

CMMI기반의 테스트 프로세스 개선 프레임워크와 단계적 개선방안

이화여자대학교 서주영 · 최병주*

1. 서 론

최근 소프트웨어 업계는 제품의 신뢰성 및 생산성과 관련하여 품질에 대한 관심이 높아져가고 있다. 품질에 관한 관심은 ISO/IEC 9126[1], ISO/IEC 12119[2], ISO/IEC 14598[3]과 같은 표준을 중심으로 하는 제품 품질 평가 활동으로부터 시작되어, 현재 제품 품질 개선의 근원인 제품 개발 프로세스를 본질적으로 파악하려는 프로세스 품질 활동으로 이동되고 있다.

소프트웨어 프로세스 품질은 SPICE[4], CMMI[5]를 중심으로 평가 및 개선이 정의되고 있으며, 이들 개념은 지금 현재도 진화 중에 있다. 국내외 소프트웨어 관련 산업체에서도 SPICE와 CMMI에 대한 인증 획득을 위해 많은 노력을 기울이고 있으며, 그 결과 프로세스 개선 활동을 수행 중인 대부분의 기업에서 현재 조직 차원의 표준화된 제품 개발 프로세스가 잘 정착되어 가고 있다. 그러나, 소프트웨어 프로세스 품질인증이 비용 감소, 일정 준수, 재사용을 증가와 같은 경제적 효과로 즉시 가지화되지 않기 때문에 인증을 획득한 기업은 보다 적극적인 프로세스 개선 활동을 추구해야 할 필요가 있다.

본 논문에서는 프로세스 품질 확보와 더불어 적극적인 프로덕트 품질 향상을 위해 품질과 직결되는 결합 제거 활동인 테스트 프로세스 개선을 위한 프레임워크와 개선 로드맵을 제안한다. 첫 개선 단계로써 최근 급성장하고 있는 임베디드 소프트웨어 분야의 테스트 프로세스를 제안한다.

테스트 프로세스는 제품의 행위, 성능, 견고성이 기대하는 기준에 부합하는지를 평가하기 위한 프로세스로, 검증(Verification) 활동과 확증(Validation) 활동을 핵심으로 한다[6]. CMMI는 검증 및 확증 프로세스 영역을 중심으로 테스트 프로세스에 대해 언급하고 있으나, 현실적인 테스트 프로세스 개선 활동을 위해선 부족함이 있다.

본 논문에서 제안하는 '테스트 프로세스 개선 프레임워크'는 CMMI의 검증 및 확증 프로세스 영역을 테스트 관점에서 확대하여 더욱 상세히 정제하여 개발하였으며, 실제 산업 현장에서 필요로 하는 보다 적용하기 쉽고 실질적인 테스트 프로세스 개선 방안이다.

본 논문의 구성은 다음과 같다. 2장에서는 CMMI의 검증 및 확증 프로세스 영역을 분석한다. 3장에서는 제안하는 '테스트 프로세스 개선 프레임워크'를 기술한다. 4장에서 테스트 프로세스 개선 프레임워크를 토대로 3단계 개선 로드맵을 제안하고, 첫 개선 단계로써 임베디드 소프트웨어 분야의 테스트 프로세스를 제안한다. 마지막으로 5장에서 결론을 맺는다.

2. CMMI의 테스트 프로세스

본 장에서는 제안하는 '테스트 프로세스 개선 프레임워크'의 기반이 된 CMMI의 공학(Engineering) 카테고리 내의 검증 및 확증 프로세스 영역에 대해 분석하고자 한다.

CMMI는 프로세스 평가와 개선을 위한 접근 방법으로 '지속적 표현법(Continuous Representation)'과 '단계적 표현법(Staged Representation)'의 두 가지 표현법을 제공한다. '지속적 표현법'은 조직의 비즈니스 목적에 맞는 특정 프로세스 영역에 대한 능력 수준을 평가하고 지속적으로 성장할 수 있도록 하는 접근 방법이다. '단계적 표현법'은 조직 전체의 관점에서 조직의 프로세스 성숙 수준을 평가하고 조직이 순차적으로 성장할 수 있도록 하는 접근 방법이다.

본 장에서는 CMMI의 지속적 표현법에 따라 테스트 프로세스를 언급하고 있는 검증 및 확증 프로세스 영역에 대해 능력 수준을 분석하여 테스트 프로세스 개선을 위해 필요한 기본 요소를 식별하고자 한다.

2.1 검증 프로세스 영역

검증은 선택된 산출물이 명세 된 요구사항을 만족시킨다는 것을 보증하기 위한 활동이다[4,5,6]. 다음 표

* 종신회원

표 1 검증 프로세스 영역의 능력 수준별 프랙티스

능력수준	특수 프랙티스	일반 프랙티스
5 최적화된 프로세스 제도화	없음	GP5.1 지속적인 프로세스 개선을 보증 GP5.2 문제 근본 원인 교정
4 정량화된 프로세스 제도화	없음	GP4.1 프로세스를 위한 정량적 목표 확립 GP4.2 서브 프로세스 수행 안정화
3 정의된 프로세스 제도화	SP1.3-3 검증 절차와 기준 확립	GP3.1 정의된 프로세스 확립 GP3.2 개선 정보 수집
2 관리 프로세스 제도화	SP1.2-2 검증 환경 수립 SP2.3-2 동료검토 자료 분석 SP3.2-2 검증 결과 분석 및 수정 활동 식별	GP2.1 조직 정책의 확립 GP2.2 프로세스 계획 GP2.3 자원 제공 GP2.4 책임 할당 GP2.5 사람 훈련 GP2.6 형상 관리 GP2.7 관련 스테이크홀더 식별 GP2.8 프로세스 감시와 통제 GP2.9 충실도 평가 GP2.10 높은 관리 수준으로 상태 검토
1 특수(기본) 프랙티스 달성	SP1.1-1 검증을 위해 작업 산출물 선택 SP2.1-1 동료검토 준비 SP2.2-1 동료검토 수행 SP3.1-1 검증 수행	GP1.1 기본 프랙티스 수행
0 Incomplete	없음	없음

