

Maintainability Based Risk Assessment in Adaptive Maintenance Context¹

W. Abdelmoez², K. Goseva-Popstojanova, H.H. Ammar

Lane Department of Computer Science and Electrical Engineering,

West Virginia University

Morgantown WV 26506

{rabie, katerina, [ammar](mailto:ammar@csee.wvu.edu)}@csee.wvu.edu

Abstract

Development of software systems utilizes only 20% - 40% of the overall project cost; the rest is consumed by maintenance. Systems with poor maintainability are difficult to modify and to maintain. Maintainability-based risk is defined as a product of two factors: the probability of carrying out maintenance tasks and the impact of these tasks. In this paper, we present a methodology for assessing maintainability-based risk in the context of adaptive maintenance. We demonstrate the methodology on a case study using UML models.

Keywords: maintainability-based risk, adaptive maintenance, software architectures.

1. Introduction

A successful maintenance project necessitates a well-planned maintenance effort to manage the maintenance process and to alleviate the risks associated with performing unnecessary maintenance tasks. Software maintainers usually are not engaged in the initial software development cycle. Before maintainers can modify a program, they must understand how it operates. They often deal with complicated and hard to comprehend systems. The condition of system documentation, the experience and skillfulness of the programmers, and the characteristics of the system itself are some of the factors that influence the maintenance progress [13].

Many types of risk are associated with the maintenance of software systems. These types of risk are project risk, usability risk and maintainability risk [14]. Project risk concern is that the maintenance project cannot be completed within the budget or timeframe due to unproductive maintenance process or deficiency of personnel and maintenance resources. While the focus of usability risk is that the maintenance conducted on the system will trigger problems and failures. Usability risk takes into account the functionality, performance, and software failure risk. Maintainability risk addresses the question how complex it will be to maintain the system in the future because of the way we handled this maintenance task.

Maintainability-based risk can be used to enhance the system architecture maintainability, to pinpoint risky components in terms of maintainability or to manage the process of system maintenance. In accordance with NASA-STD-8719 standard [11], we define maintainability-based risk as a combination of two factors: the probability of performing maintenance tasks and the effect of conducting these required tasks. Accordingly, Maintainability-based Risk for a component is defined as [1]:

*Probability of changing the component**
Maintenance impact of changing the component.

In [2], we have developed a methodology to provide the maintainer with an estimate of the maintainability-based risk of components due to changes in requirements. Our methodology allows the maintainer to identify risky components and also risky change scenarios. Furthermore, the maintainer

¹ This work is supported by the National Science Foundation through ITR program grant No CCR 0296082, and by NASA through a grant from the NASA Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program (SARP) managed through the NASA Independent Verification and Validation (IV&V) Facility, Fairmont, West Virginia.

² Correspondence author. E-mail: rabie@csee.wvu.edu

can manage the maintenance process by making a trade off between the risk associated with maintenance tasks estimated by the methodology and the anticipated added value.

The methodology for estimating maintainability-based risk presented in [2] depends on architectural artifacts such as system requirements and system design and their progression through the life cycle of the system. First, we estimated the requirements maturity by analyzing their evolution across the releases of the system. Then, we mapped the requirements stability into components stability, which reflected the likelihood of making changes to components due to changes in the requirements. Consequently, we estimated *Initial Change Probabilities ICP* of the system components. Using the initial change probabilities of the components and *Change Propagation CP* probabilities between them, we get the unconditional probability of change of the components of the system. To get the impact of the maintenance tasks, we use the *Size of Change SC* between the components of the system. Finally, the maintainability-based component risk factor is the product of unconditional change probability and the *Maintenance Impact MI*.

