특집 01

Dependable Software 관련 국제표준 동향 및 통합 프레임워크 제언

최옥주 · 백종문 (한국과학기술원)

목 차 »

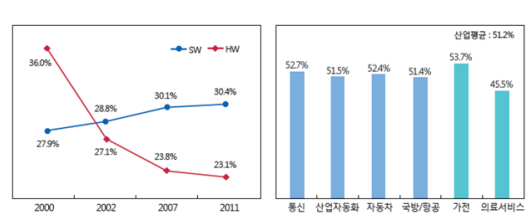
1. 서 론

최근 미국 소프트웨어 컨설팅 업체인 BARR GROUP은 인명사고로 연결된 도요타의 자동차 급발진 사고가 엔진 스로틀 컨트롤 시스템(ETCS)에 내장된 전자제어장치(ECU)의 소프트웨어 결함에 의한 것이라는 조사 보고서를 발행하였다[1]. 이로 인해 도요타는 천문학적인 벌금을 지불해야만 했다. 이와 같이 자동차 분야뿐 아니라 항공/국방, 철도, 의료 분야와 같은 안전 필수 시스템(Safety-Critical System) 또는 임무 필수 시스템(Mission-Critical System)에 탑재되는 소프트웨어의 결함은 막대한 인명 사고나 경제적 손실을 일으킬 수 있다.

최근 몇 년 사이 전 국민의 60% 이상이 사용하고 있는 스마트 기기 뿐 아니라 국가 기반을 책임지고 있는 자동차, 항공/국방, 철도, 의료 분야에 탑재되는 시스템까지 개발 원가가 전체 시스템의 50% 이상을 소프트웨어가 차지하고 있고, 그 크기나 복잡도 또한 지속적으로 크게 증가하고 있는 상태이다[2].

이와 같이 Critical System의 소프트웨어에 대한 의존성이 높아지면서 소프트웨어 결함으로 인한 시스템 고장은 사회 경제적인 손실 뿐 아니라 시스템의 신뢰성이나 안전성, 가용성과 같은 Dependability Attributes에 영향을 미칠 수 있다[3]. 따라서 소프트웨어 신뢰성이나 안전성 등의 품질 요소 향상을 통해 시스템 전체의 품질을 개선하고자 하는 노력이 요구된다.

본고에서는 소프트웨어 결함으로 인한 손실을 피하고 보다 고품질의 시스템 개발을 위한



(그림 1) 소프트웨어 비중 확대

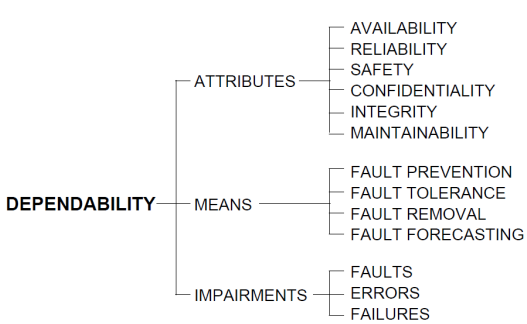
Dependable Software 관련 국제 표준을 살펴보고 Dependable Software 개발을 위한 통합 프레임워크를 제언한다.

2. Dependability Attributes

Dependability는 사용자에게 전달되는 컴퓨터 시스템의 기능 또는 서비스에 대한 신뢰의 정도를 의미한다. Dependable System은 사용자가 예상하는 대로 시스템이 작동되고 정상적으로 사용할 경우 실패 하지 않는다는 것을 의미하는 것으로 사용자가 믿을 수 있는 시스템이라 할 수 있다. Dependability는 다음과 같이 네 가지 중요한 주요 속성을 갖고 있다[3]: (1)Availability(가용성), (2)Reliability(신뢰성), (3) Safety(안전성), (4) Security(보안성)

⊙ Availability(가용성) 와 Reliability(신뢰성)

신뢰성은 주어진 환경에서 특정 목적을 위해 특정 시간동안 시스템이 실패가 발생하지 않을 확률을 의미한다. 가용성은 특정 시점에 요청된 서비스를 수행 할 수 있는 확률을 의미한다. 신뢰성과 가용성은 수학적 확률로 표현되는 밀접한 관계가 있다. 그러나 신뢰할 수 있는 시스템은 항상 사용 가능하나 사용 가능한 시스템은 항상 신뢰할 수 있다고 볼 수 없다.



