Proceedings of the Korean Nuclear Spring Meeting
Gyeong ju, Korea, May 2003

The KNICS Approach for Verification and Validation of Safety Software

Kyung Ho Cha, Han Seong Sohn, Jang Soo Lee, Jang Yeol Kim, Se Woo Cheon, Lee Young Joon, In Koo Hwang, and Kee Choon Kwon

Korea Atomic Energy Research Institute
150 Dukjin-Dong, Yusong-Gu,
Daejon, 305-353, Korea

Abstract

This paper presents verification and validation (VV) to be approached for safety software of POSAFE-Q Programmable Logic Controller (PLC) prototype and Plant Protection System (PPS) prototype, which consists of Reactor Protection System (RPS) and Engineered Safety Features-Component Control System (ESF-CCS) in development of Korea Nuclear Instrumentation and Control System (KNICS). The SVV criteria and requirements are selected from IEEE Std. 7-4.3.2, IEEE Std. 1012, IEEE Std. 1028 and BTP-14, and they have been considered for acceptance framework to be provided within SVV procedures. SVV techniques, including Review and Inspection (R&I), Formal Verification and Theorem Proving, and Automated Testing, are applied for safety software and automated SVV tools supports SVV tasks. Software Inspection Support and Requirement Traceability (SIS-RT) supports R&I and traceability analysis, a New Symbolic Model Verifier (NuSMV), Statemate MAGNUM (STM) ModelCertifier, and Prototype Verification System (PVS) are used for formal verification, and McCabe and Cantata++ are utilized for static and dynamic software testing. In addition, dedication of Commercial-Off-The-Shelf (COTS) software and firmware, Software Safety Analysis (SSA) and evaluation of Software Configuration Management (SCM) are being performed for the PPS prototype in the software requirements phase.

1. Introduction

Nowadays digital technology has been applied rapidly to design instrumentation and control (I&C) systems for railway, airplane, vehicle, communication network, etc. According to the rapid trends in digital technology, nuclear I&C systems are being designed with digital components or equipment, including Programmable Logic Controller (PLC). It is very interested in PLC-based digital safety system platforms and representative prototypes include Teleperm XS system, Common Q system and Tricon system [1]. Korea Nuclear Instrumentation and Control System (KNICS) has been projected domestically for developing digital I&C systems since the mid of 2000. The KNICS Safety system (i.e., plant protection system: PPS) consists of Reactor Protection System (RPS), Engineered Safety Features-

Component Control System (ESF-CCS), and the PPS prototype is being designed with POSAFE-Q PLC to be prototyped by proprietary design [2]. Software requirements specification (SRS) and software design specification or description (SDS or SDD) have been specified systematically for each system of the PPS prototype and Software Verification and Validation (SVV) Procedures (SVVPs) [15-26] have also been written systematically for software requirements, software design, software implementation, and software integration of the POSAFE-Q, RPS, and ESF-CCS prototypes. To meet regulatory requirements and design goals of the PPS prototype, they have been written systematically for each system of the PPS prototype and the SVV criteria or requirements in the IEEE Std. 7-4.3.2, the IEEE Std. 1012-1988, the IEEE Std. 1028, BTP-14, and regulatory positions are included as the way of checking each criteria or requirement. SVV techniques include reviews and the detailed review (i.e., Fagan Inspection), formal verification including model checking and theorem proving, and automated testing. Their tools include Software Inspection Support and Requirement Traceability (SIS-RT) for review and inspection, a New Symbolic Model Verifier (NuSMV) and Statemate MAGNUM (STM) ModelCertifier for model checking, Prototyping Verification System (PVS) for and theorem proving, and McCabe and Cantata++ for automated static and dynamic testing. Software Safety Analysis (SSA), Software Configuration Management (SCM) evaluation, and dedication of Commercial-Off-The-Shelf (COTS) software and firmware are executed as independent activities. Therefore, our approach shall be applied sufficiently and systematically for Design Certification (D.C.) or Topical Report (TR) of the PPS.