In this paper, we adapt the methodology for estimating the maintainability-based risk of software components due to adaptive maintenance tasks. We use the change reports of the adaptive maintenance for the project to estimate the initial change probabilities *ICP* for the system components. According to IEEE standard for software maintenance [8], adaptive maintenance is defined as “*Modification of a software product performed after delivery to keep a computer program usable in a changed or changing environment*”. Estimating maintainability-based risk for adaptive maintenance can be considered as a predictive model that can be used to predict the effort required for maintaining software systems as requirements and environment change.

The estimation procedure of maintainability-based risk builds on our previous work on change propagation probabilities [3] and size of change [2]. To estimate these metrics, we first analyze the architecture of the system under investigation using a structural diagram or a class diagram. From these artifacts, we identify the components and the

connectors of the component-based system architecture. Then, we analyze message protocols between every pair of components in the system to identify messages exchanged between components C_i and C_j . With the help of case tools, we get message sets for any pair of components in the system. This information can also be obtained from static analysis tools of the source code.

The paper is organized as follows. In Section 2, we present the estimation methodology for maintainability-based risk in the context of adaptive maintenance. In Section 3, we illustrate how to estimate maintainability-based risk on a case study. In Section 4, we discuss related work. We conclude the paper and discuss the future work in Section 5.

2. Maintainability-Based Risk Assessment in Adaptive Maintenance Context

In this paper, we will limit our scope of maintenance effort to adaptive maintenance. Thus, we alter the methodology for estimating the maintainability-based risk of software components presented in [2] as follows. Basically, we make use of adaptive maintenance reports of changes to estimate the initial change probabilities $ICP=[icp_i]$. First, we evaluate the rate of occurrence of changes in each component C_i of the system. Then, we estimate the initial probability of change for each component by normalizing the rate of occurrence for each component by the total number of change reports. Hence, the estimation methodology of maintainability-based risk is tailored for adaptive maintenance, as shown in Figure 1.

To take into consideration the dependency between the components of the system, we multiply the initial change probabilities vector *ICP* of the components by the conditional change propagation probabilities matrix *CP* obtained from the system architecture. The *Change propagation probability CP* = $[cp_{ij}]$ for an architecture is the conditional probability that a change originating in component C_i requires changes to be made to component C_j [3]. Thus, we calculate the unconditional probability *UCP* of change of each component of the system:

$$UCP = ICP * CP \quad (1)$$

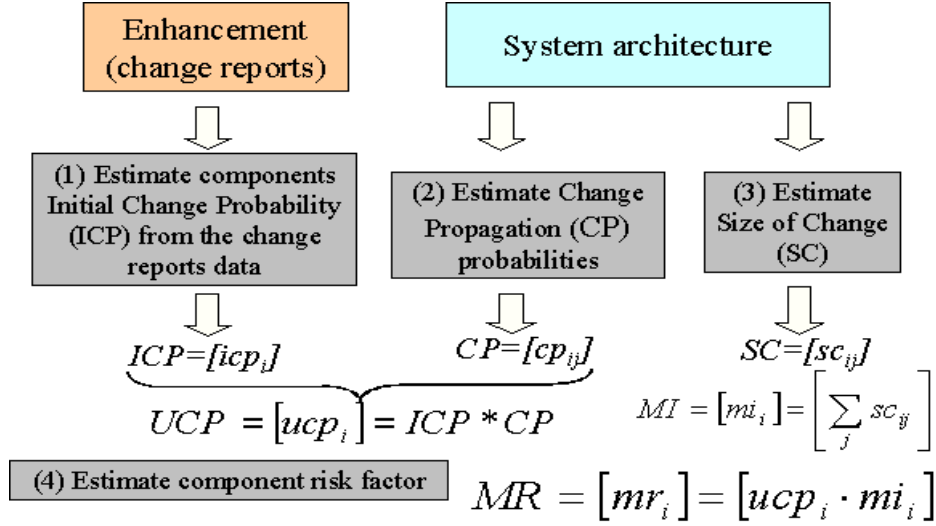


Figure 1 Maintainability-based risk estimation methodology for adaptive maintenance

