Luis Javier García Villalba (Ed.)

Advanced Science and Technology Letters ASTL 27

Next Generation Computer and IT Applications

Proceedings

International Conferences,

NGCIT, CST, IESH, MMHS 2013

September 22 - 24, 2013

Haitian Grand Theatre Hotel Qingdao, China

SERSC Science and Engineering Research Support soCiety

Empirical Practice of Embedded Software Quality Improvement for managing water resource system based on ISO/IEC 9126

Kidu Kim[1], R.Young Chul Kim[2]

[1] Telecommunications Technology Association, 47, Gyeonggi, Korea
kdkim@tta.or.kr[1]
[2] Dept. Of CIC(Computer and Information Communication), Hongik University, Sejong Campus, 339-701, Korea
bob@hongik.ac.kr[2]

Abstract. According to the experience in testing department for many years, software testing in a company is usually verifying the completeness of developed code before product release. The program, which has shipped without verification of non-functional defects, causes user inconvenience. In this paper, we describes empirical practice for quality improvement of water resources management system based on ISO/IEC 9126 including developer-centric functional test.

Keywords: ISO/IEC 9126, embedded software, SCADA, TAG

1 Introduction

In my software testing experience in TTA(Telecommunications Technology Association), most software companies had lack of testing capability. Especially in the area of embedded system, there are lots of knowledge for development, not for testing and management. Because a product is released only after verifying source code, it has latent faults of usability which causes the end user inconvenience.

The TEmb[1] method is published in the book Testing Embedded Software by Bart Broekman and Edwin Notenboom. TEmb is a method that helps to assemble a suitable test approach for a particular embedded system. It provides a mechanism for assembling a suitably dedicated test approach from the generic elements applicable to any test project and a set of specific measures relevant to the observed system characteristics of the embedded system.

2 Related works

ISO/IEC 9126 defines terms for the software quality characteristics and how these characteristics are decomposed into sub-characteristics (Figure 1). The sub-

characteristics can be measured by internal or external metrics. Software quality can be evaluated by measuring internal attributes (typically static measures of intermediate products), or by measuring external attributes (typically by measuring the behavior of the code when executed).

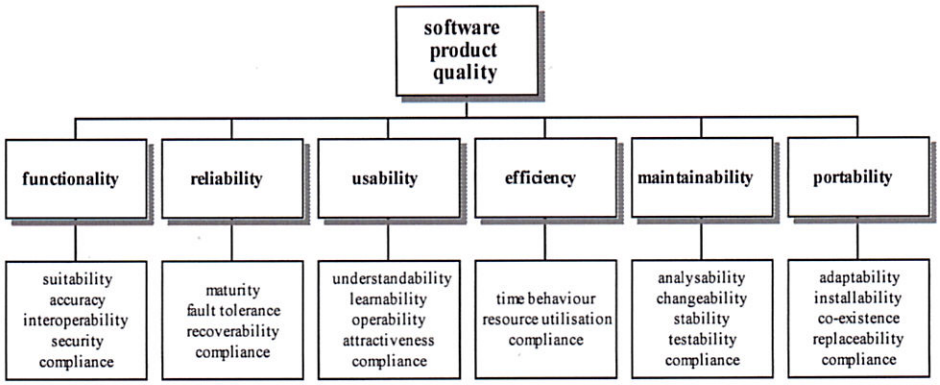

Fig. 1. ISO/IEC 9126 Software product quality [2]

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints such as cellular phone, railway signaling system, hearing aid, missile tracking system. An embedded software is sometimes used interchangeably with firmware, although firmware can also be applied to ROM-based code on a computer, on top of which the OS runs, whereas embedded software is typically the only software on the device in question. Figure 2 shows a generic layout, which is applicable to virtually all embedded systems.

Embedded software testing shares much in common with application software testing. Developer codes a software, compiles, and tests to check the function. This is a developer centric test and is not sufficient to reduce latent faults.

There are just few embedded software testing methodologies. TEmb uses the four cornerstones of structured testing as defined by the test management approach TMap

3 Testing embedded software for water resource management system

There are sequences for testing. : 1. Preparation, 2. Analysis & Design, 3. Testing, 4. Regression test, 5. Reporting.

Product under test is a water resource management system, which controls water level of a dam, and monitors condition. It requires real-time commands and controls, but we use both simulated system and real system for testing.

With consideration of the characteristics of embedded software, we wrote test cases, and added more test cases in progress of testing. The test case has a unique ID, expected result, and execution result.