2. PPS Prototype

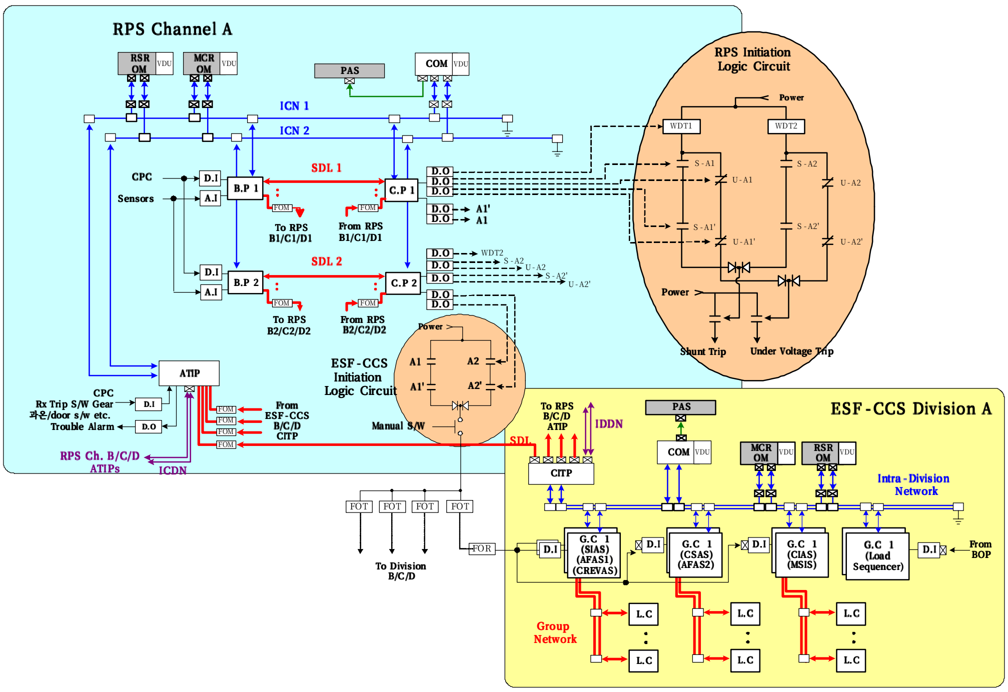

Fig. 1 The PPS Architecture in the KNICS Prototype (adapted from [3])

Fig. 1 shows the networked system architecture of the PPS Prototype and thick lines represent safety-critical data communication network.. The PPS prototype consists of RPS prototype and ESF-CCS prototype, and qualified (i.e., Q-class) POSAFE-Q PLC prototype is developed and applied for their design. The RPS prototype consists of 4-channels including Bistable Processor (BP), Coincidence Processor (CP), Automatic Test and Interface Processor (ATIP), Cabinet Operator Module (COM), and communication networks including Safety Data Links (SDLs), etc. The ESF-CCS prototype consists of 4-channels including Group Controllers (GCs), Loop Controllers (LCs), Communication Test and Interface Processor (CTIP), Cabinet Operator Module (COM), and communication networks, etc. According to its safety classification, corresponding software also are classified as safety-critical software. The PPS prototype is designed with PLCs for its computing and POSAFE-Q is being developed for the PPS prototype. PES (Process Engineering Station) is utilized for developing applications of RPS and ESF-CCS prototype and for downloading them to target system (i.e., POSAFE-Q PLC). The POSAFE-Q PLC includes the different software systems including the proprietary pCOS kernel as Real-Time Operating System (RTOS) kernel, the proprietary Operating System Software (OSS) over the pCOS kernel, and RPS and ESF-CCS applications (i.e., PLC diagrams) to be controlled by the OSS. The PPS prototype consists of 4-channels to be replicated and safety software are also replicated over each channel. Safety software are executed as online and summarized as shown in Table 1.

Table 1 Safety Classification of Software for the RPS Prototype

H/W \ S/W	Name	Main Features	Safety Class.	Dev. Type
POSAFE-Q PLC	pCOS	is RTOS kernel including Basic Interrupt Handling System, supports multitasking, and provides semaphore, mailbox, queue, etc.	SC	Proprietary (P)
	OSS	operates under pCOS and includes PES, Network, User Schedule, Sequence Solver, Display, and I/O agents	SR	
	Com. SW	is communication protocols and drivers, and supports data transmission between channels (i.e., intra - and/ or inter -networks)	SR, partly SC	P & COTS
RPS	BP SW	acquires sensor signals, compares them with trip setpoints, and transmits the results to CPs	SC	P
	CP SW	activates reactor trip and ESF actuation signals, and changes to automatically 2/3 voting logic for trip channel bypass.	SC	
	COM SW	monitors and controls operating status of RPS, is installed on MCR and RSR	Non -Safety	
	ATIP SW	supports on -line testing and monitors RPS status, and interfaces with ESF -CCS	SR	
ESF -CCS	GC SW	executes voting logic of ESF -CCS actuation, transfers the voting results to LCs, and monitors LC status	SC	P
	LC SW	controls and monitors engineered components, and monitors process control parameters	SR	
	COM SW	monitors and controls operating status of ESF -CCS, is installed on MCR and RSR	Non -Safety	
	CTIP SW	supports on -line testing and monitors ESF -CCS status, exchanges infor. between divisions, and interfaces with RPS	SR	

The software requirements are specified by natural language and partially formal languages. Formal specifications are scoped for important functional requirements including safety features, timing, interface, etc. Table 2 shows SRS and SDS methods of safety critical software for the PPS prototype.