(그림 2) Dependability Tree

⊙ Safety(안전성)

안전성은 시스템이 정상적이든 비정상적이든 시스템에 큰 손상이 없으며 인명피해를 주지 않는 범위 내에서 작동되는 시스템의 특성을 의미한다. 시스템 통합에 의해 점점 더 복잡하고 많아진 센서나 작동기들이 소프트웨어 기반으로 제어되고 있기 때문에 소프트웨어 안전성이 점점 더 중요해지고 있다. 특히 항공기, 국방 무기체계, 의료기기 등에 탑재되는 안전 필수 시스템이나 임무 필수 시스템에서의 소프트웨어의 안전성은 필수적이다. 신뢰성과 안전성은 관계가 있지만 다소 다르다고 볼 수 있다. 일반적으로 시스템 안전성을 위해 신뢰성과 가용성은 필요하지만 충분조건이 되지는 않는다.

⊙ Security(보안성)

보안성은 의도적이든 사고이든 외부의 공격으로부터 자신을 보호하는 능력을 의미한다. 보안성은 시스템이 네트워크화 되어 인터넷을 통한 외부접근이 가능해짐에 따라 점점 더 중요해지고 있다. 특히 국방 시스템에서는 고도의 보안성이 요구된다. 따라서 보안성이 확보되지 않은 시스템은 외부 공격으로 인해 정상적인 서비스가 불가능하거나 프로그램과 데이터의 손실로 공적 효력을 갖지 못할 뿐 아니라 국방 시스템의 경우 군무력화 등 여러 가지 피해를 볼 수 있다.

Dependable System은 시스템 특성이나 산업분야에 따라 각기 다른 속성이 적용되므로 모든 시스템에 같은 Dependability Attributes가 적용되는 것은 아니다. 따라서 Critical System 설계자는 개발하고자 하는 시스템의 특성 및 목적, 범위를

분석하여 다음과 같은 접근 방법을 활용하여 시스템에서 필요로 하는 Dependability Attributes을 반영한다.

- Fault Prevention(결함예방): 시스템 결함 삽입되거나 발생되기 전에 실수나 그 가능성을 최소화하거나 예방할 수 있는 개발 기술(Development Techniques)을 사용한다.
- Fault Tolerance (결함허용): 시스템 결함이 시스템 오류나 실패로 이어지지 않도록 실행시간 기술(Run-time Techniques)을 사용한다.
- Fault Removal (결함제거): 결함을 발견되면 시스템을 사용(운용)하기 전에 결함을 제거할 수 있는 확률을 증가시키는 검증기술(Verification and Validation)을 사용한다.
- Fault Forecasting(결함예측): 현재 발생된 결함 분석하여 미래에 발생할 수 있는 결함을 예측할 수 있는 기술(Forecasting Techniques)을 사용한다.

3. Dependable Software 개발을 위한 국제표준

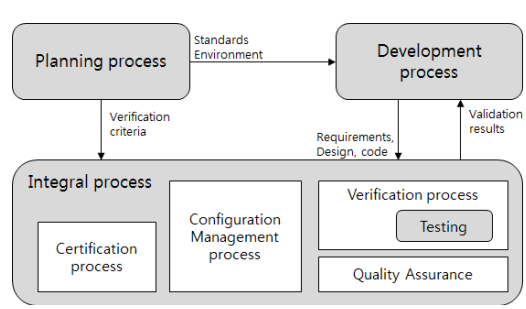
Dependable Software을 개발하기 위해서는 소프트웨어 개발 과정에서 신뢰성, 안전성 등의 Dependability Attributes가 확보되어야 한다. 본 장에서 Dependability를 확보할 수 있는 표준들을 살펴본다.

