A Case for Requirements Validation

Mats P.E. Heimdahl

April 15, 2004

In software engineering we make a distinction between the validation and the verification of a software system under development. Verification is concerned with demonstrating that the software implements the functional and non-functional requirements. Verification answers the question "is this implementation correct with respect to its requirements?" Validation, on the other hand, is concerned with determining if the functional and non-functional requirements are the right requirements. Validation answers the question "will this system, if build correctly, be safe and effective for its intended use?" There is ample evidence that most safety problems can be traced to erroneous and inadequate requirements. Therefore, to improve the safety of software intensive systems it is critical that the requirements are properly validated. Unfortunately, current certification standards, for example, DO-178B, focus almost exclusively on various verification activities; consequently, most industry practices are geared towards verification activities such as extensive testing and code inspections. Thus, one of the most critical problems with current certification standards is a lack of robust and reliable ways of assessing whether the requirements have been adequately validated.

There is a significant effort in the avionics and medical technology industry to reduce the high cost of software development. The current trend is to focus on tools and automation, for example, automatically generating certifiably correct production code from a formal requirements specification or generating MC/DC tests for certification could provide dramatic cost savings. These cost savings will be achieved by replacing time consuming and costly manual processes, for example, the definition of test-cases, with tools. Our current inability to adequately validate our requirements, however, raises a serious concern regarding the adoption of this type of automation. Manual processes, may that be design, coding, testing, or putting a medical device through clinical studies, draw on the collective experience and vigilance of many dedicated software professionals; professionals that provide "collateral validation" as they are working on the software. Experienced professionals developing code or defining test cases provide additional validation of the software system; if there is a problem with the required functionality of the system, they have a chance of noticing and taking corrective action. When replacing these manual efforts with automation, proper validation of the formal requirements specifications on which the automation is based becomes absolutely essential.

In summary, adequate validation of system requirements is a critical issue for both current and future software development practices and certification standards—proper validation is a necessary condition for certifiably dependable systems.