Table 2 SRS and SDS Methods for the PPS Prototype

Safety-Critical SW / SW Documents	pCOS kernel & OSSfor POSAFE-Q PLC SW for Safety Data Link (SDL) Loop Controllersfor ESF-CCS	Bistable Logic & Coincidence Logic for RPS
Software Requirements Specification (SRS)	Korean Language Statecharts	Korean Language NuSCR
Software Design Specification (SDS or SDD)	Korean Language Statecharts	Korean Language NuSDS (Function Block Diagram: FBD)

NuSCR [4], in which timed-automata are introduced, is used for specifying software requirements of the RPS prototype. The timed-automata are very efficient for specifying timing requirements of reactive systems including reactor shutdown system, etc. Fig. 2 shows the NuSCR specification for a trip function of RPS prototype.

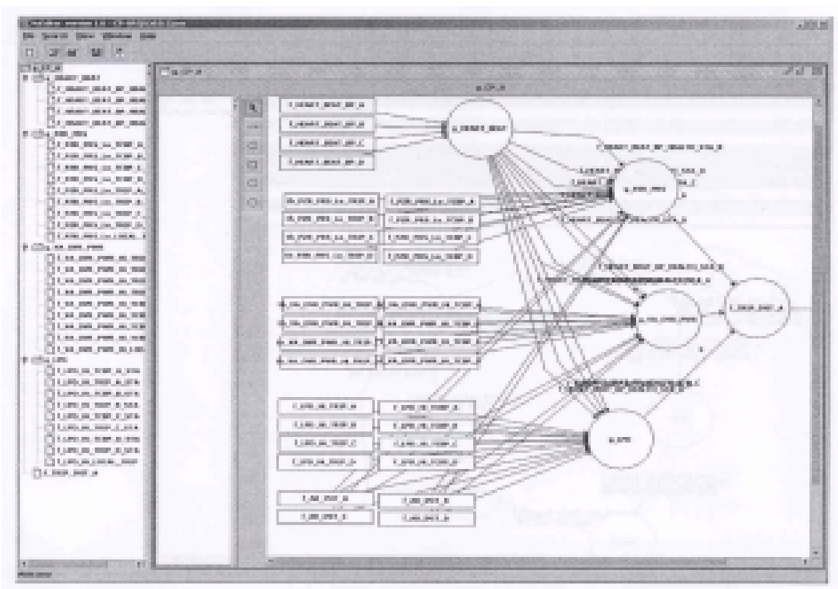

Fig. 2 Example of Formal SRS Using NuSCR

The other formalism, known as statecharts, is used for specifying software requirements for the ESF-CCS prototype and the POSAFE-Q prototype. Fig. 3 represents the statecharts specification for state transitions of pCOS tasks.

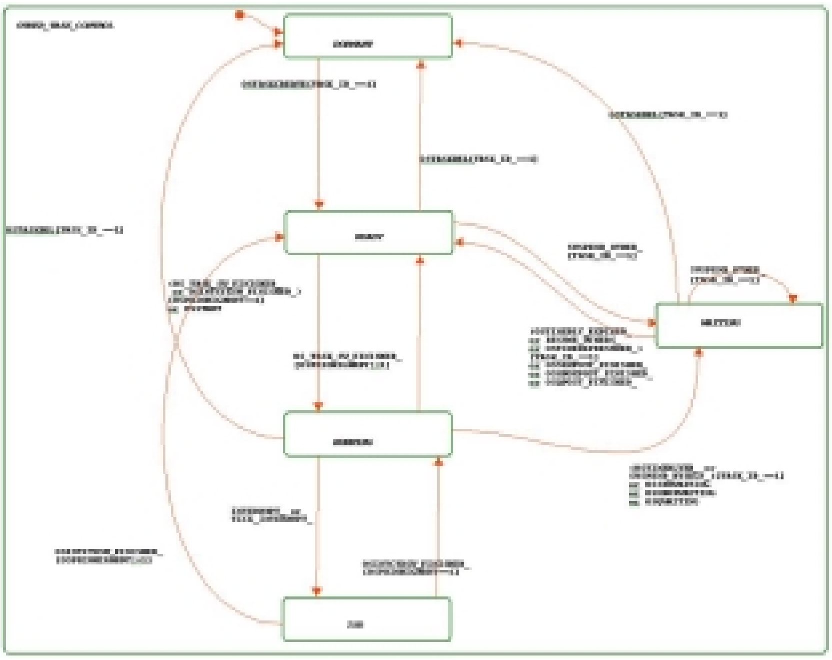

Fig. 3 Example of Formal SRS Using Statemate MAGNUM

POSAFE-Q software is programmed by ANSI C and TI assembly languages while RPS software and ESF-CCS software are implemented by PLC programming to be followed IEC 61131-3 standards [5] on Instruction List (IL), Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Functional Chart (SFC) and Structured Text (ST). Process Engineering Station (PES) provides these programming environments which include the program download into target machine (i.e., POSAFE-Q PLC) and configuration for environment of the PLC system. With the aim of reducing coding errors, programmer's guidelines are being written for safety software of the PPS prototype. Software testing is regarded as the one of SVV activities for the PPS prototype. Life-cycle testing is applied for software of the PPS prototype and testing tools are utilized for them. Representative testing includes unit or module testing, integration or subsystem testing, and system testing. Acceptance testing is to be planned for the final products to be developed. Regulatory Guide 1.171 and IEEE Std. 1008-1987 are referenced as criteria or requirements for unit testing while Regulatory Guide 1.168, IEEE Std. 1012-1992 and IEEE Std. 1059-1993 are referenced for SVV [6]. Testing tools are supported for each phase and they are introduced on the section 3.5.

