

A Development of SDLC for MMIS of SMART Research Reactor

Yong Suk Suh,^a Jae Hong Park,^a In Soo Koo,^a Jong Myung Kim,^b Dong Cheol Park,^b Hyeon Soo Kim ^c
a Dept. of I&C and HFE, KAERI, 150 Dukjin-dong, Yuseong-gu, Daejeon, Korea, 305-353, yssuh@kaeri.re.kr
b Control Tech. Research Inst., SEC Ltd., 974-1 Goyeon-ri Woongchon-myon, Ulju-gun, Ulsan, Korea, 689-871
c Dept. of Computer Science and Eng., Chungnam Nat'l Univ., 220 Gung-dong, Yuseong-gu, Daejeon, Korea, 305-764

1. Introduction

Software development concept for Man Machine Interface System (MMIS) of SMART has been researched in KAERI since 2000[1]. As a result of it, we developed Software Development Life Cycle (SDLC) based on IEEE Std 1074(1997) and submitted it to Korea Institute of Nuclear Safety (KINS) as a part of Pre-SAR. We verify that the SDLC meets IEEE Std 1074(1997) by making mapping table between them in this paper.

2. Methods and Results

In this section the SDLC, IEEE Std 1074(1997), and mapping table are introduced and the SDLC's compliance with the IEEE Std 1074(1997) is also described.

2.1 SDLC for SMART MMIS

As shown in Figure 1, the SDLC has 8 phases such as Initiation, Plan, Requirement, Design, Implementation, Integration and Validation, Installation, and Operation and Maintenance phase. The SDLC is sequentially performed based on the waterfall model. This is achievable because the MMIS system design is done before starting software development. In the SDLC, the Integration and Validation (I&V) phase consisting of software module I&V, system I&V, and MMIS I&V is incrementally performed. This is the reason why we do not separate integration phase from validation phase.

2.2 IEEE Std 1074(1997)

Nuclear regulatory guidelines such as US/RG-1.173, KINS/RR-106, and KINS-G-001 strongly recommended that SDLC be established on the basis of IEEE Std 1074(1997) with adding software safety analysis activity. The 1074 describes how to create a SDLC by presenting 65 Software Life Cycle Process (SLCP) activities as shown in Table 1. The 1074 explains these each activities in the form of input, description, output. The 1074 mentions exclusion of any mandatory activities of the 65 without any rationale of the exclusion will preclude compliance with the 1074. In Table 1, all activities except marked with "(Opt)" are mandatory.

2.3 Mapping of the 1074 onto the SDLC

We develop the SDLC based on the 1074 and are sure the SDLC meets the nuclear safety requirements. We adopt to make a mapping table to verify our SDLC complies with the 1074 as shown in Table 1. This method was already presented in other project [2]. In our project, we map the 65 activities onto the 8 phases. We exclude A.2.2.2, A.4.4.1, A.4.4.2, and A.4.4.3 activities of the 1074 because of followings: (1) MMIS system design is done previously, not during the SDLC, (2) MMIS software is not retired during the life of SMART. These factors are reflected in making the Table 1. While making the table, we found some of activities in 1074 are missed in the SDLC such as metrics and project risks. We incorporated the missing activities into the SDLC later. The KINS notes the 1074 does not have activities for software safety analysis. However, software safety analysis report is produced each phases of the SDLC. So, the SDLC has more activities than the 1074. KINS will investigate and audit the implementation of the SDLC in more detail. At that time, we will submit all productive materials to KINS for review.

3. Conclusion

We introduced the SDLC for SMART MMIS and verified that the SDLC complied with IEEE Std 1074(1997) by making the mapping table between them. The SDLC can conclusively guarantee high quality and safety software for SMART MMIS. We submitted the SDLC included in Pre-SAR to KINS for review in terms of safety. Currently, SMART MMIS targeted to operate in 2010 is being developed in "SMART MMIS Joint Research and Development Center" established by KAERI and SEC in 2004.

REFERENCES

- [1] Yong Suk Suh, et al., "Software Development Concept for SMART MMIS Design", Proceedings of the Korean Nuclear Spring Meeting, Korea, May 2000.
- [2] Doo Hwan Kim, et al., "Development Methodology for the Software Life Cycle Process of the Safety Software", Proceedings of the Korean Nuclear Spring Meeting, Korea, May 2002.

1	2	3	4	5	6	7	8
Initiation	Plan	Req't	Design	Implement'n	Integ'n & Valid'n	Installation	Oper'n & Maint'e
Contract Project	Dev't Plan	SRS (SRA)	SDS (S/W Structure)	Source Code			
	Manag't Plan	Management Report (each phases)					
	Int'n&Val'n Plan				Integration Report		
	Instal'n Plan				Installation Manual	Instal'n Rep't	
	Train'g Plan				Training Manual	Train'g Rep't	
	Oper'n Plan				Operation Manual		
	Maint'e Plan				Maintenance Manual		Maint'e Rep't
	CM Plan	Configuration Management Report (each phases)					
	SA Plan	Safety Analysis Report (each phases)					
	V&V Plan	Verification and Validation Report (each phases)					
		Unit Test Plan	Unit Test Design	Unit Test Proc'd'e	Unit Test Report		
		Module Integ'n Test Plan	Module Integ'n Test Design	Module Integ'n Test Proc'd'e	Module Integ'n Test Report		
		System Test Plan	System Test Design	System Test Procedure	System Test Report		
		MMIS Integration Test Plan	MMIS Integration Test Design	MMIS Integration Test Procedure	MMIS Integration Test Report		
		Pre-Operation Test Plan	Pre-Operation Test Design	Pre-Operation Test Procedure	Pre-Operation Test Report		
QA Manual	QA Plan	Quality Assurance Report (each phases)					