표 2 확증 프로세스 영역의 능력 수준별 프랙티스

능력수준	특수 프랙티스	일반 프랙티스
5 최적화된 프로세스 제도화	없음	GP5.1 지속적인 프로세스 개선을 보증 GP5.2 문제 근본 원인 교정
4 정량화된 프로세스 제도화	없음	GP4.1 프로세스를 위한 정량적 목표 확립 GP4.2 서브프로세스 수행 안정화
3 정의된 프로세스 제도화	SP1.2-3 확증 절차와 기준 확립	GP3.1 정의된 프로세스 확립 GP3.2 개선 정보 수집
2 관리 프로세스 제도화	SP1.2-2 확증 환경 수립	GP2.1 조직 정책의 확립 GP2.2 프로세스 계획 GP2.3 자원 제공 GP2.4 책임 할당 GP2.5 사람 훈련 GP2.6 형상 관리 GP2.7 관련 스테이크 홀더 식별 GP2.8 프로세스 감시와 통제 GP2.9 충실도 평가 GP2.10 높은 관리 수준으로 상태 검토
1 특수(기본) 프랙티스 달성	SP1.1-1 확증을 위한 프로덕트 선택 SP2.1-1 확증 수행 SP2.2-1 확증 결과 분석	GP1.1 기본 프랙티스 수행
0 Incomplete	없음	없음

1은 CMMI의 지속적 표현법에 따른 검증 프로세스 영역의 능력 수준별 수행되어야 하는 프랙티스이다. 즉, 특정 능력 수준을 위해서는 검증을 위한 특수 프랙티스

뿐만 아니라 일반 프랙티스도 만족되어야 한다. CMMI의 일반 프랙티스는 모든 프로세스 영역에서 공통적이나 실제 적용을 위해서는 프로세스 영역마다 의미를 재해석

하여 적용하여야 한다.

2.2 확증 프로세스 영역

확증은 프로덕트나 프로덕트 컴포넌트가 계획된 환경에 놓였을 때 계획된 대로 실행됨을 증명하기 위한 활동이다[4,5,6]. 다음 표 2는 CMMI의 지속적 표현법에 따른 확증 프로세스 영역의 능력 수준별 수행되어야 하는 프랙티스이다. 확증 프로세스 영역의 경우에도 확증을 위한 특수 프랙티스 뿐만 아니라 더 높은 능력 수준을 위해서는 일반 프랙티스를 중심으로 지속적인 개선을 수행하여야 한다.

조직이 실제 개발 현장에서 CMMI를 기반으로 테스트 프로세스를 개선하고자 할 때에는 CMMI의 검증과 확증 프로세스 영역을 중심으로 묵시적으로 정의되어 있는 일반 프랙티스를 보다 명확히 정의하고, 보다 손쉽게 적용할 수 있도록 상세화할 필요가 있다.

검증 및 확증 프로세스 영역은 그림 1과 같이 CMMI의 공학 카테고리에 속하는 다른 프로세스 영역들과 밀접한 연관을 맺고 있고, 공학 카테고리도 CMMI의 다른 카테고리들과 서로 상호작용하기 때문에 단순히 검증 및 확증 프로세스 영역을 상세화 하는 것만으로 테스트 프로세스를 개선시킬 순 없다.

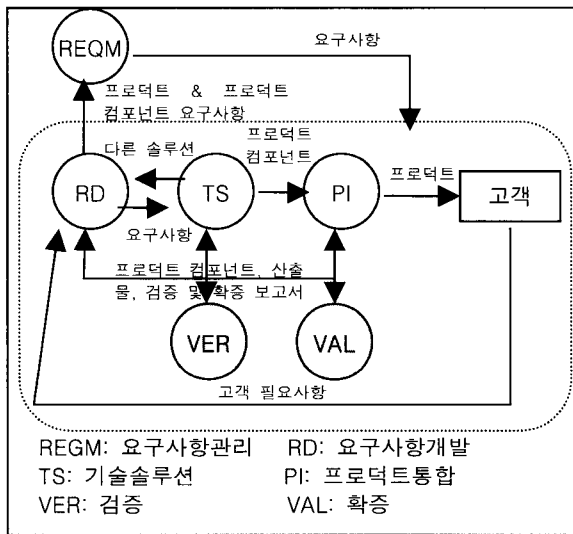


그림 1 CMMI 공학 카테고리의 프로세스 영역 연관도

즉, 산업 현장에서 쉽게 적용토록 하기 위해선 검증 및 확증 프로세스 영역을 중심으로 테스트와 관련되는 CMMI의 모든 프로세스 영역들을 분석하여 테스트 프로세스 개선 활동을 명확하고 간결하게 재정립할 필요가 있다. 이러한 목적을 위해 본 논문에서는 '테스트 프로세스 개선 프레임워크'를 제안한다.

3. 테스트 프로세스 개선 프레임워크

3.1 정의

'테스트 프로세스 개선 프레임워크는 프로세스 품질과 프로덕트 품질을 동시에 고려하기 위해 테스트 프로세스를 기반으로 간결하고, 실용적이어서 적용하기 쉽도록 하는 테스트 프로세스 개선을 위한 프레임워크다.

테스트 프로세스 개선 프레임워크는 프로젝트 성공 요소로 평가되는 숙련된 인력(People), 체계화된 프로세스(Process), 첨단기술(Technology)을 중심으로 소프트웨어의 품질보증(Quality)과 점진적 개선(Improvement)을 핵심 목적으로 한다.

3.2 구조

'테스트 프로세스 개선 프레임워크는 그림 2와 같이 테스트 프로세스 평가와 개선을 위한 핵심 모델과 능력 수준 및 조직의 테스트 프로세스 성숙 수준을 결정하기 위한 평가 기준, 체계적인 평가를 위한 평가 스크립트의 3가지 구성요소로 정의되며 본 논문에서는 테스트 프로세스 핵심 모델을 중심으로 기술할 것이다.

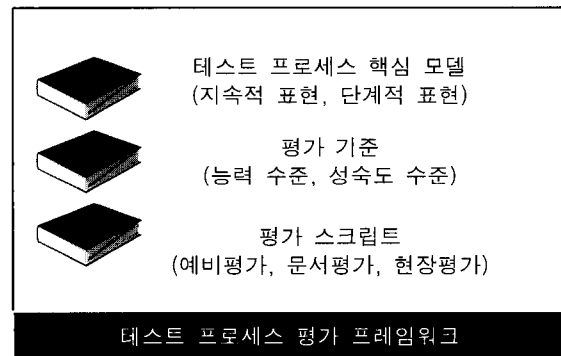


그림 2 테스트 프로세스 개선 프레임워크의 구성요소

3.3 테스트 프로세스 핵심 모델

3.3.1 정의

테스트 프로세스 핵심 모델은 CMMI의 검증 및 확증 프로세스 영역을 테스트 관점에서 상세히 분석하여 실제 산업 현장에서 필요로 하는 보다 적용하기 쉽고 실질적인 12개의 테스트 프로세스 영역으로 확대 정의한 테스트 프로세스를 위한 개선 모델이다. 테스트 프로세스 핵심 모델의 프로세스 영역은 CMMI의 검증 및 확증 프로세스 영역의 일반 프랙티스를 보다 쉽게 개발 현장에서 적용하고 평가할 수 있도록 하기 위해 '인력, 프로세스, 기술, 품질보증, 개선'의 항목으로 분석하여 정의하였다. 다음 표 3은 테스트 프로세스 핵심 모델의 프로세스 영역에 대한 설명이다.

