

패키지 소프트웨어 인증을 위한 테스트 프로세스[†]

이화여자대학교 이윤정* · 최병주**

한국전자통신연구원 장우현

1. 서론

소프트웨어 개발 산업이 발달하고, 산업 각 분야에서 소프트웨어 제품에 대한 필요성이 증가하면서, 보다 좋은 품질의 소프트웨어를 공급하기 위해 소프트웨어 제품에 대한 신뢰도를 규명해야 할 필요성이 커지고 있다. 국제 표준화 기관인 ISO/IEC에서는 소프트웨어 제품과 프로세스의 품질 향상을 위해 소프트웨어 개발 프로세스에 대한 표준을 ISO/IEC 12207에서 제공하고 있다. 표준에서 정의한 프로세스는 개발 방법론(구조적, 객체지향, 컴포넌트 기반 등), 개발 및 테스트 기법, 개발 생명 주기 모델(waterfall, incremental, evolutionary 등), 개발 언어에 상관없이 모두 준수할 수 있도록 되어 있다[1, 2]. ISO/IEC 12207에서는 소프트웨어 개발 생명주기 프로세스 과정에 적용할 수 있는 방법(method), 절차(procedure), 과업(task)을 제공하고, 구현을 위한 구체적인 방법, 절차, 과업에 대해선 '표준의 테일러링'을 하도록 정의하고 있다. ISO/IEC에서는 표준에 대한 지침을 ISO/IEC TR 15271에서 제공하고 있으나, 실제 소프트웨어 개발 및 테스트에 적용할 수준의 프로세스를 위한 구체적인 지침을 제공하진 못하고 있다.

다양한 표준에 정의된 내용을 가지고, 실제 적용할 수 있는 수준의 테스트 프로세스를 구축하는 일은 어렵다. 왜냐하면 표준의 어떤 부분을 어떻게 테일러링 하여야 하는지에 대한 지침이 부족하기 때문이다. 사실 표준은 표준대로 존재할 뿐, 실제 프로세스 정

의에는 제대로 사용되지 못하고 있다. 소프트웨어 품질 측정에 사용할 테스트 프로세스를 생성하기 위하여, 테스트 레벨, 특정 방법론의 선택과 해당 소프트웨어에 필요한 테스트 기술을 추가하는 '표준의 테일러링'을 위한 체계적인 기법이 요구된다.

소프트웨어 제품의 품질에 대한 관심이 점차 증가함에 따라, 국제적 수준의 소프트웨어 품질 인증의 필요성도 증가하고 있다. 정부에서도 소프트웨어 품질 인증제도의 시행을 통하여 일정 기준을 만족시키는 제품에 대한 품질 인증서 및 인증마크를 발급하고 있다. 소프트웨어 품질 인증제도의 시행을 통하여 해당 제품의 신뢰성을 확보하고 판매를 촉진하는 동시에, 소프트웨어 개발 업체의 자발적인 품질 개선 노력을 유도하여 업체의 기술력 향상 및 수출 경쟁력을 강화하고자 하는 시도를 하고 있다. 따라서 이를 위한 체계적이고 정량적인 테스트 프로세스의 필요성이 대두되고 있는 바, 품질을 측정하고 평가하기 위한 테스트 프로세스는 소프트웨어 품질 향상을 위한 필수적인 요소이다. 소프트웨어 개발에 중점을 둔 기존의 테스트 기법과 테스트 프로세스는 소프트웨어 산출물의 품질인증에 적용하기에 적합하지 않기 때문에, 품질 인증을 위한 효율적인 테스트 기법과 테스트 프로세스가 필요하게 되었다. 그러나 소프트웨어 품질 인증 관점에서의 효율적인 테스트 기법과 테스트 프로세스에 대한 연구는 아직 부족한 실정이다.

특정 도메인내의 유사 어플리케이션간의 재사용을 극대화 하고자 하는 노력으로 최근에 프로토타입인 개발 패러다임[3, 4]이 활성화되고 있다. 소프트웨어 품질 인증 테스트 역시, 각 도메인의 특성이나 품질 인증을 위한 테스트 기법등의 공통점과 차이점을 가지고 있기 때문에 이를 잘 분석하여 재사용 가능한 항목을 정의하고, 이러한 항목들을 컴포넌트로 개발

[†] 본 고는 한국전자통신연구원 2001년도 위탁과제 '소프트웨어 테스트 프로세스 재사용 방안 및 생성 자동화 도구 구현에 관한 연구'에 대한 결과물의 일부입니다.

