

Chapter 71

A Model to Assist the Maintenance vs. Replacement Decision in Information Systems

O. Tolga Pusatli

Cankaya University, Turkey

Brian Regan

University of Newcastle, Australia

ABSTRACT

An extensive literature review shows that Information Systems (IS) are changed and eventually replaced by substitutes under the influence of productivity, popularity, and specialisation of IS along with associated available support facilities, maintenance activities, failures, and user feedback. This chapter compiles those factors emerged from software engineering, IS, software quality assurance measurements, and computer science literature. A final product is a proposed model bringing those factors together as they are suspected to raise a need for taking the decision of evaluating change requests that may lead to a further maintenance or replace the IS. While keeping in mind that limitations on advanced testing exist, an expected service of such a model may help decision makers to explain maintenance/replacement decision of IS/component in a more itemized manner, hence diminish overburden pressure of experience responsibility on them.

INTRODUCTION AND BACKGROUND

While much research and discussion has occurred around the best processes for developing new information systems (IS), there has been less analysis of the dynamics of making the important decision of whether to move from the maintenance mode for an existing system into developing or purchasing

a new replacement software system. Throughout their lifecycles, IS evolve with continual maintenance, which fixes errors and extends functionality, but eventually most systems become unstable and unsustainable. Making the call on when that point has been reached is an important and challenging decision for many managers. The main objective of this chapter is to propose a model that may be useful to explain a path leading to a decision point of whether the IS/component should be replaced or

DOI: 10.4018/978-1-4666-4301-7.ch071

further maintained. While we investigate literature on IS including technology acceptance, success models, user feedback, software functionality, bespoke versus off-the-shelf systems, maturity models, life cycle analyses, software evolution measurement and quality assurance, we find it beneficial to mention that the objective of this chapter excludes social, economical and accounting associated factors.

Under the influence of factors such as changes in functional requirements, external constraints and new hardware environments, software systems or components are finally abandoned and a substitute is usually sought. Heales (2002) contrasts “evolution” brought by maintenance activities such as updates, error corrections and changes to improve functionality with “revolution”, the realisation that a system needs to be replaced with implications of associated upheaval. In Heales’ language, we are interested in the decision whether to evolve or to go to revolution.

In practice, decisions to maintain or replace systems are often based on a set of guidelines tailored to an organisation’s needs through system reviews and feasibility studies (Horch, 2003). The decision to replace or maintain IS or major components is presented in the literature as an organisational decision, made by some group of stakeholders. An early model of when to replace rather than maintain a system was provided in Gupta and Raghunathan (1988), based on relative costs associated with maintenance and replacement. Other factors such as availability of skills and business goals were noted but not incorporated in that model. Pragmatically, the vision or otherwise of top management leadership and organisational politics also play a role in the replacement decision, as reported in a case study (Rada & Finley, 2004) on whether or not to maintain legacy IS.

Part of the problem for management however is the difficulty in predicting life expectancy of IS so as to plan for replacement, according to (Glass, 1998). That is, it is difficult to predict when maintenance will not be enough and replacement will

be needed, although these authors say generally that as the effort involved to maintain the system increases, time for replacement approaches. Another example is reported by Salmeron and Lopez (2010), where maintenance is an “obligation” in ERP systems despite such a task is risky for its complexity and cost. Because an organisation’s IS can be composed of subsystems such as revenue, administration, e-mail and operations support, the decision to replace whole systems is not only difficult to make, but it also has significantly higher risk when compared to replacing a portion. New releases hence usually concentrate on improving a small number of subsystems (Sommerville, 2004).

One of the “laws of program evolution” (Meir Manny Lehman, Ramil, Wernick, Perry, & Turski, 1997) is that, as systems evolve, their complexities increase unless work is done to maintain them. All software systems reach a state of internal interaction such that any effort to fix one error is expected to introduce more new errors, with a resulting cascade of system failures and errors (Yourdon, 1989). Thus, the quality of systems will decline until they are eventually replaced. Addressing this issue, we can recall Heales’s research (Heales, 2002), where the term “volatility index” has been coined to describe the increasing instability of IS as they approach the stage when evolution (being maintainable) is no longer practicable and they are replaced. The higher the volatility index is, the higher the probability is that systems have to be replaced.

An early study put by Gupta and Raghunathan (1988) investigating costs associated with maintenance and replacement concluded that as the number of change requests rose, the organisation should consider replacement, rather than evolution. Increased modularity of a system increases its ability to adapt to external changes and thus extends its life span. Similarly, the flexibility of a system to adapt to changing data formats used in an industry sector has been shown to increase its longevity, by the definition given in (IEEE, 1990) (flexibility or adaptability).

18 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/model-assist-maintenance-replacement-decision/77766

Related Content

Design and Evaluation of Automated Scoring: Java Programming Assignments

Yuki Akahane, Hiroki Kitaya and Ushio Inoue (2015). *International Journal of Software Innovation* (pp. 18-32).

www.irma-international.org/article/design-and-evaluation-of-automated-scoring/133112

The Effect of Online Service Retailers' Quality Gaps on Customer Satisfaction

Asem Majed Othman, Vincent Omachonu and Emad Hashiem Abualsauod (2017). *International Journal of Systems and Service-Oriented Engineering* (pp. 21-44).

www.irma-international.org/article/the-effect-of-online-service-retailers-quality-gaps-on-customer-satisfaction/188593

Mouth Shape Detection Based on Template Matching and Optical Flow for Machine Lip Reading

Tsuyoshi Miyazaki, Toyoshiro Nakashima and Naohiro Ishii (2013). *International Journal of Software Innovation* (pp. 14-25).

www.irma-international.org/article/mouth-shape-detection-based-template/77615

Certifying Properties of Programs Using Theorem Provers

J. Santiago Jorge, Victor M. Gulias and David Cabrero (2007). *Verification, Validation and Testing in Software Engineering* (pp. 252-301).

www.irma-international.org/chapter/certifying-properties-programs-using-theorem/30754

Performance Analysis of FPGA Architectures based Embedded Control Applications

Slim Ben Othman, Ahmed Karim Ben Saleem and Slim Ben Saoud (2011). *Reconfigurable Embedded Control Systems: Applications for Flexibility and Agility* (pp. 274-310).

www.irma-international.org/chapter/performance-analysis-fpga-architectures-based/50433