Methods of Software Qualification for a Safety-grade Optical Modem to be used Core Protection Calculator (CPC) in Korea Standard Nuclear Power Plant (KSNP)

Jangyeol Kim¹ and Soongohn Kim²*

¹Instrumentation and Control & Human Factors Division/Korea Atomic Energy Research Institute (KAERI)
989-111 Daedeok-daero, Yuseong-gu, Daejeon, Republic of Korea 305-353
²Division of Computer and Game Science, Joongbu University,
101 Daehakro, Chubu-Meon, GumsanGun, Chungnam, 312-702, Korea
¹jykim@kaeri.re.kr, ²sgkim@joongbu.ac.kr

Abstract

This paper describes safety-critical software qualification methods and its approach, which corresponds to well-structured qualification organization, methods and results, software quality assurance, and software configuration management. This paper also describes the verification test environment, test components and items, a traceability analysis, and system tests as a result of software qualification based on Software Requirement Specifications (SRS) for a safety-grade optical modem of a Core Protection Calculator (CPC) in a Korea Standard Nuclear Power Plant (KSNP), and Software Design Specifications (SDS) for a safety-grade optical modem of a CPC in a KSNP. We believe that we achieve the functionality, performance, reliability, and safety, which are the software qualification objective goals of safety-critical systems.

Keywords: Software Qualification, Software Quality Assurance, Verification and Validation, Software Configuration Management, Integration & System Test, Safety-grade Optical Modem, Core Protection Calculator, Korea Standard Nuclear Power Plant

1. Introduction

To qualify as safety-critical software, the defined responsibilities among the assurance organization are very important. The development team is responsible for producing a design output during the software life cycle. The Software Verification & Validation (SVV) and Software Safety Analysis (SSA) are used for a safety qualification on the produced design output by the development team. First, prior to use, a Commercial Off-The Shelf (COTS) software tool should be dedicated by the quality assurance organization. The Software Configuration Management under Software Quality Assurance is responsible for the configuration identification, which are the status accounting, and the revision control on all of the design output and its verification results. The suggested well-structured organization described in this paper is as shown in Figure 1.
This paper describes the test environment, test components and items, a traceability analysis, and system tests as a result of system verification and validation based on Software Requirement Specifications (SRS) for a safety-grade optical modem of a Core Protection Calculator (CPC) in a Korea Standard Nuclear Power Plant (KSNP), and Software Design Specifications (SDS) for a safety-grade optical modem of a CPC in a KSNP.

2. Methods and Results

The methods used for software qualification are requirement traceability, software verification testing, software quality assurance and software configuration management. In this section, the test methods and results are described. Above all, in the case of safety-critical systems, it is important whether system test results on the host environment are satisfied with the target board have been performed or not. Functional tests, performance tests, event tests and scenario tests for safety-grade optical modem has been tested successfully. Coverage of the range value, boundary value, and equivalent value were also measured.

2.1. Traceability Analysis

A traceability analysis is performed through the Code & Standard criteria, requirements, test identifier (ID) and object code of target board of completeness, correctness, and consistency. The accuracy of a definition, input/output accuracy,
accuracy of software behavioral characteristics, and accuracy of an interface of the software function are very important. Traceability analyses are classified by basis reviews & comments and the latter details the requirement traceability management, as shown in Figure 2.

![Figure 2. Requirement Traceability Analysis](image)

### 2.2 Verification test environment

Software testing consists of a component test, an integration test, and a system test. These test executions produce a test plan generation, a test design generation, a test case generation, and a test procedure generation according to the software test life cycle, as shown in Figure 3. A test plan should be prepared to satisfy the functional requirements and performance requirements, event test requirements, and scenario test requirements of the safety-critical system.

![Figure 3. Testing Process by Test Life Cycle](image)

The system test performs functionality, performance, event tests, and scenario tests including error detection and diagnosis, scan time violation, deterministic communication, and communication independence.

Application firmware of the safety-grade optical modem to be used the Core Protection Calculator (CPC), in a Korea Standard Nuclear Power Plant (KSNP) was developed under GNU/Linux Ubuntu 11.10 of AMD64 environments. To build a system test with the host environment, firmware was ported in the target board of a safety-grade optical modem using...
USBISP. To measure the embedded software of a safety-grade optical modem, an AVR USBISP V3.0 and avrdude 5.10 utility were used as shown in Figure 4.