* 학생회원

** 정회원

할 수 있다. 이렇게 개발한 컴포넌트를 이용하여 프로세스 표준에 따른 각 소프트웨어에 적합한 테스트 프로세스를 생성할 수 있다. 이를 통해 소프트웨어 품질 인증 비용을 절감하는 효과를 얻을 수 있으며, 제품 특성에 적절한 테스트 프로세스를 생성함으로써 품질과 신뢰도가 향상된 소프트웨어 제품을 개발하는데 활용할 수가 있다.

본 고에서는 프로덕트 라인 개발 패러다임을 활용한 재사용 가능한 패키지 소프트웨어 인증 테스트 프로세스의 생성에 대하여 살펴보고자 한다. “패키지 소프트웨어 인증을 위한 테스트 프로세스 재사용 방안”을 “테스트 프로세스 컴포넌트 개발”과 “테스트 프로세스 개발”의 2단계로 구성하였으며, 전체 프로세스는 “테스트 프로세스 생성 도구”를 구현하여 관리한다. 2장에서는 테스트 프로세스 컴포넌트 개발에 대하여 살펴보고, 3장에서는 테스트 프로세스 개발에 대하여, 4장에서는 테스트 프로세스 자동 생성 도구에 대하여 살펴보겠다.

2. 테스트 프로세스 컴포넌트 개발

본 고에서는 소프트웨어 품질 인증을 위해 재사용 가능한 테스트 프로세스를 생성하기 위하여, 그 기반이 되는 컴포넌트를 구축하는 단계를 재사용 대상 항목 개발, 컴포넌트 개발, 테스트 프로세스 생성 플랜 개발의 세 단계로 나누어 구성하였다.

- ‘재사용 대상 항목의 개발’은 테스트 프로세스의 체계적인 재사용을 위하여 재사용 가능한 항목을 정의하는 단계이다.
- ‘컴포넌트 개발’은 정의한 재사용 항목을 기반으로 재사용의 대상이 되는 컴포넌트들을 개발하는 단계이다.
- ‘테스트 프로세스 생성 플랜 개발’은 개발된 컴포넌트들을 기반으로 테스트 프로세스를 개발할 때 지침이 될 테스트 프로세스 생성 플랜을 구축하는 단계이다.

컴포넌트 개발을 위해서는 표준, 테스트 레벨, 도메인의 특성, 테스트 기법이 입력으로 사용되며, 테스트 프로세스 생성 플랜 작성을 위해서는 해당 컴포넌트에 대한 맞춤 패턴과 생성 기법이 입력으로 사용된다.

2.1 재사용 대상 항목 개발

본 고에서는 소프트웨어 인증을 위한 테스트 프로세스를 체계적으로 재사용하기 위하여, 재사용 대상 항목을 그림 1과 같이 5단계로 정의하였다. 먼저, ‘프로세스 표준’을 1단계의 재사용 항목으로 정의하였다. 이는 ISO/IEC 12207, MIL-STD 498 및 ANSI/IEEE SVVP 등 프로세스 관련 표준들[1, 5, 6]로부터 필수적인 테스트 프로세스를 작업, 절차, 산출물 등을 기준으로 재사용을 위한 핵심 테스트 프로세스로 추출하여 ‘핵심 테스트 프로세스 컴포넌트’로 개발하는 것이다. 제 2단계로는 ‘테스트 레벨’을 재사용 항목으로 정의하였다. 이는 단위 테스트, 통합 테스트, 시스템 테스트 및 인증 테스트 등 테스트 레벨의 특성을 테스트 프로세스 요소와 기법을 중심으로 재사용 대상으로 추출하여 ‘테스트 레벨 플러그인 컴포넌트’로 개발하는 것이다. 제 3 단계로는 ‘테스트 대상 소프트웨어가 속한 도메인에 대한 공통 특성’들을 재사용 대상 항목으로 정의하였다. 이는 테스트 대상 소프트웨어가 속한 도메인의 특징들을 지침과 기법을 기준으로 추출하여 ‘도메인 플러그인 컴포넌트’로 개발하는 것이다. 제 4 단계로는 ‘테스트 기법’을 재사용 대상으로 정의하였다. 이는 소프트웨어 인증 등 다양한 테스트 목적과 테스트 레벨, 도메인, 소프트웨어 특성에 따른 테스트 기법의 특징으로부터 테스트 기법을 재사용 대상으로 추출하여 ‘테스트 기법 플러그인 컴포넌트’로 개발하는 것이다. 마지막 단계는 ‘특정 소프트웨어의 특성을 맞추기 위한 단계’로, 이는 인증 대상 소프트웨어의 특성을 소프트웨어 요구사항으로부터 추출하여, 최종 결과물인 테스트 프로세스 개발의 입력으로 이용하기 위한 것이다.

