Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 29-30, 2009

Study on Technical Requirements and Guidelines for Safety-Critical Software Maintenance

Young-Mi Kim and Choong-Heui Jeong

Korea Institute of Nuclear Safety, P.O.Box 114, Yuseong-gu, Daejeon, Korea, 305-600
**Corresponding author: ymkim@kins.re.kr*

1. Introduction

Most of the life times of digital I&C systems have been under maintenance phase. Safety-critical systems which have been used in operating plants contain software as well as hardware. Maintenance of software used in safety-critical system affects the dependability of entire system. The level of dependability of the safety-critical system after the software was maintained must be to keep equal or more before. Software maintenance is different from hardware maintenance because each software changes result in new software. So the maintenance of safety-critical software should be strictly controlled under the dependability management of entire system. There are many industrial standards and activities for software maintenance. In this paper, we describe the technical requirements and guidelines with respect to safety-critical software maintenance are explored.

2. Maintenance of Digital I&C System

In this paper, we focus software maintenance especially post-delivery activities.

2.1 Software Maintenance

Software maintenance is different from hardware maintenance because each software changes result in new software. Software maintenance covers the correction of errors, the enhancement, deletion and addition of capabilities, the adaptation to changes in data requirements and operation environments, the improvement of performance, usability, or any other quality attribute. IEEE 610.12 defines as follows [1]:

"Software maintenance is the process of modifying a software system or component after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment."

Requirements for software maintenance should cover the following needs:

- Changes of technical requirements
- Changes of software environment
- Anomalies found during operation

2.2 Types of Software Maintenance

There are several types of software maintenance. Wang divide maintenance into four categories: corrective, adaptive, perfective and preventive [2].

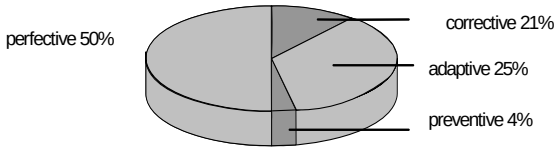

Fig. 1. Distribution of maintenance activities

Corrective maintenance is reactive modification of a software product performed after delivery to correct discovered faults. Adaptive maintenance encompasses modification of a software product performed after delivery to keep a computer program usable in a changed or changing environment. Perfective maintenance includes modification of a software product performed after delivery to improve performance or maintainability. Preventive maintenance involves making changes to software that improve neither correctness nor performance but make future maintenance activities easier to be carried out. Fig. 1 shows the distribution of maintenance activities. Half of the maintenance activities are belongs to perfective maintenance.

IEEE defines emergency maintenance instead of preventive maintenance. These definitions introduce interesting ideas such as Table 1.

Table 1: Classification of IEEE Software Maintenance

Input	Scheduled	Unscheduled
Reactive	Emergency	Corrective Adaptive
Proactive		Perfective

2.3 Maintenance Testing

Usually, software maintenance often consumes the most time in the software life cycle. It is important to establish a safe and well-controlled mechanism for software maintenance.

Regression testing is closely related software maintenance. Changed software caused by maintenance should be tested thoroughly against documented

specifications. Regression testing refers to that portion of the test cycle in which a program *P'* is test to ensure that not only does the newly added or modified code behaves correctly, but also that code carried over unchanged from the previous version *P* continues to behave correctly.

Coverage analysis results must be documented. If the coverage analysis determines that the coverage is incomplete, additional tests must be performed to complete the testing.

High maintainability helps the released software to be maintained and changed easily. But, maintainability is usually neglected by software developers. Maintainability should be considered from the very start of the software life cycle. For safety-critical software, high maintainability is more necessary.

3. Technical Requirements of Safety-Critical Software

In this section some of the technical requirements for software maintenance are described.

3.1 Technical Requirements for Software Maintenance

IEEE Std. 1012, IEEE Std. 1008, and IEEE Std. 829 are software maintenance related industry standards which are approved by U.S NRC. IEEE Std. 1219 is software engineering standard which can be applied to all software applications. Table 2 shows some of the maintenance related international standards [1, 3-7].

Table 2: Software maintenance related international standards

Standard	Title
IEEE Std. 1012	Software V&V
IEEE Std. 1008	Software Unit Testing
IEEE Std. 829	Software Test Documentation
IEEE Std. 1219	Software Maintenance
IEC 60987	Programmed Digital Computers ITS in NPP(Hardware A;B;C)
IEC 60880	Software for Computers in the Safety Systems of NPS
IEC 60880-2	Supplement to IEC 60880 (Predeveloped software A)
IEC 61513	NPP I&C Systems Important to Safety General Requirements for Computer Based Systems
IEC 62138	NPP I&C Systems ITS Computer Based Systems Software Aspects for I&C Systems of Class 2 and 3
IAEA NS-G-1.1	Software for Computer Based Systems ITS in NPP
IAEA NS-G-1.3	Instrumentation and Control Systems important to Safety in NPP

3.2 Processes of Maintenance

IEEE Std. 1219 shows the process of software maintenance. Software maintenance starts with modification request. Following phases are a) problem/modification identification, classification, and prioritization; b) Analysis; c) Design; d) Implementation; e) Regression/system testing; f) Acceptance testing; and g) Delivery. Fig 2 shows these phases. The associated metrics/measures for each phase are identified specifically in IEEE Std. 1219 [5].

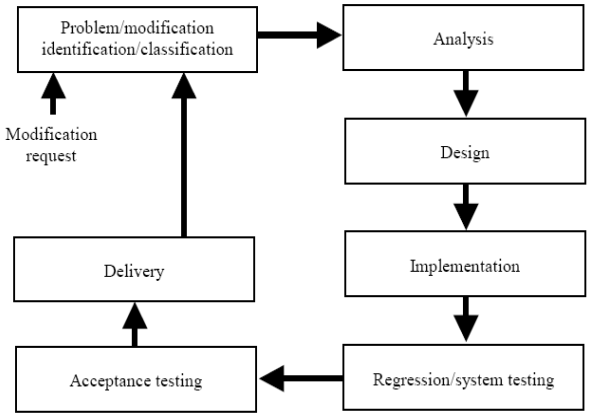

Fig. 2. Distribution of maintenance activities

4. Conclusions

As the number of operating plants and operating years are increased, as the requirements for maintenance of safety-critical systems are also increasing. Safety-critical systems which have been used in operating plants contain software as well as hardware. Maintenance of software used in safety-critical system affects the dependability of entire system. This paper addresses some issues and technical requirements for maintenance of safety-critical software.

REFERENCES

[1]IEEE Std 610.12, Standard Glossary of Software Engineering Terminology, IEEE Computer Society, 1990
[2]Ligfeng Wang and Kay Chen Tan, Software Testing for Safety-Critical Applications, IEEE Instrumentation & Measurement Magazine, June 2005
[3]IAEA Safety Standard Series No. NS-G-1.1, Software For Computer Based Systems Important to Safety in Nuclear Power Plants, 2000.
[4]IAEA Safety Standard Series No. NS-G-1.3, Instrumentation and Control Systems Important to Safety in Nuclear Power Plants, 2002.
[5]IEEE Std 1219-1998, IEEE Standard for Software Maintenance, Software Engineering Standards Committee of the IEEE Computer Society
[6]IEEE Std 1008-1987, IEEE Standard for Software Test Documentation, IEEE Computer Society
[7]IEEE Std 829-1998, IEEE Standard for Software Unit Testing, IEEE Computer Society