3.3.2 표현 방법

테스트 프로세스 핵심 모델은 CMMI를 기반으로 하

표 3 테스트 프로세스 핵심 모델의 프로세스 영역

프로세스 영역	설명
테스트 인력	테스트 계획, 테스트 설계, 테스트 수행, 테스트 결과 추적 및 평가와 같은 테스트 프로세스를 담당하는 역할 및 조직에 관한 프로세스 영역 테스트 인력의 전문성을 보증하기 위해 테스트 계획, 테스트 설계, 테스트 수행, 테스트 도구, 표준에 관한 최신 기술을 훈련하여 고급 테스터로 성장할 수 있도록 하기 위한 프로세스 영역
검증	개발 문서가 테스트케이스 선정과 관련된 테스트 기반 문서로 활용될 수 있는지를 보증하기 위한 프로세스 영역
확증	프로덕트 컴포넌트나 프로덕트가 예정된 환경에 놓였을 때 올바르게 실행되는지를 보증하기 위한 프로세스 영역
감사	독립적인 제 3의 그룹에 의해 제품과 프로세스가 요구사항, 계약사항, 일정과 비용을 준수하고 있는지를 평가하기 위한 프로세스 영역
테스트 기술	테스트 프로세스 영역별 능력을 향상시키기 위한 테스트 기술 및 방법에 관한 프로세스 영역
테스트 인프라	테스트가 수행될 하드웨어, 소프트웨어 환경과 도구에 관한 프로세스 영역
측정	테스트의 생산성, 효율성, 진척성과 같이 제품 품질을 평가하기 위해 필요한 데이터를 수집, 측정, 분석하기 위한 프로세스 영역
품질 보증	프로세스 단계별로 제품 품질을 평가하기 위한 프로세스 영역
결함 예방	결함의 패턴과 오류의 근본 원인을 분석하여 유사한 결함이 다시 발생하는 것을 막고 테스트 활동의 진행 상황을 추적할 수 있도록 하기 위한 프로세스 영역
품질 제어	과제 수행 중 현재의 제품 품질 파악하고 미래의 품질을 예측하여 적절한 제어를 수행할 수 있도록 하기 위한 프로세스 영역
테스트 프로세스 최적화	테스트 프로세스의 지속적인 개선을 위한 프로세스 영역

기 때문에 CMMI와 같이 지속적 표현과 단계적 표현의 두 가지 접근 방법을 동일하게 지원한다.

가. 지속적 표현법

테스트 프로세스 핵심 모델은 12개의 테스트 프로세스 영역 중 관심이 높은 영역만을 평가하고 개선할 수 있는 로드맵을 제안한다. 테스트 프로세스 핵심 모델의 12개의 테스트 프로세스 영역은 '인력, 프로세스, 기술, 품질, 개선'의 5개의 카테고리 그룹화되어 관리되며, 각 프로세스 영역의 평가 기준은 CMMI의 능력 수준과 동일한 의미로 평가된다.

나. 단계적 표현법

테스트 프로세스 핵심 모델은 조직 전체의 관점에서 조직의 테스트 프로세스 성숙 수준을 평가하고 조직이 추구할 순차적인 개선 로드맵을 제안한다. 테스트 프로세스 핵심 모델의 단계적 표현의 평가 기준은 CMMI의 성숙도 수준과 동일한 의미를 갖는다. 그러나, 본 논문이 제안하는 테스트 프로세스 핵심 모델은 프로세스 영역이 테스트로 집중되어 있기 때문에 제안하는 모델의 단계적 표현의 성숙도 수준별 포함되는 프로세스 영역은 테스트에 특화되어 정의되었다. 또한, 성숙도 수준의 의미도 테스트에 특화되어 정의하였다.

다음은 테스트 프로세스의 성숙 단계를 소프트웨어

표 4 테스트 프로세스 핵심 모델의 지속적 표현

카테고리	정의 / 테스트 프로세스 영역
인력	테스트 관련 인력 구조와 숙련된 인력 확보를 위한 프로세스 영역 그룹 <ul style="list-style-type: none"> • 테스트 인력 • 테스트 교육
프로세스	소프트웨어 개발 프로세스와 통합된 표준화된 테스트 프로세스 정의를 위한 프로세스 영역 그룹 <ul style="list-style-type: none"> • 검증 • 확증 • 감사
기술	첨단 테스트 기술 확보를 위한 프로세스 영역 그룹 <ul style="list-style-type: none"> • 테스트 기술 • 테스트 인프라
품질	측정 가능하고 정량화를 통한 품질 보증을 위한 프로세스 영역 그룹 <ul style="list-style-type: none"> • 측정 • 품질 보증
개선	지속적인 결함 예방과 테스트 프로세스 개선 작업을 통한 품질 제어를 위한 프로세스 영역 그룹 <ul style="list-style-type: none"> • 결함 예방 • 품질 제어 • 테스트 프로세스 최적화

개발 생명주기 모델로 잘 알려진 V 모델[8]을 기반으로 설명한 것이다.

1) 성숙도 수준 1: Initial

성숙도 수준 1인 조직의 테스트 프로세스는 독립된 테스트 조직 및 적합한 테스트 활동이 존재하지 않으며, 즉흥적인 디버깅을 테스트링으로 인식하는 단계를 의미한다. 이 단계의 테스트 목적은 단순히 소프트웨어가 작동하는 것을 보여줄 뿐이며 소프트웨어의 품질보증 없이 프로덕트가 배포되는 품질인식이 결여된 단계라 할 수 있다.

