2016 International Conference on Platform Technology and Service (PlatCon)

Proceedings

15-17 February 2016
Jeju, Korea

IEEE Catalog Number: CFP16F03-ART (Xplore)
ISBN: 978-1-4673-8685-2 (Xplore)

IEEE Catalog Number: CFP16F03-CDR (CD)
ISBN: 978-1-4673-8684-5 (CD)

Mapping SW Development Process with Safety Process for Safe Software

So Young Moon
SELab., Dept. of Computer and Information ommunication
Hongik University
Sejong Campus, 30016, Korea
msy@selab.hongik.ac.kr

R. Young Chul Kim
Dept. of Computer and Information Communication
Hongik University
Sejong Campus, 30016, Korea
bob@hongik.ac.kr

***Abstract*— Today, it does control and manage hardware depended on software in most of industry (automobile, ship, airplane, nuclear power, defense, communication). Due on this, software errors cause great damage in all industry, which becomes the rise of safe software issue. All software development companies are focused on develop and test software for high quality, and make efforts to get certificate of software quality. But any previous quality certificate cannot deal with either safe software or software safety. For safe software, it makes the minimum of software error for safe software, and is necessary to make predication, sensing, monitoring for an accident. To solve this problem, we suggest to mapping software development process with safe process. This approach makes a safety based multiple development process to extract and remove software risk elements on safe software development. This approach may possibly develop safe software very well, and to predict, sense, and monitor even a critical accident.**

Keywords*—*Software Development Process; Safety Process; Safe Software

I. INTRODUCTION

Even in Korea, there is a big issue, Software Centric Society, which is on an increasing trend to control with software in most industry fields. Specially, the automobile industry focuses on developing the unmanned vehicle. In May 2015, it is published about the frequency number of occurrence of the traffic accidents of the Google's unmanned vehicle which have happened eleven times of a trivial car accident during 27,300,000 kilometer. That is, it may assume to happen approximately 11 cases of traffic accidents per one hundred million kilometer [1]. Also even not in the automobile industry, Google and Apple are manufacturing together to develop the smart car and unmanned vehicle. In this time, we can control and mange hardware with software. The age of controlling only with hardware has ended in many fields and the age has come where hardware is controlled and managed by SW. Thus, this implies that SW is used in all fields in the Software Centric Society. However, it is the reality that plans for awareness of safety SW are not prepared. Today, big accidents due to incomplete safety SW and casualties due to SW errors frequently occur. For example, in the 2009 Washington metro collision, a moving train collided with a train that stopped ahead of it. The cause was because the following train could not be stopped due to a system error and the emergency brake also did not operate. In 2014, aircraft taking-off and landing was temporarily halted due to a system cease due to overload occurrence in the procedure of the US LA air control center system calculating the altitude and velocity of the U2 reconnaissance plane. Incomplete SW safety always has the risk of expanding to a big accident. Bugs and errors in implementation and wrong requirements cause software accidents. When we design safe software, we need to review safe management, complexity, and various systems. To solve this problem, it is considered that visualization of risk elements is more significant.

In this paper, development is conducted according to SW development process, SW development process and safety process are mapped to extract and remove SW risk factors to secure SW safety. In Chapter 2, SW development process and safety process are explained. In Chapter 3, the method of mapping SW development process and safety process to secure safety SW is proposed. In Chapter 4, the conclusion and future research are mentioned.

II. RELATED WORK

A. SW Development Process

Well known SW development processes include Waterfall Model, V-model, Incremental Process, Evolutionary Process, Spiral Model, and etc. The basic activities and steps that all processes have are composed of Requirement Analysis, Design, Implementation, Testing, and Deployment. In this paper, the V-model is mentioned. V-model is a SW development process and is an expanded form of the Waterfall Model. V-model is also called as the Verification and Validation Model. To guarantee quality of each step of SW development, the test design and test activity is performed along with the start of the project [2]. The V-model that Paul Rook first proposed was composed of 9 steps including project initiation, requirement specification, structural design, detailed design, code and unit test, integration and test, software acceptance test, maintenance, project termination, and product phase-out [3]. SW quality guarantee and cost reduction effects are gained by conducting tests in every step before implementation of mapping SW development life cycle and test steps [4]. Fig. 1 shows the initial V-model that Paul Rook proposed.