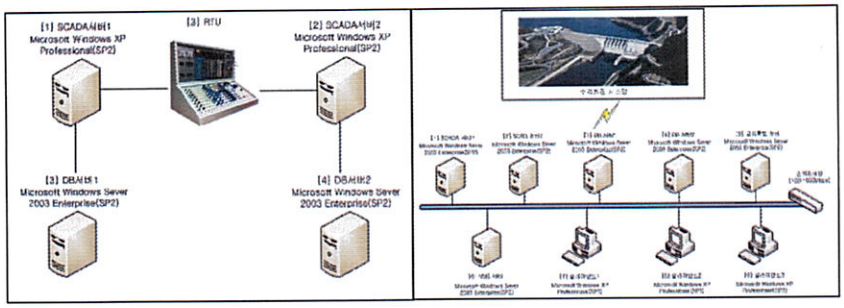

Fig. 2. Testing environment (simulated system and real system)

When performing the test, we found several faults, and classified based on ISO / IEC 9126 quality characteristics. We found 90 faults on 4 quality characteristics(functionality, reliability, usability, portability)(Figure 3)

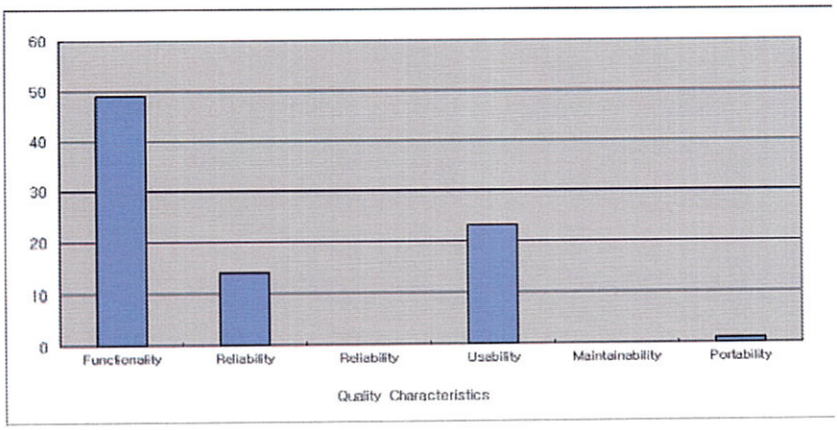

Fig. 3. Faults per quality characteristics

There was 4 patches to fix all found faults. In 1st stage, developer fixed faults without information of quality characteristics. After 2nd stage, developer fixed faults with information of quality characteristics. Rate of patch was only 6% in 1st stage, but 68% in 2nd stage.(Table 1).

Table 1. Rate of Patch

Content	1st stage	2nd stage	3rd stage	4rd stage
Rate of Patch	6 %	68 %	80 %	100 %

So providing information of quality characteristics helped developer to comprehend what was error. We also found latent fault, during patch.

4 Conclusion

Previous embedded software testing was verifying source code using functional testing. Therefore, actual testing was not sufficient except program function. In this paper, we tested an embedded system for water resources management based on ISO / IEC 9126. As a result, providing information of quality characteristics improves efficiency of modification 60%, and prevents mistakes of developer's modification. However, even one case study of ISO/IEC9126 based test is not sufficient. Later, we will derive items to improve the quality except the quality characteristics of ISO / IEC 9126.

Acknowledgments. This work was supported by the IT R&D Program of MKE/KEIT [10035708, "The Development of CPS(Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software"]

References

1. Bart, Broekman., Edwin, Notenboom.: Testing Embedded Software, (2002)
2. ISO/IEC TR 9126, Software engineering-Product quality-Part 1, 2, 3, 4, (2005)
3. ISO/IEC 14598, Information Technology-Software Product Evaluation-Part 1,2,3,4,5,6, (2005)
4. ISO/IEC 25000, Software Engineering-Software Product Quality Requirements and Evaluation-Guide to SQuaRE, (2005)
5. ISO/IEC 25010, Software Engineering-Software Product Quality Requirements and Evaluation-Quality Model, (2005)

Advanced Science and Technology Letters

The ASTL series is committed to the publication of proceedings of Advanced Science and Technology. Its objectives is to publish original research in various areas of Science and Technology. This will provide good chances for academic and industry professionals to discuss recent progress in areas of Science and Technology.

Research papers were strictly peer-reviewed by program committees to make sure that the the papers accepted were high quality and relevant to the current and future issues and trends in Science and Technology.

The scope of ASTL includes the entire area of science and technology from the current and future trends.The Language of publication is English. The Authors have to sign the SERSC ASTL copyright transfer form.

ISSN 2287-1233

ASTL

SERSC