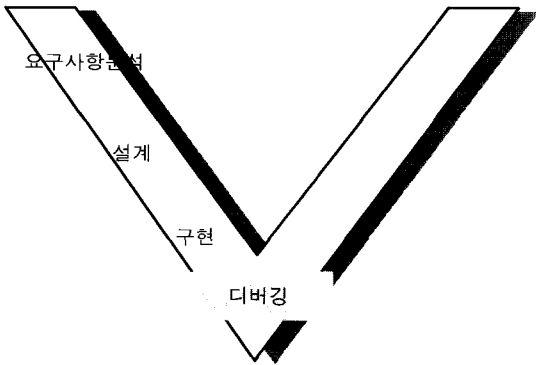


그림 3 성숙도 수준 1의 테스트 프로세스

2) 성숙도 수준 2: Managed

성숙도 수준 2인 조직의 테스트 프로세스는 테스트 활동을 디버깅과 구분하여 구현 이후의 독립된 단계로 정의한다. 그러나, 여전히 테스트 기법은 수행코드 기반으로 코드에 의존적인 간단한 화이트 테스트 기법이 적용되고, 테스트 계획도 코드가 완성된 이후에 이루어지는 수준이라 할 수 있다.

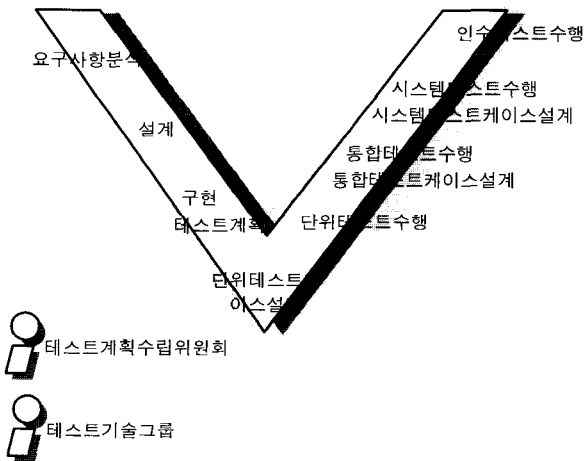


그림 4 성숙도 수준 2의 테스트 프로세스

3) 성숙도 수준 3: Defined

성숙도 수준 3인 조직의 테스트 프로세스는 테스트 활동이 구현 이후가 아닌 소프트웨어 개발 생명주기 전

체와 병행되는 구조로 조직 차원의 표준화된 테스트 프로세스가 정의되는 단계이다. 즉, 테스트 계획은 소프트웨어 개발 계획과 함께 체계적으로 수립되며, 최종 프로덕트에 대한 테스트 활동 뿐만 아니라 소프트웨어 개발 생명주기 단계마다 품질 관리를 위한 활동으로 산출물에 대한 검증 활동도 단계적으로 수행된다.

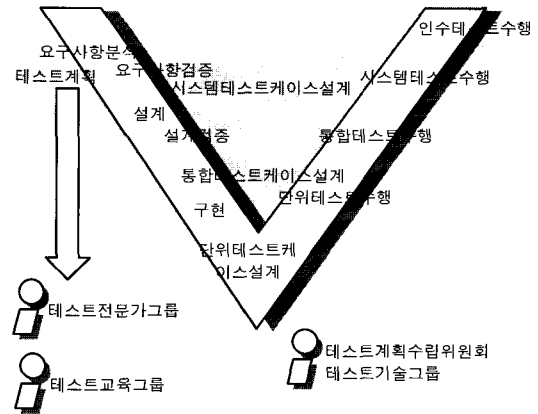


그림 5 성숙도 수준 3의 테스트 프로세스

4) 성숙도 수준 4: Quantitatively Managed

성숙도 수준 4인 조직의 테스트 프로세스는 성숙도 수준 3에서 정의된 표준 테스트 프로세스를 기반으로 단계별 품질 지표 측정이 가능한 정량화된 테스트 프로세스이다. 또한, 소프트웨어 개발 및 테스트 조직과는 독립된 제 3 조직에 의한 감사 활동을 통해 객관적인 품질 평가가 지속적으로 이루어지는 단계라 할 수 있다.

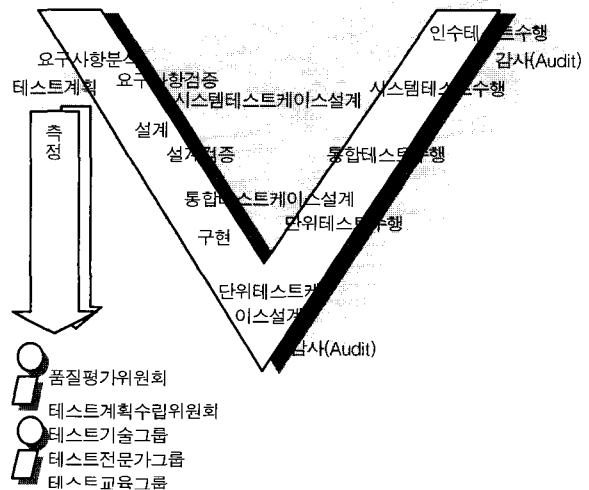


그림 6 성숙도 수준 4의 테스트 프로세스

5) 성숙도 수준 5: Optimizing

성숙도 수준 5인 조직의 테스트 프로세스는 지속적인 결함 예방과 테스트 프로세스 개선 작업을 통한 품질 예측 및 제어가 이루어지는 최상의 단계이다.

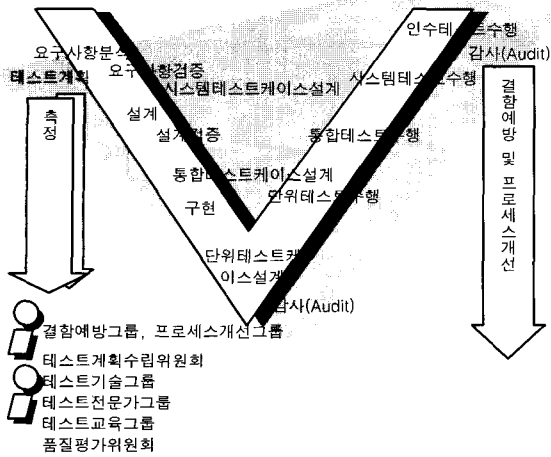


그림 7 성숙도 수준 5의 테스트 프로세스

다음 표 5는 단계적 표현에 따른 성숙도 수준별 테스트 프로세스 영역이다.