![USBISP and avrdude](image)

**Figure 4. Verification Test Environments of Safety-Grade Optical Modem**

### 2.3. Test Components and Test Items

Test components and items are as shown in Table 1.

**Table 1. Test Components and Items for Safety-Grade Optical Modem of CPC**

<table>
<thead>
<tr>
<th>NO</th>
<th>Category</th>
<th>Test Components</th>
<th>Test Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Functional test</td>
<td>Initial setup Variable of Hardware and Software - Optical Modem - LED - Timer - WDT etc.</td>
<td>Optical signal translation Voltage - Optical signal Optical signal - Voltage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data communication Sending Only (Unidirectional) Receiving Only (Unidirectional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Status indication POWER TX RX FAULT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Setup Gain, Offset Protocol Protocol Analysis (Packet) CRC8</td>
</tr>
<tr>
<td>2</td>
<td>Performance test</td>
<td>Accuracy Accuracy ±0.05% Communication speed - 4ms - 57600bps</td>
<td>Protocol Protocol Analysis (Packet) CRC8</td>
</tr>
<tr>
<td>3</td>
<td>Event test</td>
<td>Fault injection Power Fail, Abnormal State - Signal short - CRC - Timeout - Frame Error - Buffer overflow</td>
<td>Protocol Protocol Analysis (Packet) CRC8</td>
</tr>
<tr>
<td>4</td>
<td>Scenario test</td>
<td>Continuous operation test About three months burn-in test</td>
<td>Protocol Protocol Analysis (Packet) CRC8</td>
</tr>
</tbody>
</table>
In particular, the performance requirements listed above should be satisfied for the purchase order requirements of Korea Hydro and Nuclear Power Co. Ltd (KHNP) as follows.
- The response time should be less than 4ms.
- Full Range Accuracy within ± 0.05% or better should be satisfied.
- Unidirectional buffering and deterministic communication should be satisfied.

2.4. Test Results

The initialization setup, optical signal conversion capabilities, communication capabilities and accuracy, display status indication, parameter setup, and protocol were set up in the functional tests. The Performance tests were carried out as follows:
- Response time : 4ms
- Accuracy of ± 0.05%
- 57600 bps transfer rate
- Communication time between the ADC (Analog Digital Converter) and MCU (Main Control Unit)
- Communication time between the MCU and DAC (Digital to Analog Converter)
- Optical modem transmitter offset
- Gain adjustment between the MCU and DAC
- TWI (Two Wire Interface) communication as an optical transmitter
- TWI communication as an optical receiver
- Communication between the MCU of the optical modem sender and external clock
- Communication between the MCU of the optical modem receiver and external clock
- Status of communication tracking between the MCU of the optical modem receiver and optical receiver’s component

The verification results of the performance test for a 57600 bps transfer rate and response time (4ms) among several performance tests are shown in Figure 5 and Figure 6, respectively.

![Figure 5. Transfer Rate of 57600bps](image-url)
Event tests were performed based on the error injection; in particular, a signal short-circuit among several error injections was tested successfully.

The signal source of making a triangular wave under the verification test oracle equipment was used as shown in Figure 7. A scenario based burn-in test was carried out during three months and two weeks continuously.

3. Software Quality Assurance Measurement

3.1. Software Quality Assurance

After the system test is completed, an inside peer review was performed in the same way as that of the integration test and system test. The defects, problems and corrective actions list should be submitted to the Software Quality Assurance (SQA) organization for an Anomaly Report (ANR). A review meeting should be held according to the quality assurance procedure. It should focus on an objective quality goal as quality management in the system test phase. After a review of the system’s test results, the SQA should inspect and follow-up the test results with a checklist as to whether a system test will achieve enough of the original objectives. It should also determine whether the system test level was proper for the physical configuration audit and performance based audit by the SQA. After comparing the expected value and the resulting value by the SQA, it is necessary that they take the required actions. We have used a confirmation of the results of the SQA for the system test results. One can
use the software qualification results for the physical configuration audit and the performance based audit by the SQA. Software Quality Assurance activities have been performed during the software life cycle, as shown in Table 2.

