NASA Software Tools for High-Quality Requirements Engineering

James R. McCoy
Software Assurance Technology Center
NASA Goddard Space Flight Center
Greenbelt, MD  20771
james.mccoy@gsfc.nasa.gov

Abstract

Extensive research has been conducted on writing quality software requirements in a natural language, resulting in the development of guidelines for writing effective requirements and a tool for evaluating them. Concepts from this application, the Automated Requirements Measurement (ARM) tool, which scans a requirements document for specific key words and phrases that impact the quality of the requirements, were then applied to the area of systems safety, resulting in the creation of the Safety Critical Analysis Tool (SCAT). Current research states that use cases provide a more methodological basis for specifying and managing quality functional requirements than the traditional natural language approach. Therefore, the Requirements Use case Tool (RUT) was developed to provide a template and repository for high-quality use cases. This paper describes the contribution to generating high-quality requirements made by each of these three tools.

1. Introduction

It is generally considered fundamental by software engineers that system requirements are the foundation upon which an entire system is built. It is further presumed that verification and validation are needed to assure that the desired functionality represented by the total set of requirements is ultimately delivered. All too often, however, development teams fail to satisfy customer requirements, and it is this shortfall, usually discovered quite late in the schedule, that leads to a frantic cycle of fixing and patching software. The belief is that correct, complete, and testable system requirements are critical. The success of a project—as measured in either functional or financial terms—is directly affected by the quality of the requirements [8].

Requirements are the basis for project development, and the objective of work in all phases is to implement the requirements. Requirements development and management have always been critical in the implementation of software systems—engineers are unable to build what analysts can’t define. It is generally accepted that the earlier in the life cycle potential risks are identified the easier it is to eliminate or manage the conditions that introduce those risks. Problems that are not found until testing are approximately 14 times more costly to fix than if the problem was found in the requirements phase. Other literature states that problems in requirements will result in over a 100% increase in price if the problems are not found until the testing phase as opposed to the finding them in the requirements phase.

The vehicle for capturing and communicating requirements is a document known as the Software Requirements Specification (SRS). The requirements specification, as the first tangible representation of the capability to be produced, establishes the basis for all of the project’s engineering management and assurance functions. If the quality of the SRS is poor it can give rise to risks in all areas of the project. It is not sufficient that the development and customer organizations simply understand the set of systems requirements; the critical challenge is for these two stakeholders to arrive at the same understanding.

The Unified Modeling Language (UML) was created through the unification of the best engineering practices of the industry for modeling systems, especially large, object-oriented software systems. In the UML, use cases play a pivotal role in capturing the functional requirements of the system under development. The user-centered approach provided by use cases allows the specification of understandable, buildable, and verifiable requirements. The use of the UML to describe requirements is purported to result in an improved set of requirements.

This paper describes three NASA-developed tools for assessing requirements. Each tool is described in terms of the quality attributes it identifies and the process by which it evaluates a requirements document.