표 5 테스트 프로세스 핵심 모델의 단계적 표현

성숙도 수준	테스트 프로세스 영역
성숙도 수준 1	없음
성숙도 수준 2	<ul style="list-style-type: none"> 테스트 기술 테스트 인프라
성숙도 수준 3	<ul style="list-style-type: none"> 테스트 인력 테스트 교육 검증 확증
성숙도 수준 4	<ul style="list-style-type: none"> 감사 측정 품질평가
성숙도 수준 5	<ul style="list-style-type: none"> 결함 예방 품질 제어 테스트 프로세스 최적화

4. 테스트 프로세스 개선 로드맵

본 장에선 테스트 프로세스 핵심 모델을 토대로 단계적인 테스트 프로세스를 개선을 위한 로드맵을 제안한다. 테스트 프로세스 개선 로드맵은 테스트 프로세스 핵심 모델의 프로세스 성숙도 수준에 근거하여 성숙도 수준 1부터 5까지의 단계적 개선을 목표로 한다. 그림 8은 테스트 프로세스 핵심 모델의 프로세스 성숙도 수준과 제안하는 테스트 프로세스 개선 단계와의 관계이다.

테스트 프로세스 개선 로드맵은 3단계로 제안하며, 테스트 프로세스 핵심 모델의 성숙도 수준 1부터 3은 조직 내에 프로세스가 표준으로 정립되는 과정이기 때문에 하나의 개선 단계로 정의한다. 성숙도 수준 4는 표준 프로세스 준수를 통해 제품 품질이 정량적인 측정과 평가

가 가능한 단계로 개선 단계 2로 정의한다. 성숙도 수준 5는 축적된 정량적 데이터를 기반으로 미래 품질을 예측하고 결함 예방이 가능한 최적의 프로세스 단계로 최종 개선 단계로 정의한다.

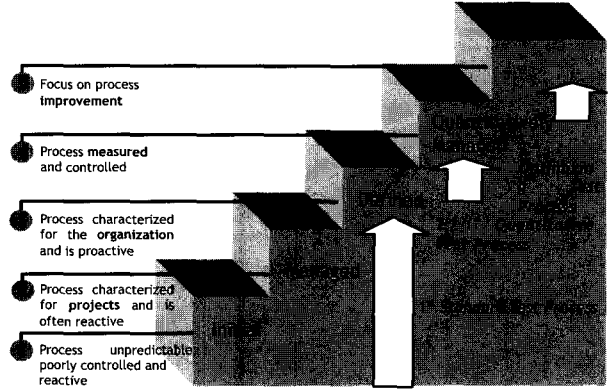


그림 8 테스트 프로세스 단계적 로드맵

4.1 개선단계 1: 표준 테스트 프로세스

첫 번째 개선 단계는 테스트 프로세스 핵심 모델의 성숙도 수준 3을 목표로 하는 표준 테스트 프로세스 정립이다. 표준 테스트 프로세스란 조직 내에 표준 제품 개발 프로세스와 통합된 테스트 프로세스를 의미한다. 표 6은 표준 테스트 프로세스를 위한 주요 테스트 프로세스 영역 별 능력 수준 3을 위한 핵심 개선 사항이다.

표 6 개선 단계 1을 위한 핵심 개선 사항

프로세스 영역	핵심 개선 사항
테스트 인력	테스트 프로세스 단계별 스테이크 홀더와 활동 정의
검증	테스트 레벨별 테스트 기반 문서로 사용될 개발 문서와 검증되어야 할 테스트 요구사항 정의
확증	반복되는 테스트 활동에 대한 테스트 목표, 범위와 대상 정의
테스트 기술	반복되는 테스트 활동에 대한 적합한 테스트 기술 정의
테스트 인프라	반복되는 테스트 활동에 대한 적합한 테스트 환경 정의

4.2 개선단계 2: 정량적 테스트 프로세스

두 번째 개선 단계는 테스트 프로세스 핵심 모델의 성숙도 수준 4를 목표로 하는 정량적 테스트 프로세스 정립이다. 정량적 테스트 프로세스란 표준 테스트 프로세스에 따라 제품을 개발할 때 프로세스 단계별로 품질 보증을 위한 데이터 측정과 평가가 이루어지는 프로세스를 의미한다. 표 7은 정량적 테스트 프로세스를 위한 주요 테스트 프로세스 영역별 능력 수준 4를 위한 핵심 개선

사항이다.

표 7 개선 단계 2를 위한 핵심 개선 사항

프로세스 영역	핵심 개선 사항
테스트 입력	품질 보증 활동을 주도할 품질 보증 위원회 정의
검증	검증 활동을 관리할 정량화된 지표 측정 및 평가 기준 정의
확증	확증 활동을 관리할 정량화된 지표 측정 및 평가 기준 정의
테스트 기술	도구를 활용한 정량화된 테스트케이스 정의
테스트 인프라	도구를 활용한 정량화된 테스트 인프라 정의

4.3 개선단계 3 : 최적화된 테스트 프로세스

세 번째 개선 단계는 테스트 프로세스 핵심 모델의 성숙도 수준 5를 목표로 하는 최적화된 테스트 프로세스 정립이다. 최적화된 테스트 프로세스란 정량적 테스트 프로세스에 따라 제품 품질을 측정하고 평가하면서 축적된 정보를 기반으로 제품 품질을 미리 예측하고, 제품이 갖는 내재된 결함을 미리 예방할 수 있는 프로세스를 의미한다. 표 8은 최적화된 테스트 프로세스를 위한 주요 테스트 프로세스 영역 별 능력 수준 5를 위한 핵심 개선 사항이다.

표 8 개선 단계 3을 위한 핵심 개선 사항

프로세스 영역	핵심 개선 사항
검증	테스트 프로세스 개선, 결함 예방, 품질 예측을 주도할 그룹 정의
테스트 입력	최적화된 검증 절차 정의 검증 활동에 따른 결함 예방 메커니즘 정의
확증	최적화된 확증 절차 정의 확증 활동에 따른 결함 예방 메커니즘 정의
테스트 기술	제품별 최적화된 표준 테스트 케이스 재사용 메커니즘 정의
테스트 인프라	제품별 최적화된 표준 테스트 인프라 재사용 메커니즘 정의

제안하는 테스트 프로세스 개선 로드맵은 적용되는 도메인과 제품에 따라 테일러링 되어야 하며 본 논문에서는 첫 번째 개선 단계인 '표준 테스트 프로세스'에 대해 최근 새로운 소프트웨어 분류로 자리잡고 있는 임베디드 소프트웨어 분야에 대한 테스트 프로세스를 정립하였다. 또한, 나머지 개선 단계에 관한 연구도 진행할 예정이다.