978-1-4673-8685-2/16/$31.00 ©2016 IEEE

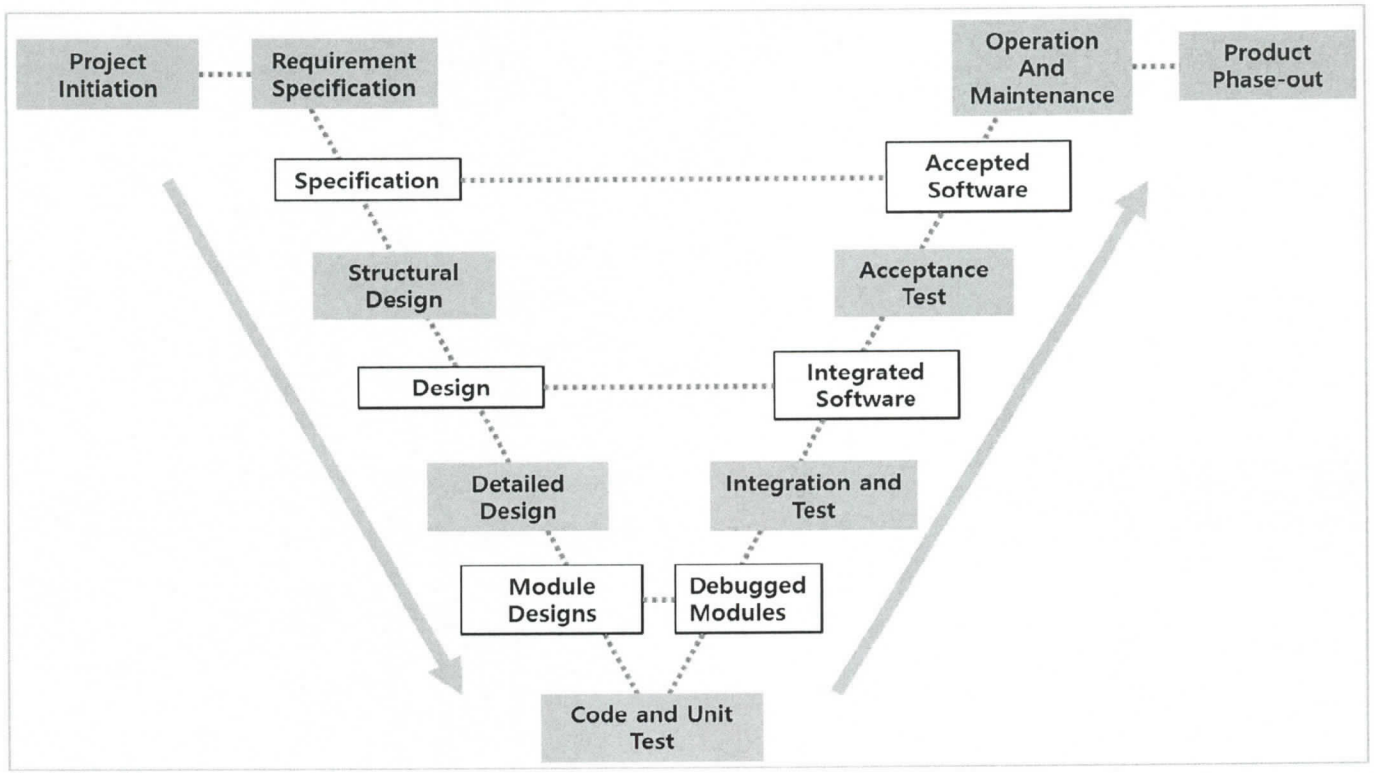

Fig. 1. V-model

- Project initiation: Plans related to projects such as main activities in each step such as milestone, resource, responsibility, and schedule are made.
- Requirement Specification: Step for analyzing requirements on SW to be developed in which demands of the SW user are analyzed. Thus, functional requirements, interface requirements, and performance requirements of the SW demanded by the user are written.
- Structural design: Overall hardware and SW architecture, control structure, data structure, initial version of user manual, and test plan are described. Technical data such as entity diagrams are calculated.
- Detailed Design: System composition, menu structure, control structure, data structure, interface, key algorithm, and etc. are described. Dependence and interface relation between database table, architecture diagram, and module are described.
- Code and unit test: Coding standards are tested and code analyzed.
- Integration and test: Module is integrated and accuracy of SW functions is reviewed by performing operation of finding errors.
- Acceptance test: Step where the customer acquires and tests.
- Operation and Maintenance: A system where all functions perfectly perform is developed by repetitively performing system operations and updates.
- Product phase-out: Estimated schedule, cost and actually developed schedule, cost are compared. Project ends.

B. Safety Process

International standard SW safety must be corresponded due to ICT convergence and the automobile function safety international standard <ISO 26262> was enforced due to SW quality issues such as the Toyota incident. Domestically, interest on quality and safety on ICT converged products such as nuclear energy, aviation, national defense, medical treatment, and shipbuilding is increasing. There are standards in each field on function safety in which this is in the contents of IEC 61508 [5] and was derived from that. In IEC 61508, all safety life cycle activities are performed through harm factor analysis, risk assessment, safety requirement development, safety requirement specification, design and implementation, and maintenance. Table. 1 shows the standard related to safety by each field. Rather than particular safety SW process, the safety process required in SW development in the total process is included. There are international standards including ISO 26262[6] for automobile fields, DO 178B [7] for aviation fields, EN 50129[8] for railroad fields, IEC 61513[9] for nuclear power fields, and IEC 60601[10] for medical care fields.

SW process for each safety SW exists in each field as shown in Table. 1, but basic SW development life cycles are maintained and there is nothing special.

Table. 1. Core Process of Safety Standard