Figure 1. SDLC for SMART MMIS SRS: S/W Requirement Specification, SRA: S/W Requirement Analysis, SDS: S/W Design Specification

Table 1. Mapping of the 1074 activities onto the SDLC phases

NO	Activities in IEEE Std 1074(1997)	Phases SDLC	in
1	A.1.1.1 Create SLCP	1	
2	A.1.1.2 Perform Estimation	1,2	
3	A.1.1.3 Allocate Project Resources	1,2	
4	A.1.1.4 Define Metrics	1,2	
5	A.1.2.1 Plan Evaluations	1,2	
6	A.1.2.2 Plan Configuration Management	2	
7	A.1.2.3 Plan System Transition (Opt)	2	
8	A.1.2.4 Plan Installation	2	
9	A.1.2.5 Plan Documentation	2	
10	A.1.2.6 Plan Training	2	
11	A.1.2.7 Plan Project Management	2	
12	A.1.2.8 Plan Integration	2	
13	A.1.3.1 Manage Risks	3,4,5,6,7,8	
14	A.1.3.2 Manage the Project	3,4,5,6,7,8	
15	A.1.3.3 Identify SLCP Improvement Needs	3,4,5,6,7,8	
16	A.1.3.4 Retain Records	3,4,5,6,7,8	
17	A.1.3.5 Collect and Analyze Metric Data	3,4,5,6,7,8	
18	A.2.1.1 Identify Ideas and Needs	1,2	
19	A.2.1.2 Formulate Potential Approaches	1,2	
20	A.2.1.3 Conduct Feasibility Studies	1,2	
21	A.2.1.4 Refine and Finalize the Idea or Need	1,2	
22	A.2.2.1 Analyze Functions	3	
23	A.2.2.2 Develop System Architecture	N/A	
24	A.2.2.3 Decompose System Requirements	3	
25	A.2.3.1 Identify Imported S/W Requirements	1,2,3	
26	A.2.3.2 Evaluate S/W Import Sources (Opt)	1,2,3	
27	A.2.3.3 Define S/W Import Method (Opt)	1,2,3	
28	A.2.3.4 Import S/W (Opt)	3,4,5,6,7	
29	A.3.1.1 Define and Develop S/W Requirements	3	
30	A.3.1.2 Define Interface Requirements	3	
31	A.3.1.3 Prioritize and Integrate S/W Requirements	3	
32	A.3.2.1 Perform Architectural Design	4	
33	A.3.2.2 Design Data Base (Opt)	4	
34	A.3.2.3 Design Interfaces	4	
35	A.3.2.4 Perform Detailed Design	4	
36	A.3.3.1 Create Executable Code	5	
37	A.3.3.2 Create Operating Documentation	4,5,6,7	
38	A.3.3.3 Perform Integration	6	
39	A.4.1.1 Distribute S/W	6,7	
40	A.4.1.2 Install S/W	6,7	
41	A.4.1.3 Accept S/W in Operational Environment	7	
42	A.4.2.1 Operate the System	8	
43	A.4.2.2 Provide Technical Assistance and Consulting	7,8	
44	A.4.2.3 Maintain Support Request Log	7,8	
45	A.4.3.1 Identify S/W Improvement Needs	3,4,5,6,7,8	
46	A.4.3.2 Implement Problem Reporting Method	3,4,5,6,7,8	
47	A.4.3.3 Reapply SLC	2,3,4,5	
48	A.4.4.1 Notify User	N/A	
49	A.4.4.2 Conduct Parallel Operations (Opt)	N/A	
50	A.4.4.3 Retire System	N/A	
51	A.5.1.1 Conduct Reviews	2,3,4,5,6,7,8	
52	A.5.1.2 Create Traceability Matrix	3,4,5,6,7,8	
53	A.5.1.3 Conduct Audits	2,3,4,5,6,7,8	
54	A.5.1.4 Develop Test Procedures	4,5,6	
55	A.5.1.5 Create Test Data	3,4,5,6	
56	A.5.1.6 Execute Tests	3,4,5,6,7	
57	A.5.1.7 Report Evaluation Results	2,3,4,5,6,7,8	
58	A.5.2.1 Develop Configuration Identification	2,3,4,5,6,7,8	
59	A.5.2.2 Perform Configuration Control	2,3,4,5,6,7,8	
60	A.5.2.3 Perform Status Accounting	1,2,3,4,5,6,7,8	
61	A.5.3.1 Implement Documentation	1,2,3,4,5,6,7,8	
62	A.5.3.2 Produce and Distribute Documentation	1,2,3,4,5,6,7,8	
63	A.5.4.1 Develop Training Materials	4,5,6,7	
64	A.5.4.2 Validate the Training Program	4,5,6,7	
65	A.5.4.3 Implement the Training Program	7	

(N/A: Not Applicable activity)