4.4 임베디드 소프트웨어를 위한 표준 테스트 프로세스

본 절에서는 테스트 프로세스 개선 로드맵의 첫 단계

로 임베디드 소프트웨어를 위한 테스트 프로세스를 제안한다. 임베디드 소프트웨어를 위한 테스트 프로세스는 개선 단계 1의 핵심 개선 사항을 중심으로 임베디드 소프트웨어 특성을 고려한 테스트 프로세스이어야 한다.

현재 국내 임베디드 소프트웨어 개발 현황이 제품 내의 임베디드 소프트웨어의 특성을 테스트 프로세스에 반영하지 못한 면에서 문제점이 발생하고 있다.

임베디드 소프트웨어의 핵심 특성은 타겟 하드웨어의 개발 상황에 따라 테스트 환경이 변화하는 제약에 있다는 점에서 시작한다. 타겟 하드웨어가 배포되기 전 소프트웨어의 테스트 환경은 하드웨어를 대체할 시뮬레이션 환경이나 레퍼런스 보드 환경으로 제한된다. 그러나, 현실에선 시뮬레이션 환경이나 레퍼런스 보드 환경은 소프트웨어의 기능을 100% 테스트하기에는 많은 제약이 존재하기 때문에, 소프트웨어의 올바른 테스트를 위해선 조기에 타겟 하드웨어의 보유가 절실히 요구된다.

그러나, 임베디드 시스템은 타겟 하드웨어도 소프트웨어와 동일한 시기에 병행되어 개발되기 때문에 소프트웨어 테스트를 위해 조기에 타겟 하드웨어를 확보하는 것은 현실적으로 불가능하며, 많은 경우 소프트웨어 개발 후반에 이르기까지 실제 타겟 하드웨어 상에서 테스트하지 못하는 상황도 흔히 발생한다.

위에 언급한 특징이 임베디드 소프트웨어의 테스트를 어렵게 하는 점이며 결과적으로 제품의 품질을 저하시키는 근본 원인으로 작용한다. 역으로, 하드웨어 테스트를 위해서도 소프트웨어가 필요로 되기 때문에 소프트웨어의 잦은 배포 원인으로 작용한다. 이렇듯, 일반적인 소프트웨어와 달리 임베디드 소프트웨어는 하드웨어와의 상호 의존적이며 병행 개발되는 특성 때문에 잦은 코드 배포 상황이 발생하며, 이에 따라 반복적 테스트가 일어난다. 그러므로, 임베디드 소프트웨어 테스트의 성공은 반복적 테스트를 얼마나 효율적이고 체계적으로 하느냐에 좌우된다.

제안하는 테스트 프로세스는 임베디드 소프트웨어의 반복적 테스트 현상을 배포되는 소프트웨어 코드의 특성에 따라 4 단계의 테스트 단계로 테스트 목표와 테스트 범위를 체계적으로 정립한 반복 검증되는 테스트 프로세스이다.

본 논문에서는 개발 프로세스 상에서 배포되는 임베디드 소프트웨어의 배포 코드 유형을 다음과 같은 5가지로 정의한다. 첫째, 개발 프로세스 초기에는 문서 기반의 모델로 정의된다. 둘째, 정의된 모델의 확증(validation)을 위해 간단한 프로토타입인 α 코드로 배포된다. 셋째, 프로토타입에 대한 확증이 끝나면 임베디드 소프트웨어 기능을 모두 갖춘 β 코드로 배포된다. 넷째,

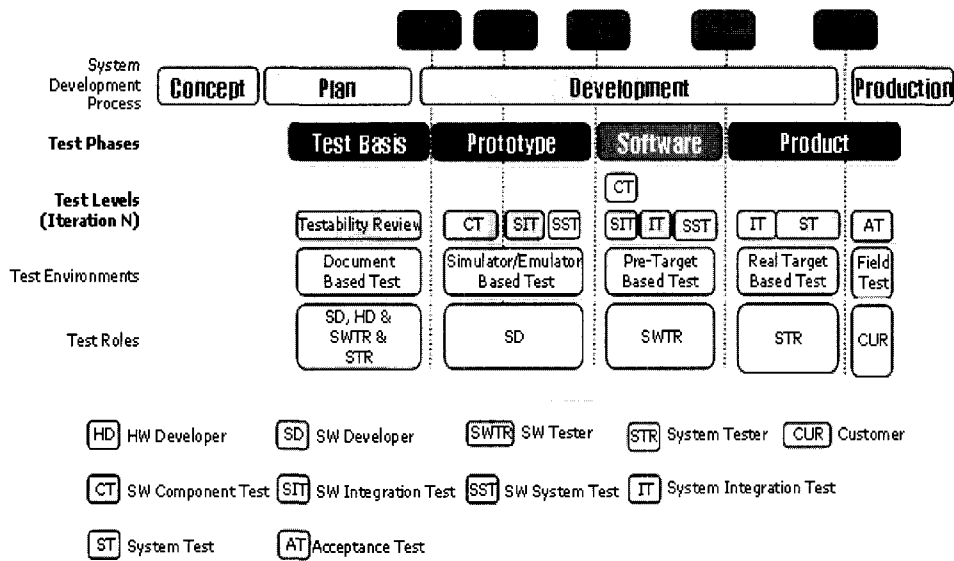


그림 9 임베디드 소프트웨어를 위한 테스트 프로세스

표 9 임베디드 소프트웨어의 테스트 단계

테스트 단계	배포 코드	하드웨어 상태	테스트 목적	반복적 테스트 활동
테스트 기반 단계	모델	하드웨어와 독립적 (하드웨어 없음)	개발 문서가 테스트케이스 선정에 필요한 정보가 포함되어 있는지에 대한 테스트 가능성 검증	프로덕트 요구 명세서 검증 시스템 설계서 검증 소프트웨어 요구 명세서 검증 소프트웨어 기본 설계서 검증 소프트웨어 상세 설계서 검증
프로토타입 테스트 단계	α 코드	시뮬레이터/이플래이터 필요 (하드웨어 개발 진행 중)	소프트웨어 개발자에 의한 '소프트웨어 요구 기능을 중심으로 모델 구현의 타당성' 검증	소프트웨어 컴포넌트 테스트 소프트웨어 통합 테스트 소프트웨어 시스템 테스트 코드
소프트웨어중심 테스트 단계	β 코드	프리-타겟 (하드웨어 검증 중)	소프트웨어 테스트 전문가 그룹에 의한 '소프트웨어 요구 기능을 100% 모두 만족하는 가'를 중심으로 객관적인 소프트웨어 인증	소프트웨어 컴포넌트 테스트 소프트웨어 통합 테스트 시스템 통합 테스트 소프트웨어 시스템 테스트
프로덕트중심 테스트 단계	γ 코드	실제-타겟 (하드웨어 검증 완료)	시스템 테스트 전문가 그룹에 의한 '프로덕트 요구 기능을 100% 모두 만족하는가'를 중심으로 객관적인 프로덕트 인증	시스템 통합 테스트 시스템 테스트 인수 테스트