Then, The *Maintenance Impact* $MI=[mi_i]$ of the change in component C_i on the rest of the components of the system is predicted as:

$$mi_i = \sum_j sc_{ij} \quad (2)$$

where sc_{ij} is the size of change of component C_i due to changes in the interface elements of component C_j based on the architecture artifacts. Each element of the Size of Change $SC=[sc_{ij}]$ matrix is defined as the ratio between the number of affected methods of the receiving component caused by the changes in the interface elements of the providing components and the total number of methods in the receiving component [2]. Finally, the maintainability-based risk of a component C_i due to adaptive maintenance changes mr_i is given by

$$MR=[mr_i]=[ucp_i \cdot mi_i] \quad (3)$$

We propose to use the maintainability-based risk of the system components to order the adaptive maintenance tasks for a certain project.

3. Case Study

The maintainability-based risks are evaluated for the components of the CM1 case study from the Metrics Data Program [9]. The Metrics Data Program is a database that contains data about problems, products and metrics of a number of software projects. The main objective of the program is to gather, validate, arrange, save and provide software metrics data for the software engineering community. The case study CM1 is a software component of a data processing unit used in an instrument, which gather data to probe the universe. A UML model [15]

for CM1 is constructed from the artifacts provided. In this section, we present the results of the maintainability-based risk for the CM1 case study. Then, we discuss the results of the maintainability-based risk.

The maintenance data of the CM1 case study contain 31 change reports. We want to prioritize the tasks of the adaptive maintenance effort. First, we calculate the frequency of requested change occurrences in the components of the system. Second, we estimate the initial change probability ICP of the components of CM1 by normalizing the frequency of change occurrences by the total number of change reports. The estimated initial change probabilities ICP for CM1 components are shown in Figure 2. Then using the software architecture artifacts of CM1, we estimate the change propagation probabilities and size of change, as shown in Figure 3 and Figure 4.