3.1 DO-178B/C

DO-178B/C는 항공기 시스템과 장비 인증에 관한 소프트웨어 고려사항(Software Considerations in Airborne Systems and Equipment Certification)으로 항공 소프트웨어의 개발 전 과정에서 소프트웨어 신뢰성 및 안전성을 검증하기 위한 가이드라인이다[5,6]. 이는 미국 항공기 소프트웨어의 안전성 인증을 위한 표준이었으나 최근에는 국방 무기체계나 의료기기 등 일반 산업 분야까지 확대 적용되고 있다.

DO-178B/C는 소프트웨어 개발 프로세스를 규정하고 각 단계의 산출물과 검증요건을 명시하고 있다. 특히 항공기의 특수한 안전성을 보장하기 위해 다른 소프트웨어 개발 표준에 비해 보다 더 엄격한 기준을 제시하고 있으며 소프트웨어 프로세스 활동에 중점을 두고 있다. DO-178B/C는 소프트웨어 결함 시 발생할 수 있는 위험성에 따라 레벨 A에서 레벨 E까지 모두 다섯 단계로 차등 검증을 하고 있고 각 소프트웨어 레벨에 따라 달성해야 하는 산출물 목표 수를 다르게 정의하고 있다. 이러한 차등화를 통해 위험성에 따른 합리적인 프로세스를 적용할 수 있다.

1980년대에 제정된 DO-178B는 부정확함이나 불일치 등의 문제점이 존재하고 새로운 소프트웨어 개발 방법과 검증 방법들이 반영 되어있지 않다. 이러한 문제점을 수정하고 DO-178B에서 다뤄지지 않았던 소프트웨어 도구의 인증과 고려사항, 모델 기반 설계, 객체지향 설계, 정형화된 기법 등 새로운 소프트웨어 방법론과 기술을 적용하여 2011년 11월 DO-178C를 발표하였다.



(그림 3) DO-178B/C Life-cycle Process

〈표 1〉 DO-178B/C Level

Level	Effect of anomalous behavior	DO-178B	DO-178C	Change
A	Catastrophic failure condition	66	71	+5
B	Hazardous/severe failure condition	65	69	+4
C	Major failure condition	57	62	+5
D	Minor failure condition	28	26	-2
E	No effect	0	0	No change

3.2 IEC 61508

IEC 61508 “Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems”는 전기/전자/제어 장치의 생명주기 동안 수행되는 기능 안전에 대한 활동을 정의하고 있으며 위험 정도에 따라 안전 관련 시스템의 기능이 갖추어야 할 요구사항을 제시한다[7]. 기능 안전성이란 시스템이나 장비가 정상적으로 동작하여 물리적 위험이나 인명과 재산 손실에 대한 위험으로부터 자유로운 것을 의미한다.

IEC 61508에서는 위험이 발생할 수 있은 확률과 재난의 결과로 안전에 영향을 주는 심각성 정도에 따라 안전 무결성 등급(Safety Integrity Level: SIL)을 정의하고 있다. SIL 등급에 따라 실패율 범위를 제시하고 있으며 이에 대한 인증을 통해 소프트웨어 결함과 실패를 통제하고 있다. SIL 등급에 따른 실패율 범위는 신뢰성과 안전성을 달성하기 위한 기준으로 활용될 수 있다.

IEC 61508의 Part 3에서는 Software safety lifecycle을 통해 소프트웨어 설계 및 개발 과정에서 안전성 보장을 위해 수행해야 하는 프로세스와 활동들의 요구사항을 제시하고 있다. SIL 등급 별로 수행해야 하는 코딩규칙이나 정적 또는 동적 분석 등의 항목을 테스팅을 통해 수행함으로서 소프트웨어 신뢰성 향상을 꾀할 수 있다. IEC 61508을 기반으로 각 산업별l 특성을 고려한 기능 안전 표준이 제정되고 있다: 자동차(ISO 26262), 철도(EN 51208), 원자력(IEC 61513), 의료(IEC 60601)