	Industry area	Standard	Common Software related Process
IEC 61508	Automobile	ISO 26262	3. Concept phase 3-6. Hazard analysis and risk assessment 4.Product development: system level 4-5. Specification of technical safety concept 4-6. System design 4-7. Item integration and testing 4-8. Safety validation 4-9. Functional safety assessment 4-10. Product release 5. Product development hardware level 6. Product development software level 6-5. Specification of software safety requirements 6-6. Software architectural design 6-7. Software unit design and implementation 6-8. Software unit testing 6-9. Software integration and testing 6-10. Software safety acceptance testing
	Aviation	DO-178B	1. System Requirements 2. Hardware/Software 2-1. Hardware Requirements 2-2. Software Requirements 2-2-1. Preliminary Design 2-2-2. Code 2-2-3. Unit Test 2-2-4. Software Integration Testing 3. Hardware/Software Integration 4. System Testing 5. Validation Testing 6. FAI
	Railway	EN 50129	1. Lifecycle Process Development 2. Plan 3. Requirements Specifications 4. Design 5. Coding 6. Verification & Validation 7. Installation
	Nuclear Power	IEC 61513	1. Lifecycle Process Development 2. Plan 3. Requirements Specifications 4. Design 5. Coding 6. Verification & Validation 7. Installation
	Medical	IEC 60601	1. Software Development Plan 2. Software Requirement Analysis 3. Software Construction Design 4. Software Detailed Design 5. Software Implementation and Validation 6. Software Integration and Integration Testing 7. Software System Testing 8. Software Release

III. MULTI V-MODEL

Existing development process, safety process, and test process were mapped for the safety SW development process to propose a new Multi V Model. If the existing method was a process extracting requirements, the method proposed in this paper also includes the step of analysis and building strategies in which architecture and design for quality improvement and continuous safety SW integrated tests are conducted. As seen in Fig. 2, all steps in the SW development process, safety process, and test process are mapped for performance. Requirements, analysis, design, and implementation procedures are conducted in the development process. Safety requirements, safety analysis, safety design, and safety SW implementation procedures are conducted in the safety process. Test requirements, analysis, design, and implementation are conducted in the test process. SW quality factors and safety are to be secured through unit test, component test, integrated test, system test, and acquired test. Also, SW verification and validation is simultaneously performed in each step to support every process of the safety SW development. Fig. 2 shows the Multi V Model where SW development process, safety process, and test process are mapped. The SW safety process proposed in this paper is as follows.