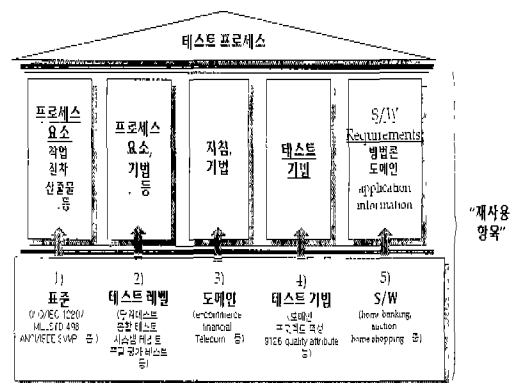


그림 1 테스트 프로세스 재사용 항목

재사용 대상 항목을 5단계로 세분화하여 정의함으로써 테스트 프로세스 컴포넌트의 재사용성을 높이고 각 소프트웨어에 보다 적합한 테스트 프로세스를 생성할 수 있다. 즉, 재사용 항목의 정의를 통해서, 표준을 준수하며, 테스트 레벨과 테스트 대상 소프트웨어가 속한 도메인에 따른 지침과 테스트 기법을 반영한, 품질 인증을 위한 소프트웨어 테스트 프로세스를 생성할 수 있도록 하는 기반을 제공할 수 있다.

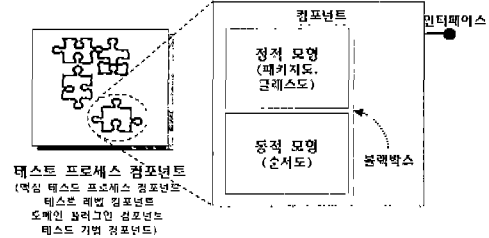


그림 2 테스트 프로세스 컴포넌트

2.2 컴포넌트 개발

컴포넌트 개발 단계에서는 정의한 재사용 항목에 따라 재사용의 대상이 되는 컴포넌트들을 개발한다. 이는 표준을 반영한 핵심 테스트 프로세스 컴포넌트, 단위 테스트, 통합 테스트, 시스템 테스트 및 인증 테스트 등 테스트 수준을 반영한 테스트 레벨 컴포넌트, 테스트 대상 소프트웨어가 속한 도메인의 특성을 반영한 도메인 플러그인 컴포넌트와 테스트 기법 플러그인 컴포넌트로 구성된다.

본 고에서는 소프트웨어 테스트 프로세스 재사용을 위한 컴포넌트를 개발하는데, 소프트웨어 프로젝트의 구현 기술로만 활용되어 온 CBD(Component-Based Development) 기술을 적용한다. 이는 CBD 기술이 전형적인 소프트웨어 프로젝트뿐만 아니라 “소프트웨어 프로세스”에도 적용 가능하기 때문이다. 일반적인 컴포넌트들이 설계 산출물을 기반으로 한 실행 코드인 것에 비해, 소프트웨어 테스트 프로세스 재사용을 위한 테스트 프로세스 컴포넌트는 자연어로 기술한 소프트웨어 테스트 프로세스를 객체 지향적으로 모델링하여 UML[7]로 기술한 메타 모델로 정의한다. 소프트웨어 테스트 프로세스를 위한 컴포넌트에 대한 정의는 [정의 1]과 같다[8].

[정의 1] 테스트 프로세스 컴포넌트

테스트 프로세스를 구성하기 위한 컴포넌트는 표준에 따라 반드시 수행되어야 하는 수정 불가능한 ‘블랙박스’ 영역과 프로젝트의 특성에 따라 수정 가능하도록 하는 ‘인터페이스’로 구성한다. 이러한 컴포넌트를 UML로 기술한 메타 모델로 정의하여, 그림 2와 같이 컴포넌트의 정적 모형(structural feature)은 패키지도와 클래스도로, 컴포넌트의 동적 모형(behavioral feature)은 순서도로 표현한다.

테스트 프로세스를 구성하기 위한 컴포넌트들은 재사용 대상 항목에 따른 테스트 레벨, 도메인 특성, 테스트 기법으로부터 특성을 추출하는 단계와, 텍스트 형태로 추출된 특성들을 [8]에서 제안한 “component modeling rule(CMR)”에 따라 컴포넌트로 개발하는 세부 절차를 통해 개발된다.