〈표 2〉 Safety Integrity Level(SIL) and Failure Rate

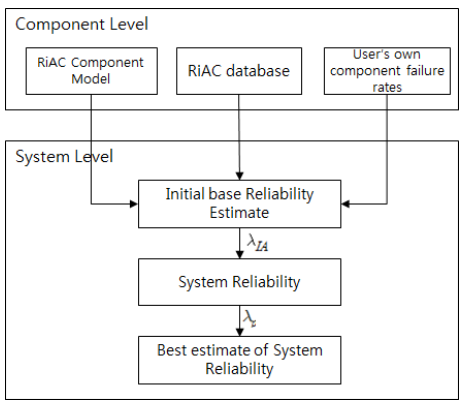
SIL Level	Average probability of failure on demand (unreliability)	Probability of dangerous failure per hour(unsafety)
4	$\geq 10^{-5}$ to $< 10^{-4}$	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-4}$ to $< 10^{-3}$	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-3}$ to $< 10^{-2}$	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-2}$ to $< 10^{-1}$	$\geq 10^{-6}$ to $< 10^{-5}$

3.3 MIL-HDBK-217Plus

MIL-HDBK-217Plus “Reliability Prediction Models”는 국방 분야 시스템 신뢰성을 예측하기 위한 방법론 및 관련 소프트웨어 도구를 제공한다. MIL-HDBK-217Plus는 시스템에 대한 정보가 부족한 설계 초기단계부터 신뢰성 예측 활동을 수행하므로 설계 이후에 발생하는 결함을 예방하고 결함 제거나 결함 관리에 대한 노력이 감소시킴으로서 보다 신뢰성 있는 시스템을 확보할 수 있다.

MIL-HDBK-217Plus는 컴포넌트 신뢰성 예측 모델과 시스템 레벨 모델로 이루어져 있다. 시스템 실패율(System Failure Rate)을 추정하기 위해서는 먼저 각 컴포넌트에 대한 실패율을 추정하고 각각의 컴포넌트에 대한 실패율을 반영하여 시스템 실패율을 구한다.

시스템 레벨에서 설계, 부품, 제조, 시스템 관



(그림 4) 시스템 실패율 계산

리, 마모 등 비 부품 요인들을 포함하여 초기 고장이나 신뢰성 성장에 영향을 미치는 팩터를 고려하여 시스템 실패율을 산출한다.

신뢰성에 영향을 미치는 팩터들은 MIL-HDBK-217Plus에서 제공하는 질문서를 통해 값을 산출한다. 이 질문서는 팩터 당 적게는 수십 개에서 많게는 백여 개의 질문이 있는 경우도 있으며 도메인 지식이 있는 담당자가 지침에 따라 질문서를 작성한다.

질문서가 작성이 되면 실패 요인 등급(R)과 실패요인에 대한 팩터($\prod$)를 계산하여 (식 1)을 활용하여 시스템 실패율을 계산한다.

$$R = \frac{\sum_{j=1}^{n} G_j W_j}{\sum_{j=1}^{n} W_j}$$

- R = 실패요인에 대한 등급
- n = 해당 실패 요인의 질문수
- G_j = 실패 요인의 j번째 질문에 대한 등급
- W_j = 실패 요인의 j번째 질문에 대한 가중치

(식2),

$$\prod = \beta \times (-\ln R)^{\frac{1}{\alpha}},$$

- $\prod$: 특정 실패요인의 팩터
- R : 특정 실패요인의 등급(식1)
- α : 해당 실패요인 형상모수
- β : 해당 실패요인의 특성 비율

(식3)

3.4 ISO/IEC 25000 Series

SQuaRE(Systems and Software Quality Requirements and Evaluation)라 불리우는 ISO/IEC 25000 시리즈는 시스템과 소프트웨어 품질 요구사항 설정 및 측정, 평가에 관한 국제 표준이다[9]. ISO/IEC 25000 시리즈는 소프트웨어 품질 속정을 정의한 ISO/IEC 9126과 소프트웨어 제품의 품질 평가를 제시한 ISO/IEC 14598을 통합하였다. 기존의 두 표준은 서로 상호보완적이며 함께 고려해야 할 사항이 많았으나 서로 다른 표준으로 제정되어 있어서 서로 일관되지 못한 결과가 나타나는 문제가 있었다.