- Step 1: For the requirement step, SW requirements, safety requirements, and test requirements are collected. When writing safety requirements, the safety behavior analysis (UBAF) method is used. After this step, SW architecture is established and the test plan is written.
- Step 2: For the analysis step, SW analysis, safety analysis, and test requirement analysis are conducted. During safety analysis, FTA and FMEA are used for description. After this step, safety test specification and test specification are written.
- Step 3: For the design step, SW design, safety design, and test design are conducted. After this step, design review and test design are written.
- Step 4: For the implementation step, safety assessment procedure and test procedure are established for SW implementation and design review. Then, test is conducted.
- Step 5: For the unit test step, unit test on the safety and unit test on implementation are conducted.
- Step 6: Component testing and safety component testing are conducted.
- Step 7: System testing on safety and system testing on SW are conducted.
- Step 8: For the acquired test, the user conducts safety and SW testing.

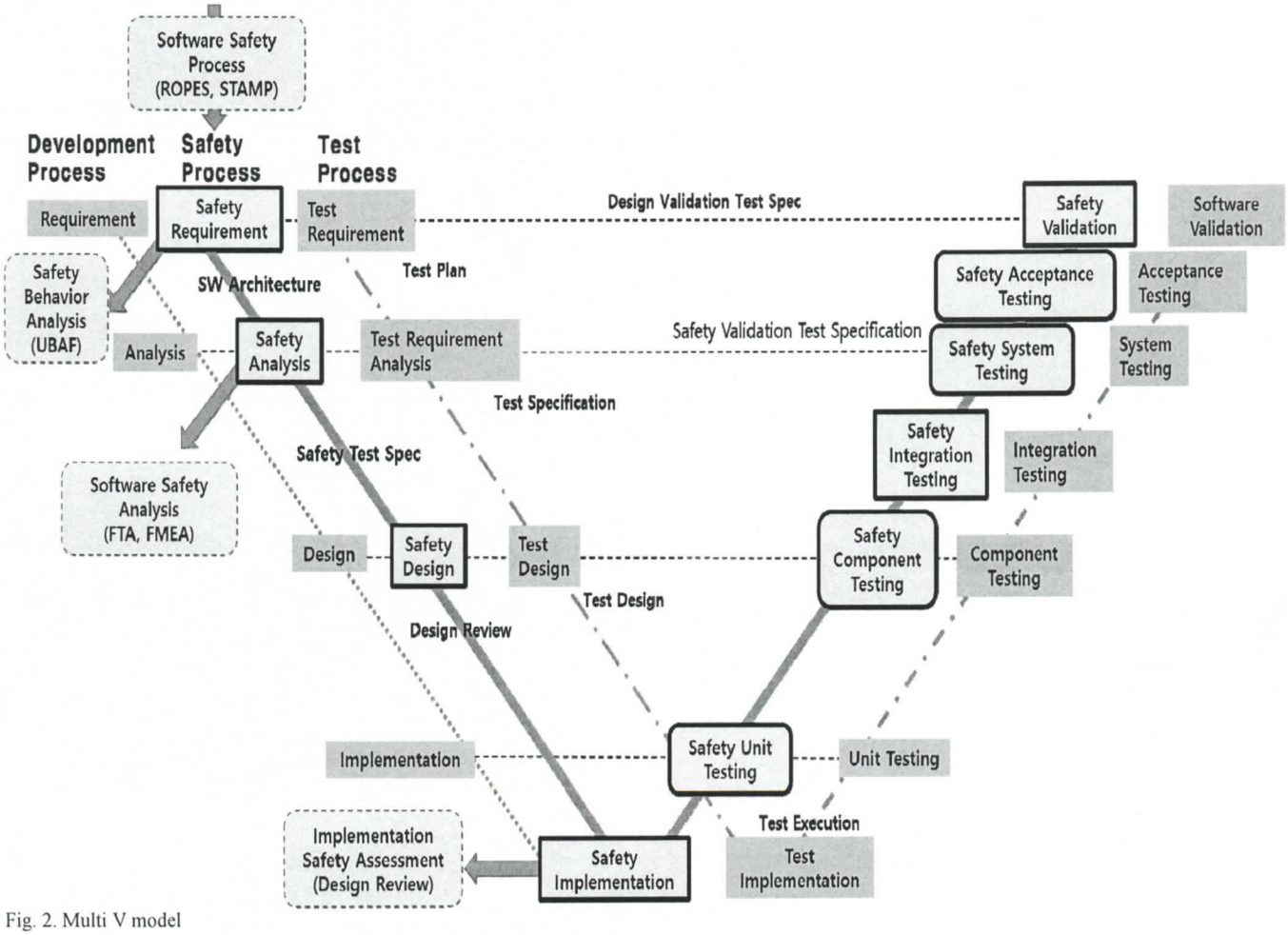

Fig. 2. Multi V model

The safety SW process proposed in this paper aims to prepare a system that can manage all steps of the test process and safety SW development by using forward engineering and reverse engineering. In step 1, requirements are extracted and analyzed to build strategies. By extracting accurate and proper requirements and based on the results, requirements are compared, deleted, and applied by judging level of significance adding blue ocean strategy. A continuous integrated model is used for quick and effective verification of safety SW and real-time verification beyond existing static model verification for developers to easily update systems by real-time.

IV. CONCLUSION

