Software Architecture Risk Assessment (SARA) Tool

K. Sheik, W. Abdelmoez, H. Ammar
Lane Department of Computer Science and Electrical Engineering,
West Virginia University
Morgantown, WV26506-6109
{khaders, rabie, ammar}@csee.wvu.edu

Abstract

Risk assessment helps projects to avoid unpredicted catastrophic problems. Also, it largely prevents wrong allocation of resources. In this paper, we present a tool that support architectural level model-based risk assessment, which includes reliability-based risk, requirements-based risk and maintainability-based risk. The tool accepts different kind of inputs. It parses these input files and produce quantitative metrics that are used to estimate the required risk factors.

1. Introduction:

A sound architecture is the means to build a software system with high quality attributes. Software architecture explicates the structure of the system in terms of components and interactions among them to accomplish the desired requirements. As architecture became a more significant artifact in developing software systems, the need to quantitatively analyze the architecture has become eminent. The architecture quantitative analysis should reflect its pertinent quality attributes and help us to predict the quality of the software products instantiated from it.

According to NASA-STD-8719.13A standard [6] risk is a function of the anticipated frequency of occurrence of an undesired event, the potential severity of resulting consequences, and the uncertainties associated with the frequency and severity. This standard defines several types of risk, such as for example availability risks, acceptance risk, performance risk, cost risk, schedule risk, etc. It helps project managers in avoiding unpredicted catastrophic problems. Also, it prevents wrong allocation of resources and taking decisions without proper knowledge or adequate information on anticipated future consequences. To manage software development projects, managers and developers should rely on processes, methods and tools to facilitate assessment, prioritization and mitigation of various risk aspects. Therefore, risk assessment is an essential part in the management of software development.

2. Software Architecture Risk Assessment (SARA) Tool:

In this paper, we present a tool that support architectural level model-based risk assessment, which includes reliability-based risk, requirements-based risk and maintainability-based risk. The Software Architecture Risk Assessment (SARA) tool extends our earlier Architectural-level Risk Assessment Tool [9] by providing support for more architectural models and different perspective of risk assessment other than reliability-based risk.

Reliability-based risk takes into account the probability that the software product will fail in the operational environment and the consequences of that failure [4]. While, requirements-based risk deals with the probability that the software will experience a requirement failure and the consequences of such failures [2]. On the other hand, maintainability-based risk takes into account the probability that the software product will need to endure a certain type of maintenance and the consequences of performing this maintenance on the system.[1].

The architecture of SARA tool is shown in Figure 1. It accepts different input formats, such as Rose RealTime [7] models, StarUML [8] models and Java Understand [5] static analysis files. First, we extract the required architectural –level information from the examination of these inputs and store it in the software architecture artifacts repository for further analysis. According to the type of risk assessment to be performed, the tool evaluates the metrics required such as cyclomatic complexity, dynamic coupling, change propagation probabilities, size of change and error propagation probabilities using the stored artifacts in the repository. Then, the tool admits the analyst to provide the severity analysis corresponding to the considered type of risk. Finally, the tool provides the analyst with the risk estimates for the components of the system.

---

1 This work is funded in part by grants to West Virginia University Research Corp. from the National Science Foundation Information Technology Research (ITR) Program grant number CCR-0082574, and 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.
In Figure 2, a snap shot of the tool is showing the results of change propagation probabilities obtained from a StarUML model. In Figure 3, a snap shot of the tool showing the results of the maintainability-based risk for corrective maintenance of the same case study.

Among our venues of further research, we are considering to support more input formats for the tool and add other risk assessment perspectives like performance-based risk [3].

3. References