**Table 2. Software Quality Activities under a System QA Program**

<table>
<thead>
<tr>
<th>SWLC</th>
<th>Software Development Baselines</th>
<th>Recommended Software Quality Assurance Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>SQAP, SVVP, SRS</td>
<td>. SRS review .</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. In-process audit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. Software Test Plan Review</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. Managerial Review</td>
</tr>
<tr>
<td>Design</td>
<td>Preliminary Design Description</td>
<td>. Preliminary Design Review</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. In-process Audit</td>
</tr>
<tr>
<td>Implementation</td>
<td>Code Listing, Other Documentation(Code Implementation Specification)</td>
<td>. Detailed Design Review</td>
</tr>
<tr>
<td>Test</td>
<td>Test Documentation</td>
<td>. In-Process Audit</td>
</tr>
<tr>
<td>Installation and Checkout</td>
<td>Deliverable items, Installation Report etc., SVVR</td>
<td>. Physical Configuration Audit</td>
</tr>
<tr>
<td></td>
<td>User Documentation</td>
<td>. Performance-based Audit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. User Documentation Review</td>
</tr>
</tbody>
</table>


The division of responsibilities among the System-Level Quality Assurance and Software-Level Quality Assurance (SQA) are that the former primarily focuses on quality assurance criteria and quality assurance procedure in the upper level, whereas the latter implements a review and audit from the technical perspective in the lower level. The classification in the audit consists of a Functional Configuration Audit, Physical Configuration Audit, and In-Process Audit. A Functional Configuration Audit compiles safety-grade optical modem software, whether or not it meets the requirements in testing level from functional entities (logical viewpoint). Although an In-Process Audit is largely applicable in all of the software lifecycle, it is applied in the design and implementation phases. Physical Configuration Audit and Performance-based Audit are planned to be applied in the release phase (final phase). A review performs a requirement phase, design phase, and user documentation to ensure the quality assurance criteria and quality assurance procedure.
3.2. Software Configuration Management

Software Configuration Management (SCM) under a software quality assurance policy is performed during the entire software lifecycle. Inconsistencies among the software configuration items were found to be the date and revision number of the software documents and source codes. Some of the reported anomalies have been resolved throughout the software configuration management process. Figure 7 shows an example of a SCM process.

![Figure 7. Software Configuration Management by NuSCM Tool](image)

4. Conclusions

Our safety-critical software qualification methodology through the projects was well established. The toolsets used was a self-developed one and a commercially available one. The technique has taken advantage of qualification techniques that use Code and Standard criteria, and requirement traceability analysis with verification testing techniques. We investigated software qualification processes, i.e., the requirements, design, implementation, and test phase, using the proposed well-structured qualification organization and qualification methodologies. Our major qualification techniques are a requirement traceability analysis, formal verification testing as a verifier, software configuration management under the quality assurance system including the integration, and a system test. The applied qualification methodology satisfies the SRP/BTP-14 criteria for a safety-critical system in nuclear power plants.

All tests were performed according to the test plan and test procedures. Functional testing, performance testing, event testing, and scenario based testing for a safety-grade optical modem of a Core Protection Calculator in a Korea Standard Nuclear Power Plant as a thirty-party verifier were successfully performed. We confirmed that the coverage criterion for a safety-grade optical modem of a Core Protection Calculator was satisfactory using a traceability analysis matrix between high-level requirements and a lower-level system test case data set. The qualification methodology and our experience are going to be continually upgrading in upcoming projects.
Acknowledgements

This work, described herein, is being performed for “Development of the Suitability Assessment on nuclear I&C, Surveillance and Early Response” as part of the Korea Atomic Energy Research Institute (KAERI) projects and funded by the Korean Ministry of Science and Technology since on January the 1st, 2012.

References


Copyright © 2014 SERSC
Authors

**Jangyeol Kim**, he received Ph.D. degrees from University of CHUNG-ANG, Seoul, Korea, in Computer Science in 1994. He has been working as an Principal Researcher in Korea Atomic Energy Research Institute (KAERI) from March 1985. His research interests include Distributed Operating System, System Programming, Embedded System, Safety-critical Software Engineering, Software Qualification, Hardware Qualification, Commercial Off-the-Shelf Software Dedication, Quality Assurance, and Safety Analysis, and so on.

**SoonGohn Kim**, he received Ph.D. degrees from Chonbuk National University, Seoul Korea, in Computer Engineering in 1999. He has been working as a Professor in Joongbu University from March 1995. His research interests include Ubiquitous Computing, Distributed Computing, Database Integrity, Cryptographic Protocol, A methodology of Software Development, Software Evaluation, Networks, and so on.