Risk-Based Trade-Off Between Verification and Validation – An Industry-Motivated Study

Kennet Henningsson and Claes Wohlin

Blekinge Institute of Technology, PO BOX 520, 37225 Ronneby, Sweden
{Kennet.Henningsson, Claes.Wohlin}@bth.se

Abstract. Within industry the demand for short lead-time and reduced effort consumption is in focus. For an associated industry partner the lead-time and effort focus has meant turning the interest towards the Verification and Validation (V&V) process. The industry cooperation motivating this study aims at providing a tailored and applicable V&V process, where the order of verification and validation may be changed as well as the amount of V&V activities conducted. Through the industry cooperation as well as industrial and academic experience, a method has been formulated that address how to select a suitable V&V process depending on the functionality being developed. The method describes how a suitable process is created and selected, where the appropriate process is identified based on functionality and coupling between the system entities being developed. It is concluded that the method provides support, structure and clarification to address the possibilities to a trade-off between verification and validation.

1 Introduction

Software process models have evolved over the years starting with the waterfall model [1] into agile processes [2] In-between several process models have been proposed and used, including the spiral model [3], incremental and evolutionary development [4, 5] among others. The introduction of new models is a response to the constant need of improvement in software development. Companies are constantly making trade-offs between cost, delivery time, quality, and delivered functionality. The software community also recognizes that not all types of software need the same treatment. Different processes, methods, techniques, and tools are used depending on the type of software [6] [7, 8]. Safety-critical software has different requirements than computer games, and hence the development process has to be adapted accordingly.

It is clear that different types of software need different processes. If taking this one-step further, different functionality needs different processes. However, this requires processes that are possible to tailor [9, 10]. In other words, the software process needs to be flexible. Process tailoring is discussed in the context of the Capability Maturity Model (CMM) in [9]. In a joint project with an industrial partner, in this case UIQ Technology (a subsidiary of Symbian), it was decided that an attempt should be made to tailor their verification and validation (V&V) activities [11] based

F. Bomarius and S. Komi-Sirviö (Eds.): PROFES 2005, LNCS 3547, pp. 443–457, 2005.

on the types of functionality being developed. Verification is concerned with whether the software is developed correctly and validation is focused on whether the correct software is developed. The company works in an environment where some functionality is fairly standard or similar to other functionality that has been developed previously and other functionality is leading edge services that have not been developed before. The company wants to work with flexible processes to ensure as short lead times as possible for their software.

These different types of functionality require different approaches to V&V, or at least open up for an opportunity to run the V&V activities differently. For example, standard functionality is straightforward to develop from a user point of view; the company is confident of what functionality the user desires. On the other hand, the user has high expectations when it comes to the quality of service for this type of functionality. Thus, in this case a possible scenario is that there is rather little need for validation, but a high need for verification. At the other end of the spectrum, there are completely new functions that are not currently available on the market. To ensure that these services meet the needs on the market, it is important to validate their usefulness with users or representatives for the markets in some sense. Normally, the expectations on correctness are less for new functionality and hence it may be discussed how much verification is needed. In this case, the validation is probably more critical than the verification, the situation can occur when it is necessary to negotiate quality, or functionality, of the delivered product [6]. This reasoning implies that different ways of handling V&V can be applied for different types of functionality within the same project.

The functionality characteristic indicates the uncertainty whether the correct functionality is implemented or not, i.e. likelihood for changes and, to some extent, faults. However, this does not provide the true picture of the risk (risk meaning exceeded budgets, too low quality, and increased project lead-time). There is a need for determining the risk impact as well. A system entity's coupling, i.e. connections to other system entities indicate the magnitude of the risk. The magnitude shows the number of dependencies each entity has, and how large the ripple effect would be by a late change or fault. It is likely that for a system entity with a high number of connections, the impact would be larger than for a low-coupled entity. By determining the functionality characteristic for the system entity as well as the coupling, sufficient information is available for risk assessment.

Bottom-line is that the company wanted a risk-based approach to their V&V activities, providing the possibility to determine which risks to take in terms of how much verification, and how much validation that should be performed for different types of functionality.

This paper presents a method for selecting a suitable V&V process based on being flexible regarding both the amount (in relative terms) of V&V needed and the order of the activities. The method is motivated by industry needs, and it is developed from an industrial case using an exploratory approach. The method is presented in a number of steps that can be adapted to different situations depending on the functionality being developed. This method is partly illustrated using input from the industrial partner where the need for this type of approach was identified.