ISO/IEC 25000 시리즈는 이런 두 표준을 통합하여 시스템과 소프트웨어 제품의 품질 요구사항 명세하고 시스템과 소프트웨어의 품질 측정 및

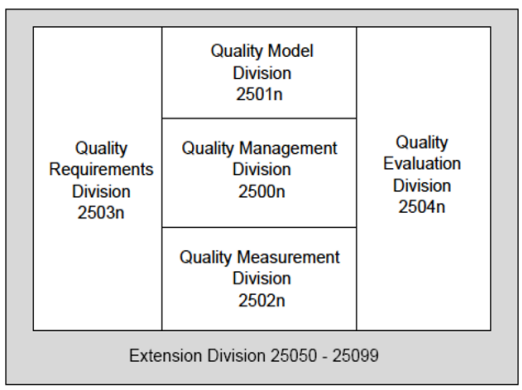
〈표 3〉 예제: '신뢰성 성장' 팩터와 관련 된 질문

질문	가중치	점수
1. 운영될 시스템을 위해 FRACAS와 같은 효과적인 고장 보고 및 수정 시스템이 존재하는가?	6	6
2. 고장의 주 원인이 식별될 확률은?	4	0
3. 고장이 재발 할지에 대한 판단하기 위한 분석이 수행 되었는가?	2	0
4. 잠재적으로 수정 가능성 있는 설계, 제조, 시스템 관리 부분이 식별 되었는가?	4	4

$$\lambda_p = \lambda_{IA} \times f(\prod) + \lambda_{SW}$$

- λ_p = 시스템 예측 실패율
- λ_{IA} = 실패율 초기 평가값
- $f(\prod)$ = 팩터(설계, 제조, 품질, 마모 등 비부품관련 요소들)들의 함수
- λ_{SW} = 소프트웨어 예측 실패율

(식 1)



(그림 5) ISO/IEC 25000 Series 구성도

평가를 위한 기준들을 제시하고 있다.

⦿ ISO/IEC 2500n, Quality Management Division

2500n에서는 표준 전반적으로 사용되는 공통의 모델과 용어를 정의를 하고 있다. 또한 전체 표준 문서들을 참조하기 위한 참조 모델과 제품 품질 요구사항 명세, 측정, 평가를 관리하기 위한 요구사항과 지침들도 포함하고 있다.

⦿ ISO/IEC 2501n, Quality Model Division

시스템/소프트웨어 제품 품질 모델, 사용 중 품질 모델, 데이터 품질 모델로 분류하여 SQuaRE의 품질 모델을 상세하게 설명하고 모델 적용을 위한 지침을 제공한다.

⦿ ISO/IEC 2502n, Quality Measurement Division

시스템과 소프트웨어 제품 품질 측정을 위한 참조 모델과 품질 측정을 위한 수학적 정의 와 품질 측정을 위한 지침이 포함되어 있다. 측정을 위한 기초를 구성하는 Quality Measure Elements (QME) 가 정의되고 설명되어있다.

⦿ ISO/IEC 2503n, Quality Requirements Division

2503n에서는 품질 요구사항 명세에 대한 내용이다. 품질 요구사항 도출 프로세스에 사용 가능하며 도출된 품질 요구사항은 평가 프로세스에서 평가 기준으로 사용 가능하다.

⦿ ISO/IEC 2504n, Quality Evaluation Division

2504n에서는 제품의 품질 평가를 위한 요구사항이나 추천사항, 지침들에 대하여 평가자, 고객, 개발자 각각의 측면에서 설명하고 있다.