Component modeling rule(CMR)

- (a) 컴포넌트의 정적 모형은 클래스, 패키지도로 표현한다. 즉, 작업 패키지로 변환하고, 클래스도로 세분화되어 표현한다. 작업의 각 절차는 메소드로, 산출물은 클래스로 표현한다. 지침, 기법에 따른 구체적인 프로세스는 인터페이스 클래스로 표현한다.
- (b) 컴포넌트의 동적 모형은 순서도로 표현한다. 즉, 순서도는 절차의 상호작용을 표현한다.

2.3 테스트 프로세스 생성 플랜 개발

테스트 프로세스 개발 단계에서 테스트 프로세스 컴포넌트를 기반으로 소프트웨어 품질 인증을 위한 테스트 프로세스를 생성하기 위해서는 체계적인 테스트 프로세스 생성 플랜이 필요하다. 각각의 소프트웨어 인증 테스트에 적용할 소프트웨어 테스트 프로세스는 표준을 준수하는 핵심 테스트 프로세스 컴포넌트에 테스트 레벨과 테스트 대상 소프트웨어가 속한 도메인, 테스트 기법의 플러그인 컴포넌트를 컴포넌트 맞춤을 함으로써 생성된다[8].

2.3.1 CBD 맞춤 패턴과 생성 기법

CBD에서 컴포넌트 사이의 연결 관계에 따라 맞춤 패턴이 다르다. 일반적으로 컴포넌트 사이의 관계는

메소드, 클래스와 패키지 단위로 성립하며, 표 1에서 처럼 각 단위별로 연관(association)관계와(또는) 상속(inheritance)관계가 존재한다[9, 10]. 따라서 이들 각 관계에 따라 4가지의 맞춤 패턴, cp1, cp2, cp3, cp4를 정의할 수 있다[8].

표 1 컴포넌트 맞춤 패턴

단위	관 계	맞춤 패턴
메소드 사이	클래스내의 두 메소드 사이의 연관 관계	cp1
클래스 사이	두 클래스 사이의 연관 관계	cp2
	두 클래스 사이의 상속 관계	cp3
패키지 사이	두 패키지 사이의 연관 관계	cp4

일반적인 CBD의 맞춤은 직접 인터페이스를 수정하거나, 플러그인 컴포넌트를 연관 관계로 연결하거나, 상속 관계로 연결함으로써 수행될 수 있다. 따라서 핵심 테스트 프로세스 컴포넌트와 플러그인 컴포넌트를 연관/상속 관계로 연결함으로써 테스트 프로세스를 생성할 수 있다. 각 맞춤 패턴별 생성 기법은 표 2와 같으며, 이를 기반으로 소프트웨어 테스트 프로세스를 생성하기 위한 테스트 프로세스 생성 플랜을 작성한다.

표 2 테스트 프로세스 생성 기법

맞춤 패턴	생 성 기 법
cp1	1) 새로운 메소드를 관련 클래스도에 연관 관계로 추가 2) 새로운 메소드 호출을 관련 순서도에 추가
cp2	1) 새로운 클래스를 관련 클래스도에 연관관계로 추가 2) 새로운 메소드 호출을 관련 순서도에 추가
cp3	1) 새로운 클래스를 관련 클래스도에 상속관계로 추가
cp4	1) 새로운 패키지를 관련 패키지도에 연관관계로 추가 2) 새로운 순서도를 추가

2.3.2 테스트 프로세스 생성 플랜

핵심 테스트 프로세스 컴포넌트와 테스트 레벨/도메인/테스트 기법 플러그인 컴포넌트를 표 1의 맞춤

패턴에 따라 연관/상속 관계로 연결하는 표 2의 생성 기법을 기반으로 테스트 프로세스 생성 플랜을 수립한다. 테스트 프로세스 생성 플랜은 소프트웨어 테스트 프로세스 개발 단계에서 테스트 프로세스를 생성하는데 이용된다.

3. 소프트웨어 테스트 프로세스 개발

테스트 프로세스 개발 단계에서는 이미 개발한 테스트 프로세스 컴포넌트들과 소프트웨어 요구사항을 기반으로 테스트 프로세스 생성 플랜에 따라 소프트웨어 품질 인증을 위한 테스트 프로세스를 생성한다.