3. SVV Approach to the PPS prototype

SVV tasks have been established for safety software of the PPS prototype and Fig. 4 illustrates the SVV tasks for SRS and SDS including formal specifications, and codes. The SVV evaluates that software requirements are satisfied at each stage of software development life-cycle (SWLC) and it also identifies that design results are consistent, complete, and correct with the ones in the previous stage of SWLC processes.

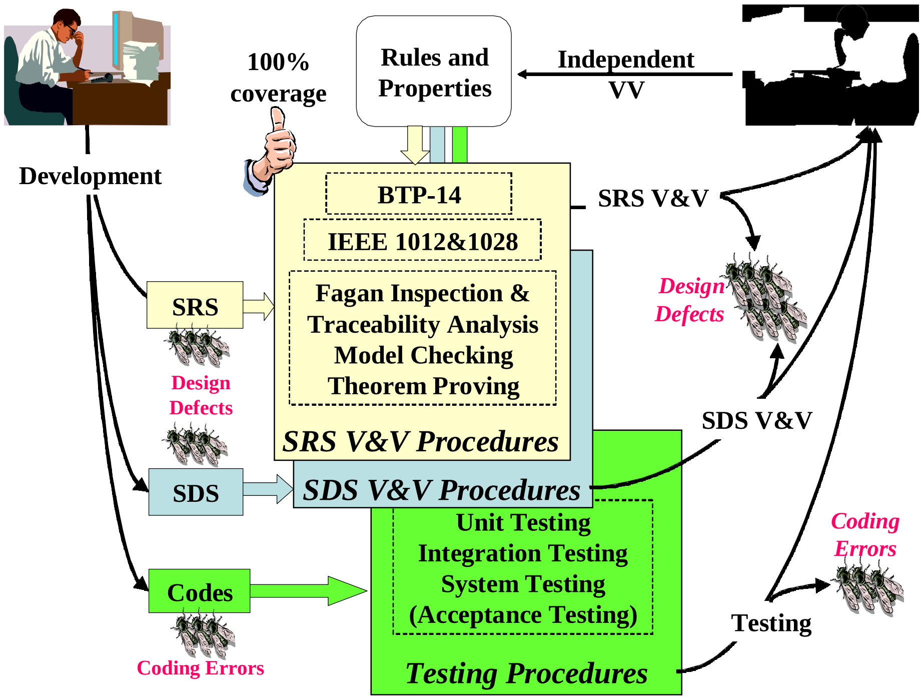

Fig. 4 SVV Tasks for the PPS Prototype

The SVV is performed systematically with well-designed SVV procedure (SVVP). Review and inspection (R&I) using well-defined checklists, formal verification techniques such as model checking and theorem proving, and validation and testing with the supports of automated tools are detailed and guided within the SVVPs. The use of SVVP is regarded as the way of validating and verifying safety software systematically due to well-defined SVV tasks and processes, and checklists. For the second year of the KNICS project, the drafted SVVPs include the SVVPs for software development of POSAFE-Q PLC [15, 18, 21, 25], the SVPP for software development RPS [16, 19, 22, 25], and the SVVP for ESF-CCS [17, 20, 23, 26]. In the realm of safety-critical software for safety systems, software testing has been executed rigorously although high cost and men hours are required for testifying safety-critical software. An example is the case of Sizewell B plant under commercial operation in U.K. Software unit or module testing, sub-system or integration testing, and system testing are planned for safety software of the PPS prototype. COTS software (i.e., Profibus protocol) is partly considered for communication units of POSAFE-Q PLC and qualification of COTS software is covered with COTS dedication plan and its procedure. In addition, SSA and SCM evaluation will be performed independently. SVV Procedures (SVVPs) have been developed for verifiers of safety software and the SVVPs are applied for the life-cycle SVV of POSAFE-Q prototype, RPS prototype and ESF-CCS prototype. Fig. 5 shows the development strategy and application scope to the selected checklists for the SVVPs of the PPS prototype.