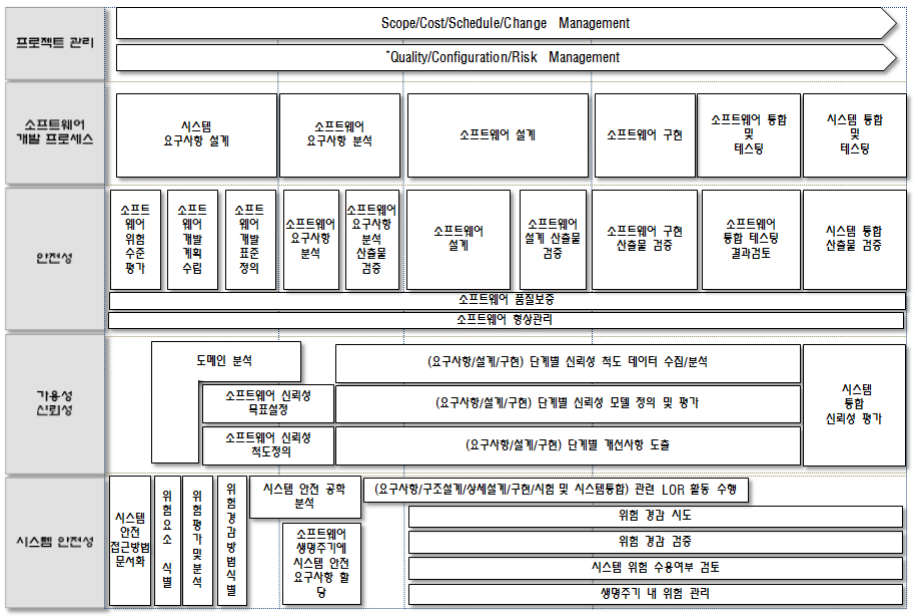
⦿ 25050~25099, Extension Division

SQuaRE 의 보완이나 기술 문서, 특정 분야에 대한 확장 등의 내용을 담고 있다.

4. Dependable Software를 위한 통합 프레임워크

소프트웨어 시스템의 Dependability를 확보하기 위해서는 소프트웨어 제품 자체 뿐 아니라 개발 프로세스 초기부터 단계별로 Dependability Attribute를 평가하고 측정을 통해 결함을 최소화하기 위한 활동이 수행되어야 한다. 따라서 고품질의 Dependable Software를 개발하기 위해서는 소프트웨어 개발 프로세스를 기반으로 신뢰성, 안전성, 보안성 확보를 위한 보증 활동을 정의한 통합 프레임워크가 요구된다. 그러나 개발 목적이나 범위에 따라 Dependability Attribute는 적용이 달라지므로 본 연구에서는 소프트웨어 개발 단계별 신뢰성과 안전성을 고려한 통합 프레임워크를 다음과 같이 제언한다.

Dependable Software를 개발하기 위해서는 프로젝트 관리 측면에서 PMO(Project Management



(그림 6) Dependable Software을 위한 통합 프레임워크

Office)를 두고 비용관리, 일정관리, 범위관리, 변경관리, 형상관리 등을 개발 전 과정에서 통제하고 프로세스를 관리 할 필요가 있다. 이는 프로젝트를 진행하는 과정에서 기술적 문제 외에도 비용이나 일정, 요구사항이나 설계 변경 등의 요인에도 불구하고 신뢰성이나 안전성 목표를 달성해야 Dependability가 확보되기 때문이다. 프로젝트 관점에서 품질관리, 형상관리 및 위험관리 등은 신뢰성과 안전성 프로세스에도 포함되어 있으므로 산업 분야별로 조직 목적이나 개발 환경에 따라 조정하여 사용할 필요가 있다.

안전성 확보를 위해 개발 초기에 시스템이나 소프트웨어 위험성을 분석하고 위험 수준을 정의한다. 이러한 활동은 결함/실패/심각도를 정의해야 하는 신뢰성 보증 활동과 중복되므로 그 결과를 신뢰성 파트에서 반영할 수 있다. 또한 소프트웨어 요구사항 분석단계에서 신뢰성이나 안전성 요구사항을 할당하는 활동은 하나의 업무 활동으로 통합할 수 있다. 이렇듯 Dependability Attributes간의 활동이 중복되거나 순차적으로 수행해야 하는 경우가 있으므로 Dependability 보증 활동을 개발 단계별로 상세화하여 통합 프로세스를 정의하고 각 활동의 산출물과 책임부서를 명확하게 정의하는 것이 중요하다.