3.1 테스트 프로세스 테일러링 체계

프로세스 생성 단계에서는 개발한 테스트 프로세스 컴포넌트와 테스트 프로세스 생성 플랜을 기반으로 테스트 프로세스를 생성한다. 그림 3과 같이 소프트웨어 인증 테스트 프로세스는 핵심 테스트 프로세스 컴포넌트에 개발한 컴포넌트 중 단위 테스트, 통합 테스트 시스템 테스트 및 인증 테스트 등 테스트 목적을 반영하는 ①테스트 레벨 플러그인 컴포넌트, 테스트 대상 소프트웨어에 해당하는 ②도메인 플러그인 컴포넌트, 품질 추정을 위한 ③테스트 기법 플러그인 컴포넌트를 2.3절의 테스트 프로세스 생성 플랜에 따라 각각 맞춤을 함으로써 생성된다.

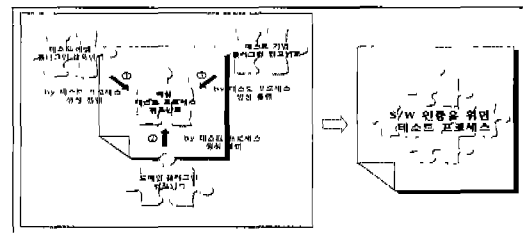


그림 3 컴포넌트 맞춤을 통한 테스트 프로세스 생성

품질 인증을 위한 테스트 프로세스 테일러링 체계는 그림 4에서 처럼 ‘핵심 테스트 프로세스 컴포넌트에 1)테스트 레벨 → 2) 도메인 → 3) 테스트 기법’의 순으로 이루어지게 된다. 이렇게 테스트 레벨, 도메인, 테스트 기법 단계까지 맞춤하여 생성된 소프트웨어 테스트 프로세스에 마지막으로 해당 소프트웨어의 요구사항들을 입력으로 하여 패키지 소프트웨어 인증을 위한 테스트 프로세스를 생성한다.

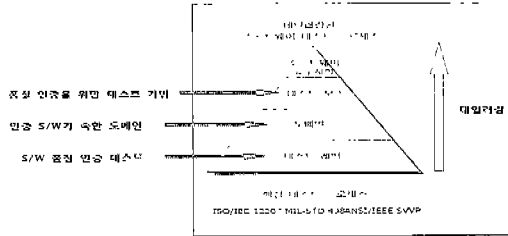


그림 4 인증을 위한 테스트 프로세스 체계도

4. 소프트웨어 테스트 프로세스 생성 자동화 도구

개발한 테스트 프로세스 컴포넌트의 저장과 테스트 프로세스 개발의 전체 프로세스는 '소프트웨어 테스트 프로세스 생성 자동화 도구'를 구현하여 관리한다. 이 도구는 테스트 프로세스 컴포넌트 개발 단계에서 생성되는 컴포넌트들을 저장소에 관리하며, 테스트 프로세스 개발 단계의 전체 프로세스, 즉 테스트할 소프트웨어에 적합한 테스트 수준, 도메인과 테스트 기법에 해당하는 컴포넌트를 선정하여 테스트 프로세스를 생성하는 프로세스를 관리한다.

'소프트웨어 테스트 프로세스 생성 자동화 도구'는 기술한 테스트 프로세스 테일러링 기법을 적용하여 소프트웨어 개발 도메인과 테스트 기법이 테일러링된 테스트 프로세스를 자동 생성할 수 있도록, XML 기술을 이용하여 개발하였다. 이 도구는 UML notation으로 표현된 테스트 프로세스 컴포넌트를 입력으로 받아 XML로 변환한 후 XML 수준에서 테일러링 패턴에 따라 해당 테일러링 시나리오를 수행함으로써, 각 소프트웨어에 적합한 테스트 프로세스로 테일러링한다. 테일러링이 완료된 테스트 프로세스 메타 모델은 사용자가 보기 편리하도록 TXT나 HTML 형태의 문서로 자동 생성된다.

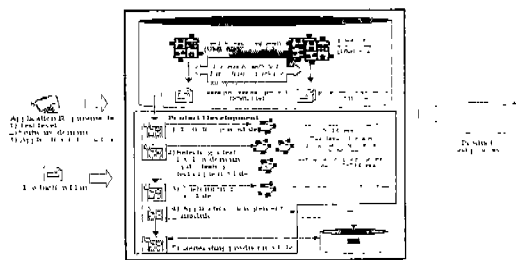


그림 5 테스트 프로세스 생성 도구의 프로토타입

5. 결론

본 고에서는 패키지 소프트웨어 인증을 위한 테스트 프로세스 생성 방안을 테스트 대상 컴포넌트 개발과 테스트 프로세스 개발의 두 단계로 나누어 기술하였고, 이를 자동화 하는 소프트웨어 테스트 프로세스 생성 도구에 대하여 살펴보았다.