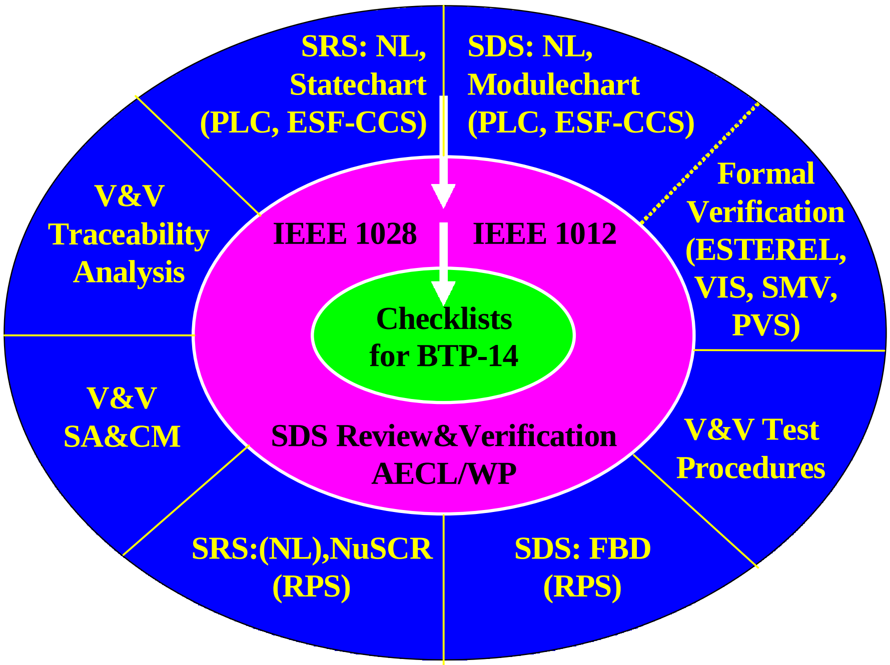

Fig. 5 Development strategy and Application Scope of the SVVP Checklists

The SVVPs contain checklists for acceptance criteria of digital computer-based I&C systems to be defined in the Appendix 7-A Branch Technical Positions (BTP) HICB-14 of NUREG-0800 Standard Review Plan (SRP) [7], checklists for the IEEE requirements for software VV plan [8] and software reviews [9], and checklists for formal software specifications by NuSCR and statecharts. The SVVP for SRS consist of the review procedure and the detailed verification procedure. The review procedure includes checklists for functional characteristics and process characteristics while the detailed verification procedure includes the detailed checklists for software requirements traceability, correctness, consistency, and completeness. The SVVPs for software design, implementation and integration of the POSAFE-Q PLC prototype, the RPS prototype, and the ESF-CCS prototype have been developed and each SVVP has its specialized SVV procedures including checklists. The SVV techniques and their tools to be referred on the SVVPs are described on the next section. Review and Inspection (R&I) are representative and popular VV techniques and formal methods are being applied for design verification. Several kinds of SVV tools have been developed for recent decade and the SVV is being more effective. Table 3 summarizes SVV techniques and tools for safety software of the PPS Prototype.

Table 3 SVV Techniques and Tools for the PPS Prototype

Class. / System	SRS		SDS		Implementation		Integration	
	Technique	Tool	Technique	Tool	Technique	Tool	Technique	Tool
POSAFE-Q	• Review & Inspection • Model Checking	• SIS-RT • Statemate MAGNUM Model Certifier & Simulator	• Review & Inspection • Model Checking	• SIS-RT • Statemate MAGNUM Model Certifier & Simulator	• Code Inspection • Static & Dynamic Testing	• McCabe • Cantata++	• Code Inspection • Static & Dynamic Testing	• McCabe • Cantata++
RPS	• Review & Inspection • Model Checking • Theorem Proving	• SIS-RT • NuEditor • NuSMV • PVS	• Review & Inspection • Model Checking • Theorem Proving	• SIS-RT • FBDEditor • NuSMV • PVS	• Fagan Inspection • Static & Dynamic Testing	• (TBD) -McCabe -Cantata++	• Fagan Inspection • Static & Dynamic Testing	• (TBD) -McCabe -Cantata++
ESF-CCS	• Review & Inspection • Model Checking	• SIS-RT • Statemate MAGNUM Model Certifier & Simulator	• Review & Inspection • Model Checking	• SIS-RT • Statemate MAGNUM Model Certifier & Simulator	• Fagan Inspection • Static & Dynamic Testing	• (TBD) -McCabe -Cantata++	• Fagan Inspection • Static & Dynamic Testing	• (TBD) -McCabe -Cantata++

Review is defined as "an evaluation of software element(s) or project status to ascertain discrepancies from planned results and to recommend improvement" in the IEEE Standard 1028 [8]. A number of review methods including technical reviews, management reviews, one third presentation, and walkthroughs are known [10, 11] and technical reviews are applied mainly for the SVV of the PPS prototype. For the more formal reviews, inspection tends to be combined with the reviews. Fagan Inspection (F.I.) is used for the SVV of the PPS prototype and automated tool, to be called as SIS-RT [12]. The SIS-RT is being developed in the Dept. of Nuclear and Quantum Physics of KAIST and provides structural view, inspection view, and traceability view for R&I. R&I are also applied to source codes and the R&I may use different inspection tool to be suited for source codes. Traceability analysis is required for software architecture design stage and detailed design stage. Specific inspection, analysis, or testing technique is used to determine whether software elements are designed correctly or not through traceability analysis at the stage of software architecture design. Model checking and theorem proving are formal techniques for verifying software quality properties using mathematical methods and it is applied for improving safety of high-integrity systems. They are mainly applied for formal SRS, formal SDS, and source codes. For the automated model checking and theorem proving, a New Symbolic Model Verifier (NuSMV) and Prototype Verification System (PVS) theorem prover shall be applied for the formal software requirements and design to be specified using NuSCR and STM ModelCertifier is used for the formal software requirements and design to be specified by statecharts. In addition, VIS and ESTEREL tools are being evaluated for formal verification. Software Testing Life-Cycle (STLC) process includes unit or module testing, sub-system or integration testing, system testing, and acceptance testing. Fig. 6 shows the STLC processes following Software Development Life Cycle (SDLC) processes.