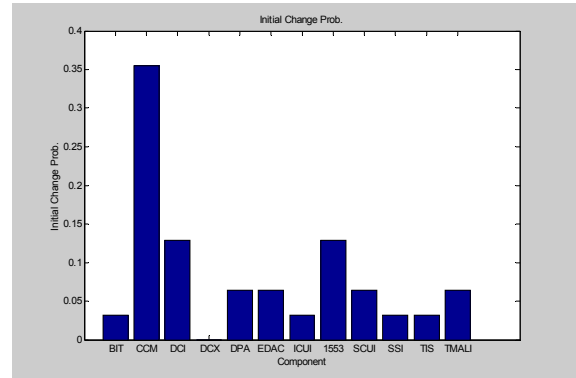


Figure 2 Initial change probabilities for CM1 components

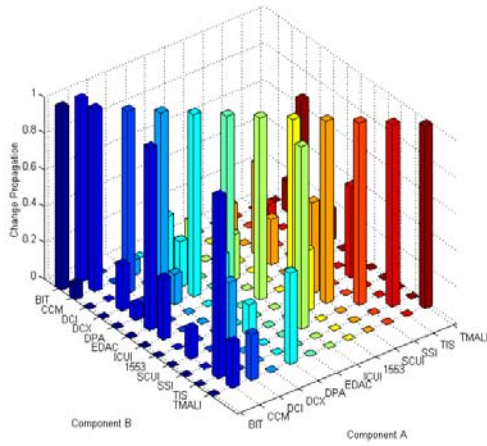


Figure 3 Change propagation probabilities for CM1

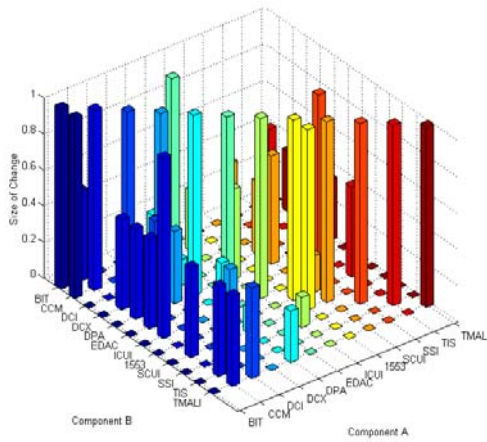


Figure 4 Size of change for CM1

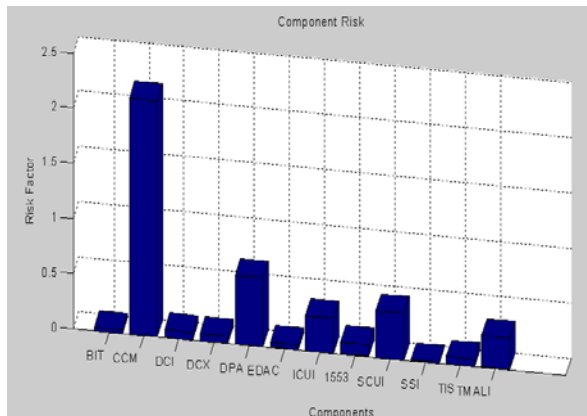


Figure 5 Maintainability-based risk for CM1 components

Substituting with the initial change probabilities of the components and the change propagation probabilities between them in equation (1), we

estimate the unconditional probability of change of the CM1 components. Then, we use the size of change between the components to account for the maintenance impact, based on equation (2). Using equation (3), the maintainability-based component risk factor for each CM1 component is estimated. The results are shown in Figure 5.

The most risky component with respect to adaptive maintenance is CCM. This is a result of CCM having the highest initial change probability. Moreover, CCM is coupled to most of the components, so it is likely to be affected by the changes introduced in these components (CP values are high). Furthermore, CCM has a high maintenance impact on the rest of CM1 components (Σsc_{ij} is large). As it is coupled to other components in the system, the change is likely to propagate further.

On the other hand, even though component 1553 has a relatively high initial change probability value, but it is coupled to a limited number of components in CM1 (CP values are low). Moreover, it has a limited maintenance impact (Σsc_{ij} is small) and it is less risky in terms of maintainability. On the contrary, component DPA has a relatively low initial change probability value. But due to change propagation (CP values are relatively high) and maintenance impact (Σsc_{ij} is not small), it is more risky in terms of maintainability.

4. Related Work

In this section, we discuss the related work in the literature. Our proposed maintainability-based risk assessment methodology relates to probabilistic risk assessment (PRA). PRA is an exercise for evaluating the probability of failure or success of a mission. Generally, decisions concerning choice of upgrades, scheduling of maintenance, etc are based on the outcome of the probabilistic risk assessment exercise. In [5], Bin et al. presented a framework for systematic integration of the software contribution to the risk in system failure analysis.

Several studies addressed the quantification of hardware maintainability but only few attempted to quantify software maintainability. One of the famous studies [12] introduced the Maintainability Index (MI) measure which is calculated using a polynomial of widely used code level measures such as Halstead measures and McCabe's cyclomatic complexity. In [10], Muthanna et al. conducted a similar study, which used design level metrics to statistically estimate the maintainability of software systems.

They constructed a linear model based on a minimal set of design level software metrics to predict Software Maintainability Index.

5. Conclusion

In this paper, we introduce and discuss the concept of architectural level maintainability-based risk assessment in the context of adaptive maintenance. Then, we present an estimation procedure based on change propagation probabilities using architectural information of the system and change reports of the system components. We also discuss a case study to illustrate our risk assessment methodology. This research work is a part of a wider effort that considers other architectural level risks such as reliability-based risk [7] and performance-based risk [6].

Among our venues of further research, we are considering to explore more case studies to test the maintainability-based risk of the components taking into consideration different types of maintenance. We also plan to automate the computation of the maintainability-based risk by expanding the Software Architectures Change Propagation Tool (SACPT) [4] and to augment it in our Architectural-level Risk Assessment Tool [16].

