maintenance costs [1] — those costs may be even higher [2][3]. That level of costs is not a surprise considering the remarkable amount of effort already invested into an old legacy system. It is, however, the case that an information system has to evolve in order to maintain its usability in the changing world and that as a consequence of the evolution the system becomes more and more complex and difficult to manage [4]. For every system there will be a day when the user organization and the service provider organization have to decide what to do with the system.

The cost of the modernization effort depends greatly on the complexity of the system [5]. Although complexity and potential costs are major factors in the modernization and replacement decisions [6], the actual need for the change should be considered. The systems' ability to perform its assigned task is one of the most important decision factors. The suitability of the system for the processes can be estimated by analyzing the business value of the system and the exceptions and other types of problems encountered in the normal use of the system. Exceptions are real-world cases which have been handled in order to perform the business process but which cannot be handled by using the system in question [7].

The most common software process improvement frameworks do not provide required answers for an organization that wants to improve its decision processes regarding old software systems. At least CMMI [8] and ISO [9] reference models do not include necessary information. There are some decision frameworks proposed. The most promising ones are the planning framework for reengineering proposed by Sneed [10] and the decisional framework for legacy system management proposed by De Lucia, Fasolino and Pompella [11]. Both of those frameworks stress that the business value of the existing software system should be evaluated and the ability of that system to serve the business should be analyzed.

Neither of those frameworks do, however, provide a sufficiently detailed description of the actual decision making process. Therefore an organization faced with such decisions has to develop and define a process of its own. In this article such a process is defined and tested with a real case. The structure of this article is: the rationale and the requirements set for the process and the definition of the process in Section 2 and outline of the industrial setting in which the process has been used for a test-run in Section 3, an overview of the data collected during the test-run (Section 4), a brief analysis of the process and its applicability in the real-world (Section 5), and discussion on the results and on the general importance of performing studies like the reported one (Section 6).

2 Defining the Process for Making Software System Modernization Decisions

The available models for the modernization decision process include the sub-process of finding out the business value of the analyzed system. Sneed's model [10] proposes the use of a business value evaluation table and De Lucia et al [11] propose a fairly similar approach to be used. Both of those approaches assume that the business value