프로덕트 라인 개발 패러다임을 적용하여, 프로세스를 구성하는 요소들의 공통점과 차이점의 분석을 통해 표준 프로세스, 도메인, 테스트 기법과 소프트웨어 특징으로 재사용 항목을 세분화해서 테스트 프로세스 생성을 위한 컴포넌트를 구성하였으며, 이에 따른 체계적인 컴포넌트 개발에 대하여 살펴보았다. 또한, 재사용 단위로 개발된 컴포넌트를 가지고, 패키지 소프트웨어 인증을 위한 테스트 프로세스를 생성하기 위한, CBD의 맞춤 패턴에 기반한 테스트 프로세스 생성 플랜에 대하여 살펴보았다. 테스트 프로세스 개발 단계에서는 프로세스 생성 자동화 도구가 개발된 테스트 프로세스 컴포넌트와 소프트웨어 요구사항을 기반으로 테스트 프로세스 생성 플랜에 따라 패키지 소프트웨어 품질 인증을 위한 테스트 프로세스를 생성한다.

참고문헌

- [1] ISO/IEC 12207 : Information Technology - Software Life Cycle Process.
- [2] ISO/IEC TR 15271 : Information Technology - Guide for ISO/IEC 12207.
- [3] CMU/SEI, Product Line Practice, <http://www.sei.cmu.edu/plp/>, January 2001.
- [4] John Bergcy, Matt Fisher, Brian Gallagher, Lawrence Jones, Linda Northrop, "Basic Concepts of Product Line Practice for the DoD", CMU/SEI-2000-TN-001, February 2000.
- [5] MIL-STD-498, Software Development and Documentation.
- [6] ANSI/IEEE Std 1012-1986 IEEE Standard for Software Verification and Validation Plan.
- [7] Martin Fowler and Kendall Scott, *UML Distilled : Applying the Standard Object Modeling Language*, Addison-Wesley, Aug. 1997.

- [8] Seo, Jooyoung, and Choi, Byoungju, "Tailoring Test Process By using the Component Based Development Paradigm and XML Technology," 7th Proceeding of Asia-Pacific Software Engineering Conference (APSEC) 2000 in Singapore, pp.356-363, Dec. 2000.
- [9] 윤회진, 최병주, "컴포넌트 기반 개발 개념을 활용한 테스트 프로세스 tailoring", 정보과학회논문지 : 소프트웨어 및 응용, 27(12):1163-1173, 2000.12.
- [10] Desmond Francis D'Souza and Alan Cameron Wills, *Objects, Components, and Frameworks With Uml : The Catalysis Approach*, Addison-Wesley Object Technology Series, Oct. 1998.
- [11] Wolfgang Pree, *Design Patterns for Object-Oriented Software Development*, Addison-Wesley, 1995

이윤정



1992~1996 이화여대 수학과 학사
 1999~현재 이화여대 컴퓨터학과 석사과정
 관심분야 : Software Testing, Software Quality Evaluation, Software Process
 E-mail: yoonjung@ewha.ac.kr

최병주



1983 이화여대 수학과 학사
 1988 Purdue Univ. 전산학 석사
 1990 Purdue Univ. 전산학 (소프트웨어공학 전공) 박사
 1991~1992 삼성종합기술원 선임연구원
 1992~1995 용인대학교 전산통계학과 전임강사
 1995~현재 이화여자대학교 컴퓨터학과 부교수

관심분야 : Object-Oriented Software Testing, Component-based Software Testing, Test Automation, Test Agent System, Data Quality, Software Quality, Software Engineering for Internet, Software Engineering for Embedded System, Safety, Security
 E-mail: bjchoi@ewha.ac.kr

장우현



1982.2 중앙대학교 전자계산학과 졸업(학사)
 1991.8 중앙대학교 컴퓨터공학과 대학원 졸업(석사)
 2001.2 대전대학교 컴퓨터공학과 대학원 졸업(박사)
 1982.3~현재 한국전자통신연구원(ETRI) 책임연구원, S/W 평가기술팀장
 관심분야 : 소프트웨어공학, 컴포넌트 기술, S/W 품질평가기술

E-mail: whjang@etri.re.kr