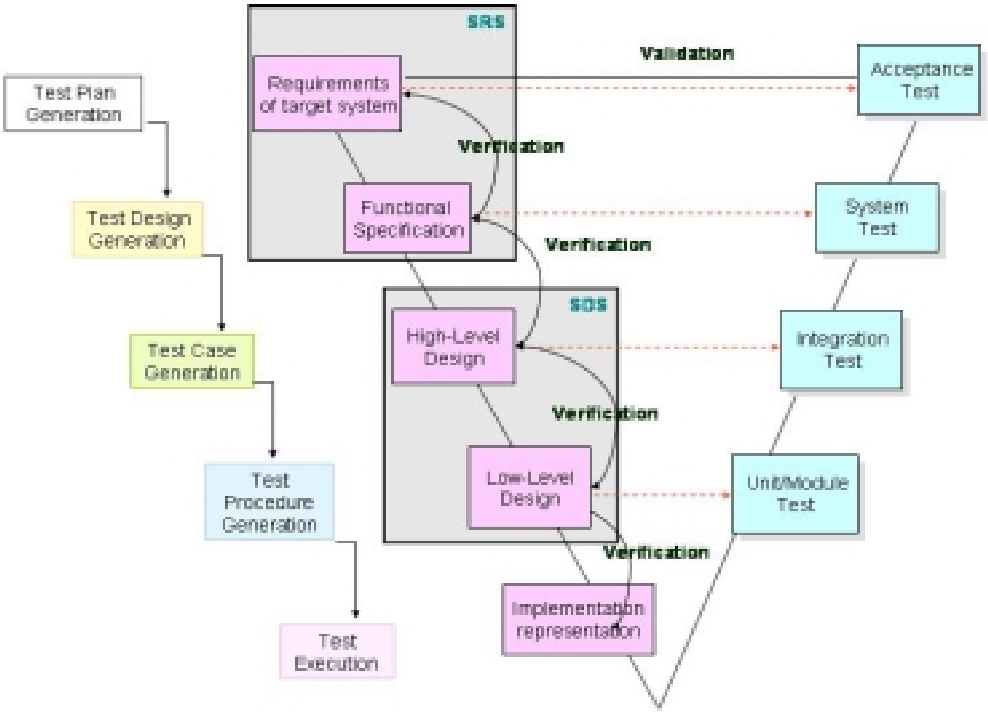

Fig. 6 Software Testing Life-Cycle Processes

Each STLC process includes test tasks such as test plan generation, test design generation, test case generation, test procedure generation, and test execution. Reg. Guide 1.168, IEEE Standard 1059-1993, and IEEE Standard 1012-1998 are applied for software VV, Reg. Guide 1.171 and IEEE Standard 1008-1987 are applied for software unit testing, and Reg. Guide 1.170 and IEEE Standard 829-1998 are applied for software documentation of the PPS prototype [5]. Software testing techniques include boundary value analysis, regression or mutation test, etc. State-based testing, logic-based testing, and fault-injection testing are specially studied for software testing of the POSAFE-Q PLC prototype [13]. To support software testing of the PPS prototype, McCabe QTM and Cantata++ testing tools have been reviewed for structured testing and dynamic testing. Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment [14] shall be applied for dedicating COTS software of the PPS prototype due to commercial protocol firmware for communication units of the POSAFE-Q prototype. Fig. 7 shows the dedication processes including planning, selection of QA items from ASME/NQA-1, third-party (software supplier) evaluation, writing audit reports, and selection of dedication methods including Commercial Grade Survey (Method 2) and Operating Experience Data (Method 4) to be defined in the EPRI TR-106439 [14].