The Multi V Model proposed in this paper limits safety of SW system itself and SW supporting implementation of safe society in which the goal is continuous development and management of technology by applying the core safety analysis technology on the software development process. By distinguishing hazard/risk and defects, it was aimed to enhance safety quality. Also, a Korean type safety process was defined through user and system safety requirement defined scenario design. For future research, reliability methods (FTA, FMEA) and safety methods (STAMP, HAZOP) are focused on the existing safety critical system (nuclear power, electrical grid).The current huge sized SW and rapidly changing requirements are not enough for solving safety issues. Thus, it is limited to the quality of existing software itself. Today, operation quality is also an important issue. A past method of reliability analysis called formal specification technique required high cost and much time for solving these issues and these issues are to be solved by applying the SW visualization method [11].

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2011601) and the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2015H1C1A1035548).

REFERENCES

[1] TS Korea Transportation Safety Authority, "A Study on Analysis of Reality of Vehicle Miles Traveled (2012)", December 2013.

[2] Roger S. Pressman, "Software Engineering: A Practitioner's Approach 7th Ed", MacGrawHill, p.40-41, 2010.

[3] Paul Rook, "Controlling software projects", IEEE Software Engineering Journal, vol. 1, no. 1, p.7-16, 1986.

[4] Sonali Mathur, Shaily Malik, "Advancements in the V-Moel", International Journal of Computer Applications, vol. 1, no. 12, p.29-34, 2010.

[5] IEC 61508, "Functional safety of electrical/electronic/programmable electronic safety related systems", 1998.

[6] ISO 26262, "Road vehicles -- Functional safety", 2011.

[7] DO-178B, "Software Considerations in Airborne Systems and Equipment Certification", 1992.

[8] EN 50129, "Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signalling", 2003.

[9] IEC 61513, "Nuclear power plants - Instrumentation and control important to safety - General requirements for systems", 2011.

[10] ISO 60601-1, "Medical electrical equipment- Part 1: General requirements for basic safety and essential performance", 2012.

[11] So Young Moon, Sang Eun Lee, R. Youngchul Kim, "Internal Code Visualization for Analyzing Code Complexity", The 5th ICCT 2015., vol. 5, no. 1, pp. 268-269, June 2015.