6. References

- [1] AbdelMoez W., I. Shaik, R. Gunalan, M. Shereshevsky, K. Goseva-Popstojanova, H.H. Ammar, A. Mili, C. Fuhrman, "Architectural Level Maintainability Based Risk Assessment", *IEEE International Conference on Software Maintenance, Poster proceedings (ICSM 2005)*, September 25-30, 2005, Budapest, Hungary.
- [2] AbdelMoez W., K. Goseva-Popstojanova, H.H. Ammar, "Methodology for Maintainability-Based Risk Assessment", *Proc. of the 52nd Annual Reliability & Maintainability Symposium (RAMS 2006)*, Newport Beach, CA., January 23-26, 2006.
- [3] AbdelMoez W., M. Shereshevsky, R. Gunalan, H.H. Ammar, Bo Yu, S. Bogazzi, M. Korkmaz, A. Mili, "Quantifying Software Architectures: An Analysis of Change Propagation Probabilities", *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 05)*, Cairo, Egypt, January 3-6, 2005.
- [4] AbdelMoez W., R. Gunalan, M. Shereshevsky, H.H. Ammar, Bo Yu, M. Korkmaz, A. Mili, "Software Architectures Change Propagation Tool (SACPT)", *Proc. 20th IEEE International Conference on Software Maintenance (ICSM 2004)*, Chicago, IL, September 2004.
- [5] Bin Li; Ming Li; Ghose, S.; Smidts, C., "Integrating software into PRA," *14th International Symposium on Software Reliability Engineering (ISSRE 2003)*, 17-20 Nov. 2003, Denver, CO., pp. 457- 467
- [6] Cortellessa V., K. Goseva-Popstojanova, K. Appukkutty, A. Guedem, A. Hassan, R. Elnaggar, W. Abdelmoez, and H. Ammar, "Model-Based Performance Risk Analysis", *IEEE Transaction on Software Engineering*, Vol.31, No.1, January 2005.
- [7] Goseva-Popstojanova K., A. Hassan, A. Guedem, W. Abdelmoez, D. Nassar, H. Ammar, A. Mili, "Architectural-Level Risk Analysis using UML", *IEEE Transaction on Software Engineering*, Vol.29, No.10, October 2003.
- [8] IEEE std 1219- 1998 IEEE standard for software maintenance
- [9] Metrics Data Program, NASA IV&V Facility <http://mdp.ivv.nasa.gov/>.
- [10] Muthanna S., K. Ponnambalam, K. Kontogiannis and B. Stacey, "A Maintainability Model for Industrial Software Systems Using Design Level Metrics", *7th Working Conference on Reverse Engineering (WCRE'00)*, Brisbane, Australia, November 23 - 25, 2000
- [11] NASA Technical Std. NASA-STD-8719.13A, *Software Safety*, 1997. http://satc.gsfc.nasa.gov/assure/nss8719_13.html
- [12] Oman, P. & Hagemester, J. "Constructing and Testing of Polynomials Predicting Software Maintainability." *Journal of Systems and Software* 24, 3 (March 1994), pp. 251-266.
- [13] Pigoski T.M., *Practical Software Maintenance: Best Practices for Managing Your Software Investment*, John Wiley & sons, 1996.
- [14] Sherer S., "Using Risk Analysis to Manage Software Maintenance", *Software Maintenance: Research and Practice*, Vol. 9, 345-364, 1997.
- [15] Unified Modeling Language OMG Resource Page <http://www.uml.org/>.
- [16] Wang T., A. Hassan, A. Guedem, W. Abdelmoez, K. Goseva-Popstojanova, H. Ammar, "Architectural Level Risk Assessment Tool Based on UML Specifications", *25th International Conference on Software Engineering*, Portland, Oregon, May 3 - 10, 2003.