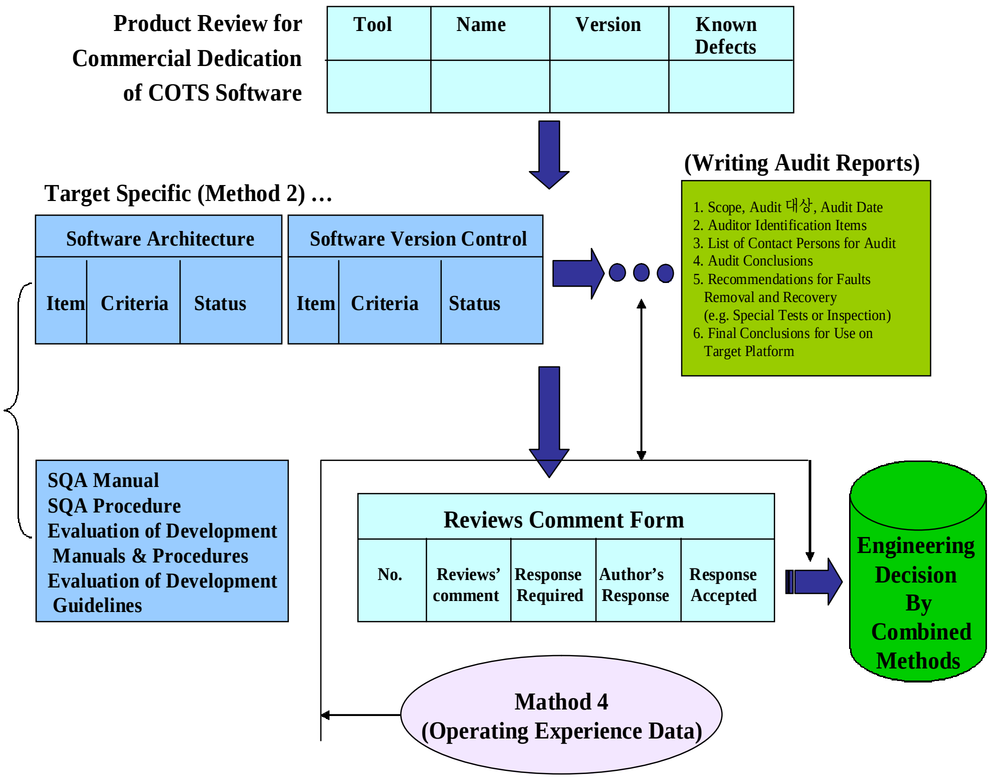

Fig. 7 Dedication Processes of COTS Software for the PPS Prototype

In addition to the SVV, SSA and SCM evaluation are also applied for the overall SDLC processes of the RPS prototype, ESF-CCS prototype, and the POSAFE-Q Prototype. The SSA is performed for software requirements, software architecture and detailed design, code, testing, and changes to be defined in the Software Safety plans of each prototype [27, 28, 29]. SCM follows the SCM plan of each prototype [30, 31, 32] and the SCM is applied for design documents, data, and software (i.e., program code). To support the SCM, SCM tool has been developed and is being applied for controlling the SCM items [33]. SQA is being planned and will be applied for the 2nd phase of the KNICS project.

4. Conclusions

This paper has described an SVV approach for the RPS, the ESF-CCS, and the POSAFE-Q PLC to be prototyped through the KNICS project. The SVV approach considers both regulatory positions on safety software and design features of safety software for the PPS prototype. To meet regulatory requirements and design goals of the PPS prototype, SVV procedures have been prepared systematically for each system of the PPS prototype. SVV criteria or requirements are selected from the IEEE Std. 7-4.3.2, the IEEE Std. 1012-1988, IEEE Std. 1028, and BTP-14, and the selected ones are included as format of checklists. SVV technique with the support of its tool includes review and the detailed review (i.e., Fagan Inspection) using SIS-RT tool, model checking using NuSMV and STM ModelCertifier, theorem proving using PVS, and automated software testing using McCabe and Cantata++. SSA, SCM evaluation, and dedication of COTS software and firmware would be performed as independent activities. As a result, the SVV approach is regarded as supporting design certification or topical report of the PPS. Future work will be continued on SVV for final product.

Acknowledgements

This work, described herein, is being performed for "Development of the Licensing Technology for Digital I&C" as part of the KNICS project and funded by the Korean Ministry of Science and Technology since the mid of 2000.

- Statemate MAGNUM (STM) is the trademark by I-Logic Inc.
- McCabe is the trademark by T.J McCabe and Associates, Inc.
- Cantata++ is software testing tool and is trademarked by IPL Inc.
- POSAFE-Q is Q-class PLC platform to be prototyped by POSCON Inc.
- SIS-RT supports software requirements traceability and has been developed at KAIST.
- NuSCR is formal specification system to be prototyped at KAIST.
- Statemate MAGNUM (STM) ModelCertifier is the trademark by I-Logic Inc.

References

[1] J. Naser, "Qualification, Acceptance, and Implementation of Programmable Logic Controller-Based Platforms for Safety-Related Applications in Nuclear Power Plants," Proceedings on NPIC&HMIT 2000, Washington, DC, November, 2000.

[2] KAERI, "Project Proposal for Development of the Digital Reactor Safety System in the KNICS," July 1, 2001. (In Korean)

[3] Chang H. Kim, "Development of Plant Protection System and Digital Controller," Proceedings of The First KNS-KIEE Joint Workshop on I&C Technology, Seoul Educational Culture Convention Center, Seoul, November 7, 2001, pp. 29-40.

