Requirements for Technical Quality of Software Products

[SimSat case study – abstract]

Tiago L. Alves
Software Improvement Group, The Netherlands, and University of Minho, Portugal
t.alves@sig.nl

ABSTRACT
Ground Systems software follows a well-defined process as defined by the Software Engineering Standards for Ground Segments in ESA (BSSC). From the specification to the actual deployment and operation of the software, every step of the process is rigorously documented to ensure a correct implementation of the functional requirements. Software technical quality is addressed, as part of that process, in the Software Product Assurance Plan and through code reviews using Telelogic’s Logiscope. However, is this enough to ensure high technical quality of ground system software?

Using as case study SimSat, we present the findings of an assessment of the technical quality of software products. We identify several technical quality problems. Low technical quality leads to high maintenance costs. We conclude, that to prevent or reduce costs ESA shall impose and enforce measurable requirements on technical quality of software products.

The ISO 9126 international standard for software product quality identifies maintainability as one of the six main quality characteristics. Maintainability is then sub-divided into analyzability, changeability, stability and testability. For rating these characteristics on the basis of the source code a quality model employing metrics has been proposed by the Software Improvement Group: Volume (the overall size in man-years), Duplication (the percentage of code that is exactly copied), Unit size (size of methods and procedures measured in LOC), Unit complexity (number of decisions per method or procedure), and Test quality (the existence of both unit and integration tests). The analysis of these metrics allows the identification of potential risks and the calculation of ratings. Ratings are expressed in stars: five stars mean excellent quality, four stars good, three stars fair, two stars poor and one star very poor.

We analyzed the source code quality of the Simulation Infrastructure for Modelling of Satellites (SimSat), version 4.0.1, according to the metrics above referred. SimSat is available as two modules: Kernel and MMI. The Kernel is mainly written in C++, containing near 140K lines of code (LOC). The MMI is mainly Java, containing over 170K LOC. For volume, Kernel and MMI together represent a rebuild value of about 47 staff years, leading to a score of three stars. Since the two modules are fairly independent of each other, SimSat is still manageable. However, to reduce risks further splitting MMI into two physical components would be advisable. For duplication, SimSat ranks two stars. 7% of the Kernel overall code and 11% of the MMI overall code could be removed. This duplication is caused by the existence of several duplicated files and weak reuse of encapsulation and abstraction. For unit size, SimSat ranks one star. In the Kernel only 24% of the code is in small methods, while 45% of the code is in very large methods (> 100 LOC). The MMI has better quality. 44% of the code is in small methods but 12% of code is in very large methods. Methods with a higher size are more difficult to comprehend, test and maintain requiring more maintenance effort. For unit complexity, SimSat ranks two stars. In the Kernel 85% of the code contains low complexity, but there is 2% with very high complexity (> 50 decisions). We observed similar complexity in the MMI, 81% of the code contains low complexity, but there is 3% with very high complexity. We have discovered that very high complexity is localized in just a few files. Methods with very high complexity are considered as untestable, and avoiding them would improve maintainability. For test quality, SimSat ranked three stars. Automatic functional test was found, but no unit testing.

The inexistence of unit tests increases maintenance costs since unit tests help to reduce time to reproduce problems and prevent the reintroduction of bugs.

The overall rating for SimSat is two stars, indicating poor maintainability. Based on industry average productivity statistics, and assuming that 15% of the code is changed yearly, we estimate that SimSat has a maintenance cost of 6 man-years. We estimate, that when duplication and complexity are improved, this could reduce maintenance by 17%, to 5 man-years.

To conclude, although SimSat was developed following a strict process, quality problems were still found, in particular regarding duplication, complexity and unit size. We recommend that actions should be taken to solve these problems and that the degradation of quality is monitored. This would not only improve overall quality but also reduce maintenance costs. Moreover, new projects could benefit from adding to the requirements that all software have at least four stars for quality. Generalizing from the SimSat case study, the application of these techniques to other ESA software systems would offer a better overview of the overall quality, insight about the existent risks and decision support for controlling and improving software quality.