Using Maintainability Based Risk Assessment and Severity Analysis in Prioritizing Corrective Maintenance Tasks [1]

W. Abdelmoez, K. Goseva-Popstojanova, H.H. Ammar
Lane Department of Computer Science and Electrical Engineering,
West Virginia University
Morgantown WV 26506
{rabie, katerina, ammar}@csee.wvu.edu

Abstract:- *A software product spends more than 65% of its lifecycle in maintenance. Software systems with good maintainability can be easily modified to fix faults. In this paper, we adapt our methodology for assessing maintainability-based risk into the context of corrective maintenance. The methodology depends on the architectural artifacts and their evolution through the life cycle of the system. In order to prioritize corrective maintenance tasks, we combine component maintainability – based risk with the severity of a failure that may happen as a result of unfixed fault. We illustrate the methodology on a case study using UML models.*

1. Introduction

Corrective software maintenance deals with fixing defects that escape detection before release and manifest as field failures. It is usually expensive, yet crucial to guarantee customer satisfaction. In this paper, we introduce and discuss the concept of architectural level maintainability-based risk assessment [1] in the context of corrective maintenance. Generally, maintainability-based risk takes into account the probability that the software product will need to endure corrective maintenance task and the consequences of performing this maintenance task on the system. Then, we present an estimation procedure based on change propagation probabilities using architectural information of the system and error reports of the system components.

Furthermore, we combine component maintainability based risk with the severity level of having an error in the component manifesting itself into a failure in order to prioritize corrective maintenance tasks. We used CM1 case study from the Metrics Data Program [4] to illustrate our risk assessment methodology and how to use the severity analysis of the system to set the corrective maintenance schedule.

2. Maintainability-Based Risk Assessment in Corrective Maintenance Context

The estimation procedure of maintainability-based risk presented in this paper builds on our previous work on change propagation probabilities $CP=[cp_{i/j}]$ [2] and size of change $SC=[sc_{i/j}]$. 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.

We will limit our consideration of maintenance effort to corrective maintenance. Therefore, we use error reports of errors that have not been yet fixed. To estimate the initial change probabilities $ICP=[icp_i]$, we first evaluate the frequency of occurrence of errors in each component C_i of the system. Then, we estimate the initial probability of change for each component by normalizing the frequency of occurrence for each component by the total number of error reports. Hence, the estimation methodology of maintainability-based risk is adapted for corrective maintenance.

To account for the dependency among 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. Then, The Maintenance Impact of the change in component C_i on the rest of the components of the system is estimated using the size of change metric SC. Finally, we estimate the components maintainability-based risk MR [1].

3. Prioritizing Corrective Maintenance Tasks

To order the corrective maintenance tasks for a certain project according to the importance of each task, we propose using the maintainability-based risk of the components that need to be fixed. Also, we propose to consider the severity-level of failures that may be manifested from the errors in these components. For maintenance tasks of components with critical or catastrophic severity-levels, the maintainability-based risk should not be of concern because of the consequences of such potential failures on the system. Such tasks should be of high priority in the maintenance plan.

[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.**

Table 1 Components severity of the CM1 case study

	Components											
	BIT	CCM	DCI	DCX	DPA	EDAC	ICUI	1553	SCUI	SSI	TIS	TMALI
Severity Level	Minor	Cat.	Cat.	Minor	Major	Major	Critical	Cat.	Critical	Cat	Major	Critical

On the other hand for maintenance tasks of components that have severity-levels of minor or major, we should examine the components maintainability-based risk. So, maintenance tasks of low severity-level and high maintainability-based risk should be avoided or delayed in the maintenance plan. Thus, we can prioritize the maintenance tasks accordingly.

4. Case Study

The severity analysis and the maintainability-based risk are assessed for the components of the CM1 case study. CM1 is a software component of a data processing unit used in an instrument, which exploits data to probe the early universe. For planning corrective maintenance of this system, we should think about components maintainability-based risk. Also, we should take into consideration the severity level of potential failures that could be caused by errors in components needed to be fixed.

The CM1 case study has 98 error reports of components bugs. Assuming that these errors have not been yet fixed, we want to prioritize the tasks of the corrective maintenance effort. First, we calculate the frequency of errors 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 error occurrences by the total number of error reports. Then using the software architecture artifacts of CM1, we estimate the change propagation probabilities and size of change. The estimated maintainability-based component risk factors for CM1 component are shown in Figure 1.

According to MIL_STD_1629A [5], severity considers the worst-case consequence of a failure determined by the degree of injury, property damage, system damage, and mission loss that could ultimately occur. Based on hazard analysis, we identify the severity classes: Catastrophic, Critical, Major and Minor. The assignment of component severity level of each component, shown in Table 1, is based on the hazard analysis conducted by domain experts knowledgeable about the case study. Due to space limitation the details of the case study and the discussion of the results cannot be presented.

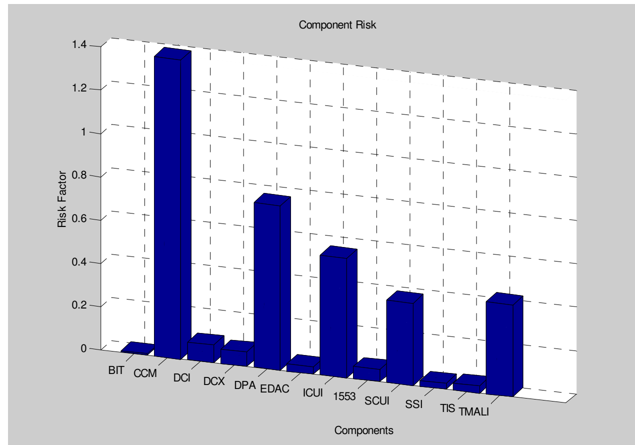

Figure 1 Maintainability- based risk for CM1 components

Among our venues of further research, we are considering to carry out more case studies to examine the maintainability-based risk of the components of system considering 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].

5. References

[1] 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.

[2] AbdelMoez W., M. Shereshevsky, R. Gunnalan, H.H. Ammar, Bo Yu, S. Bogazzi, M. Korkmaz, A. Mili , “Quantifying Software Architectures: An Analysis of Change Propagation Probabilties”, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 05), Cairo, Egypt, January 3-6, 2005.

[3] AbdelMoez W., R. Gunnalan, 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.

[4] Metrics Data Program, NASA IV&V Facility http://mdp.ivv.nasa.gov/.

[5] US Military Standard, “Procedures for Performing Failure Mode Effects and Criticality Analysis”, US MIL_STD_1629A / Notice 2, Nov. 1984.