하드웨어와 통합된 임베디드 시스템인 γ 코드로 배포된다. 다섯째, 모든 검증이 완료되어 최종 프로덕트에 내장되는 양산 코드(ship release code)로 배포되게 된다.

그림 9와 표 9와 같이 제품 개발 프로세스에 통합된 테스트 프로세스를 최종 양산 코드를 제외한 4가지 배포 코드를 기준으로 관련되는 인력, 반복되는 검증 및 검증

활동, 테스트 인프라를 정의하였다.

4.4.1 테스트 기반 단계

테스트 기반 단계는 임베디드 소프트웨어의 요구사항과 구조를 정의한 개발 문서에 대해 향후 테스트케이스 선정을 위한 테스트 기반 문서(test basis)로 활용할 수 있는지를 검증(verification)하는 단계이다. 그림 10은 테스트 기반 단계에서의 모델 검증 활동을 정의하였다.

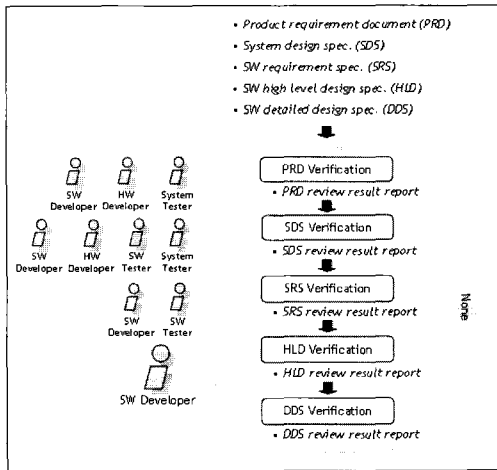


그림 10 테스트 기반 단계

4.4.2 프로토타입 테스트 단계

프로토타입 테스트 단계는 개발 초기에 소프트웨어 개발자 책임 하에 프로토타입 개발을 통해 임베디드 소프트웨어의 요구사항을 검증하는 것을 핵심 목표로 한다.

임베디드 시스템의 요구사항 변경은 소프트웨어 자체 변경 외에도 하드웨어 변경에 따른 소프트웨어 변경이 새로운 위험 원인으로 작용하기 때문에, 일반적인 소프트웨어의 경우 보다 변경 위험이 더 크다고 할 수 있다. 이 단계의 테스트는 프로토타입인 α 코드를 대상으로 '소프트웨어 컴포넌트 테스트, 소프트웨어 통합 테스트, 소프트웨어 시스템 테스트'의 소프트웨어 테스트를 위한 테스트 레벨을 반복한다. 테스트 환경의 경우 프로토타입 테스트 시점은 하드웨어 역시 요구사항을 검증하는 단계라 할 수 있기 때문에 검증된 타겟 하드웨어가 존재하지 않는다. 그렇기 때문에 프로토타입 테스트 단계는 하드웨어와 독립되어 소프트웨어 요구사항 중심의 테스트를 보증하기 위해선 반드시 하드웨어를 대체할 시뮬레이션 환경이 필수적으로 요구된다. 그림 11은 프로토타입

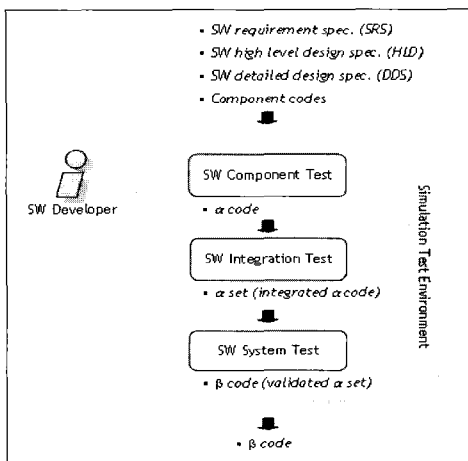


그림 11 프로토타입 테스트 단계

입 테스트 단계의 반복되는 테스트 활동과 테스트 인프라, 테스트 인력을 정의한 것이다.

4.4.3 소프트웨어 중심 테스트 단계

소프트웨어 중심 테스트 단계는 소프트웨어 테스트 전문가의 책임 하에 소프트웨어를 제품 테스트 전문가에게 인수하기 전, 소프트웨어 기능이 완벽하게 수행되는 것을 최종 검증하여 소프트웨어 인수하는 것을 핵심 목표로 한다.

임베디드 시스템 전체 개발 프로세스 상 소프트웨어 테스트 단계는 하드웨어 역시 검증 단계이기 때문에 검증이 덜된 하드웨어 환경에서 소프트웨어를 탑재하여 테스트하게 된다. 그렇기 때문에 하드웨어와 소프트웨어의 통합이 불가피하게 발생하며, 소프트웨어 시스템 테스트 이전에 간단히 하드웨어와 소프트웨어 인터페이스를 확인하는 작업인 시스템 통합 테스트가 우선되어야 한다. 그러나, 프리-타겟의 특성상 하드웨어가 100% 검증된 것이 아니기 때문에 이 테스트 단계에서의 시스템 통합 테스트 비중은 낮으며, 시스템 통합 테스트를 위한 테스트 케이스 중 부분 집합을 활용한다. 시스템 통합 테스트는 프로덕트 테스트 단계에서 집중적으로 테스트되어야 한다.

그림 12는 소프트웨어 테스트 단계의 반복되는 테스트 활동과 테스트 인프라, 테스트 인력을 정의한 것이다.