5. 결 론

VDC분석에 의하면 2000년대 초 국방 항공 시스템의 소프트웨어 비중이 39.7%에서 2000년 중반 이후에는 약 50%이상 증가하였고, 소프트웨어 크기는 1960년대 10%에서 2000년 이후 약 80%를 차지하고 있다. 이와 같이 소프트웨어가 차지하는 비중이나 크기가 증가하면서 시스템 전체의 신뢰도와 품질을 좌우하고 있다. 따라서 Dependable System을 개발하기 위해서는 소프트웨어의 Dependability 확보가 우선되어야 한다.

본고에서는 DO-178B/C, IEC61508과 같은 안전성 관련 국제표준과 MIL-HDBK-217Plus, ISO/

IEC 25000과 같은 신뢰성 국제표준을 살펴보고 Dependable Software개발을 위해 개발 단계별로 프로젝트 관리 측면, 안전성, 신뢰성 보증활동을 정의한 통합 프레임워크를 제언하였다. 향후에는 통합 프레임워크에서 정의한 각 절차를 상세화하고 절차간의 중복이나 유사한 활동은 통합하고 절차간의 우선순위를 정의할 예정이다. 또한 각 절차의 산출물 정의와 역할과 책임을 지정하여 Dependable Software개발을 위한 지침을 마련하고자 한다.

참 고 문 헌

[1] http://english.etnews.com/electronics/2933971_1303.html

[2] 박일준, 창조경제 실현을 위한 소프트웨어 혁신 전략, KIET 산업경제, 2013.12

[3] Jean-Claude Laprie, Dependability of Computer Systems: Concepts, Limits Challenges, the 25th IEEE International Symposium on Fault-Tolerant Computing, 1995

[4] Algirdas Avizienis, Jean-Claude Laprie, et al., Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 1, NO. 1, JANUARY-MARCH 2004

[5] DO-178B, http://en.wikipedia.org/wiki/DO-178B

[6] DO-178C, http://en.wikipedia.org/wiki/DO-178C

[7] IEC 61508, Functional safety and IEC 61508 A basic guide, 2002

[8] MIL-HDBK-217Plus, "Handbook of 217PlusTM Reliability Prediction Models", RiAC, 2006.

[9] ISO/IEC 25000, Systems and software engineering – System and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE, 2013

저 자 약 력

최 옥 주

이메일 : okjoo.choi@kaist.ac.kr

- 1987년 숙명여자대학교 전산학과(학사)
- 1990년 숙명여자대학교 전산학과(석사)
- 2008년 숙명여자대학교 컴퓨터과학과(박사)
- 1990년~1996년 LG전자 생산기술연구원 / 주임연구원
- 1996년~2009년 한국 오라클 / Consulting Technical Manager
- 2009년~현재 한국과학기술원 전산학과 / 연구조교수
- 관심분야 : 소프트웨어 신뢰성/안전성, 소프트웨어 품질 보증, 소프트웨어 프로세스, 소프트웨어 프로젝트 관리

백 종 문

이메일 : jbaik@kaist.ac.kr

- 1993년 조선대학교 컴퓨터과학 및 통계학과(학사)
- 1996년 미국 University of Southern California Computer Science(석사)
- 2000년 미국 University of Southern California Computer Science(박사)
- 2001년~2005년 미국 모토로라 SSERL(Software and System Engineering Research Lab.) / 수석연구원
- 2005년~2009년 2월 한국 정보통신대학교 공학부 / 부교수
- 2009년 3월~현재 한국과학기술원 전산학과 부교수.
- 관심분야 : 소프트웨어 신뢰성, 소프트웨어 매트릭스, 소프트웨어 비용 추정, 소프트웨어 동적모델링, 소프트웨어 프로세스 개선, 소프트웨어 품질 보증, 소프트웨어 식스 시그마