[4] Y. J. Oh, J. M. Cho, J. B. Yoo, S. D. Cha, "A Technique to Specify and Analyze reactive and real-Time Software," -, KAIST, 2002.

[5] IEC 61131-3, "Programmable controllers - Part 3: Programming languages," 1993.

[6] Kyung Ho. Cha, Jang Soo Lee, Kee Choon Kwon, "Regulatory Requirements Analysis for Software Development of Digital Safety Systems," Proceedings of The First KNS-KIEE Joint Workshop on I&C Technology, Seoul, Korea, Nov. 7, 2001, pp. 59-68.

[7] USNRC, BTP-14, Guidance on Software Reviews for Digital Computer-based I&C Systems. July 1997.

[8] IEEE Std. 1012-1992, IEEE Standard for Software Verification and Validation Plans.

[9] IEEE Std. 1028-1988, IEEE Standard for Software Reviews and Audits.

[10] G.J. Kurt Asmis, "Inspection of Safety Critical Software: Expectations of the Regulator," Proceedings of the 1st Workshop on Inspection in Software Engineering (edited by M. Lawford and D.L. Parnas), 2001.

[11] S. Golubic and D. Marusic, "Reviews and Inspections – An Approach to the Improvement of Telecom Software Development Process," Proceedings of the 1st Workshop on Inspection in Software Engineering (edited by M. Lawford and D.L. Parnas), 2001.

[12] Seo Ryong Koo, et al., "Development of Software Requirement Analysis Tool for NPP Software Fields Based on Software Inspection and Formal Method," Proceedings of International Symposium on the Future I&C for NPP, Seoul, Korea, Nov.7-8, 2002, pp.159-164.

[13] B. J. Choi, A. Y. Seong, and J. S. Lee, "A Study on Verification and Validation Methodology for Improving Licensing Acceptability Using Formal Methods and Testing," Activity Report, Dec., 2001. (In Korean)

[14] EPRI Topical Report TR-106439, "Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications." Electric Power Research Institute, October 1996.

[15] KAERI, "SVVP for SRS of POSAFE-Q PLC," KNICS-SED-SVP121 (Rev.00), Jan. 14, 2002. (In Korean)

[16] KAERI, "SVVP for SRS of RPS," KNICS-RPS-SVP121 (Rev. 00), Jan. 14, 2003. (In Korean)

[17] KAERI, "SVVP for SRS of ESF-CCS," KNICS-ESF-SVP121 (Rev. 00), Jan. 14, 2003. (In Korean)

[18] KAERI, "SVVP for SDS of POSAFE-Q PLC," KNICS-SED-SVP131 (Draft), 2003. (In Korean)

[19] KAERI, "SVVP for SDS of RPS," KNICS-RPS-SVP131 (Draft), 2003. (In Korean)

[20] KAERI, "SVVP for SDS of ESF-CCS," KNICS-ESF-SVP131 (Draft), 2003. (In Korean)

[21] KAERI, "SVVP for Software Implementation of POSAFE-Q PLC," KNICS-SED-SVP141 (Draft), 2003. (In Korean)

[22] KAERI, "SVVP for Software Implementation of RPS," KNICS-RPS-SVP141 (Draft), 2003. (In Korean)

[23] KAERI, "SVVP for Software Implementation of ESF-CCS," KNICS-ESF-SVP141 (Draft), 2003. (In Korean)

[24] KAERI, "SVVP for Software Integration of POSAFE-Q PLC," KNICS-SED-SVP161 (Draft), 2003. (In Korean)

[25] KAERI, "SVVP for Software Integration of RPS," KNICS-RPS-SVP161 (Draft), 2003. (In Korean)

[26] KAERI, "SVVP for Software Integration of ESF-CCS," KNICS-ESF-SVP161 (Draft), 2003. (In Korean)

[27] KAERI, "Software Safety Plan for POSAFE-Q PLC," KNICS-SED-SVP107 (Rev.00), Jan. 14, 2003. (In Korean)

[28] KAERI, "Software Safety Plan for RPS," KNICS-RPS-SVP107 (Rev.00), Jan. 14, 2003. (In Korean)

[29] "Software Safety Plan for ESF-CCS," KNICS-ESF-SVP107 (Rev. 00), Jan. 14, 2003. (In Korean)

[30] KAERI, "Software Configuration Management Plan for POSAFE-Q PLC," KNICS-SED-SVP107 (Rev. 00), Jan. 14, 2003. (In Korean)

[31] KAERI, "Software Configuration Management Plan for RPS," KNICS-RPS-SVP107 (Rev. 00), Jan. 14, 2003. (In Korean)

[32] "Software Configuration Management Plan for ESF-CCS," KNICS-ESF-SVP107 (Rev. 00), Jan. 14, 2003. (In Korean)

[33] Bo-Gyun Byoun, et al., "A Design of Project-based Software Configuration Management System," Proceedings of International Symposium on the Future I&C for NPP, Seoul, Korea, Nov.7-8, 2002, pp.165-168.