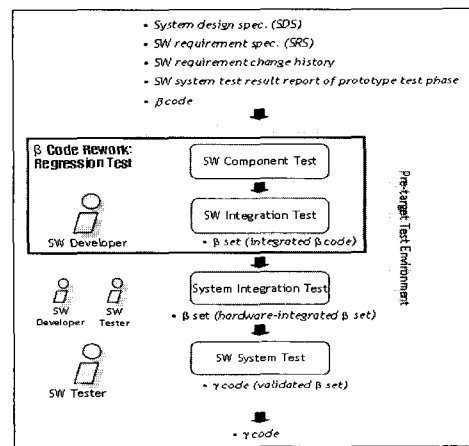


그림 12 소프트웨어 중심 테스트 단계

4.4.3 프로덕트 중심 테스트 단계

프로덕트 중심 테스트 단계는 임베디드 시스템의 구성 요소인 하드웨어와 소프트웨어가 독립적으로 검증이 된 상황에서 시스템 테스트 전문가 책임 하에 완제품 레벨에서 제품 요구 명세서에 정의된 모든 제품 기능을 완벽하게 수행하는 것을 검증하는 것을 목표로 한다. 경우에 따라 실제 프로덕트가 사용될 사용 환경 (field)에 프로덕트를 설치한 후 프로덕트를 인수 받을 고객이나, 제 3의 품질 인증 기관의 책임 하에 수행되는 프로덕트 인수

테스트가 최종 프로덕트 확증을 위해 포함되기도 한다.
그림 13은 프로덕트 테스트 단계의 반복되는 테스트 활동, 테스트 인프라, 테스트 인력을 정의한 것이다.

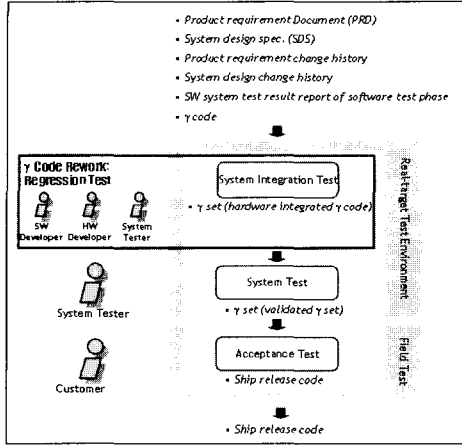


그림 13 프로덕트 중심 테스트 단계

5. 결 론

테스트 프로세스는 프로덕트 개발 과정과 밀접한 연관을 맺고 생산성에 직접 영향을 미치기 때문에 다른 어떤 프로세스보다도 개발 과정에 바로 쉽게 적용할 수 있는 실용적인 프로세스이어야 한다. 이를 위해 본 논문에서는 프로덕트 품질과 직접적인 연관을 맺는 테스트 프로세스 품질 개선을 위해 다음과 같은 개선 방안을 제안하였다.

첫째, 테스트 프로세스 개선을 위한 프레임워크를 개발하였다. 제안하는 '테스트 프로세스 개선 프레임워크'는 테스트에 집중된 개선 프랙티스를 정의한 '테스트 프로세스 핵심 모델'과 능력 수준 및 성숙 수준을 평가하기 위한 '평가 기준', 체크리스트 중심의 '평가 스크립트'의 3가지 구성요소로 정의되며 본 논문에서는 '테스트 프로세스 핵심 모델'을 중심으로 기술하였다.

둘째, 테스트 프로세스 핵심 모델을 토대로 성숙도 수준에 기반한 테스트 프로세스 개선 로드맵을 제안하였다. 테스트 프로세스 개선 로드맵은 '표준 테스트 프로세스, 정량적 테스트 프로세스, 최적화된 테스트 프로세스'의 3단계로 개선을 제안한다.

셋째, 첫 번째 개선 단계로 최근 비중이 높아지고 있는 임베디드 소프트웨어의 테스트 프로세스 개선을 위해 임베디드 소프트웨어에 특화된 '표준 테스트 프로세스'를 제안하였다.

현재 제안하는 '테스트 프로세스 개선 프레임워크'는 국내 산업체의 테스트 프로세스 실태 조사에 적용되어서 실효성이 증명되었고, 보다 더 개발 현장에 실용적일 수 있도록 지속적으로 보완, 정제되고 있다. 또한, 사용자

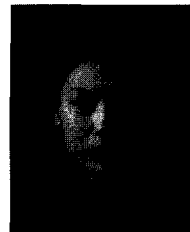
편의를 위해 '테스트 프로세스 개선 프레임워크'를 지원 하는 평가 자동화 도구를 개발 중에 있다.

향후 '테스트 프로세스 개선 프레임워크'에 관한 연구는 제한한 테스트 프로세스 개선 로드맵 중 임베디드 소프트웨어 분야를 위한 '정량적 테스트 프로세스, 최적화된 테스트 프로세스' 정립을 위한 연구로 특화 시킬 예정이다.

참고문헌

- [1] "Software Product Evaluation-Quality Characteristics," ISO/IEC 9126, June, 1998.
- [2] "Software Packages - Quality Requirements and Testing," ISO/IEC 12119, 1994.
- [3] "Software Product Evaluation," ISO/IEC 14598, November, 1997.
- [4] "Software Process Improvement Capability and Determination (SPICE)," ISO/IEC 15504, 1998.
- [5] "CMMI: Capability Maturity Model Integration for System Engineering, Software Engineering, Integrated Product and Process Development and Supplier Sourcing," March, 2002.
- [6] "IEEE Standard for Software Verification and Validation, March," IEEE Std 1012, 1998.
- [7] "V Model: Lifecycle Process Model - Development Standard for IT Systems of the Federal Republic of Germany," 1998.

서주영



1993 이화여자대학교 컴퓨터학 학사
2001 이화여자대학교 컴퓨터학 석사
2004~현재 이화여자대학교 컴퓨터학 박사과정
관심분야: 소프트웨어 프로세스 개선, 임베디드 소프트웨어 테스트, XML, 서비스 기반 아키텍처
E-mail: jyseo@ewhain.net

최병주



1983 이화여자대학교 수학과 학사
1988 Purdue Univ. 전산학 석사
1990 Purdue Univ. 전산학 박사
1995~현재 이화여자대학교 컴퓨터학과 교수
관심분야: 소프트웨어공학, 소프트웨어 테스트, 소프트웨어 및 데이터 품질 측정, 소프트웨어 프로세스, 임베디드 시스템 테스트, 서비스 기반 아키텍처
E-mail: bjchoi@ewha.ac.kr