

소프트웨어 자동 테스트 도구의 발전 로드맵 분석

정 창 신*, 정 순 기**

Development Roadmap Analysis for Automatic Testing Tools of Software

Chang-shin Chung*, Soon-Key Jung**

요 약

소프트웨어 자동 테스트 도구를 이용하여 테스트 프로세스의 전부 또는 일부를 자동화함으로써 테스트 시간의 단축과 테스트 비용을 줄일 수 있다. 그러나 하나의 테스트 도구를 이용하여 이질적인 컴퓨팅 환경에서 다양한 종류의 테스트 요구 사항을 모두 충족시킨다는 것은 현실적으로 불가능하다. 본 논문에서는 소프트웨어 자동 테스트 도구의 분류 기준을 제시하고, 이를 기반으로 상용화된 테스트 도구들의 특성을 조사, 분석하여 테스트 도구의 발전 로드맵을 제시함으로써 테스트 도구들 상호간의 관계성 이해와 향후 테스트 도구들의 발전 방향을 모색하고자 한다.

Abstract

Testing tools help software developers and test engineers to automate a part or the whole of test process and reduce the time and cost of software test. However, no automatic tools meet a variety of test requirements with various domains. In this paper, we suggest a categorical taxonomy of automatic testing tools and analyze numerous commercial testing tools on its basis. Then we propose a development roadmap of software automatic testing tools. As a result of those analysis, we try to understand the complementary relations of the testing tools and seek for the direction of further development of the testing tools.

▶ Keyword : Software Testing Tools, Roadmap

* TTA IT시험연구소 SW시험인증센터(cschung@tta.or.kr)

** 충북대학교 컴퓨터공학과(soonkey@chungbuk.ac.kr)

I. 서론

고품질의 소프트웨어 개발을 위하여 소프트웨어의 품질 관리에 대한 요구가 더욱 증가되고 있다. 신뢰성이 결여된 소프트웨어의 사용은 많은 경제적인 손과 직결되기 때문에 컴퓨터 하드웨어의 잠재적인 계산 능력을 최대한 활용할 수 있는 고품질의 소프트웨어 개발 문제는 소프트웨어 공학 분야에서 중요한 연구과제로 등장하고 있다.

소프트웨어 품질관리에 이용되고 있는 다양한 기법들 중에서 가장 널리 이용되고 있는 기법은 소프트웨어 테스트이다. 소프트웨어 테스트는 과거에는 소프트웨어 개발자의 책임 하에 수행되었으나, 현재 많은 소프트웨어 개발업체에서는 테스트 업무를 별도의 소프트웨어 품질관리팀에 위임하고 있다. 따라서 소프트웨어 개발 조직과 소프트웨어 품질관리 조직의 생산성 제고 문제는 매우 중요한 이슈로 등장하고 있다. 품질관리 조직의 생산성을 향상시키기 위해서는 근본적으로 소프트웨어의 모든 테스트 과정에 자동화된 테스트 도구를 투입해야 한다. 소프트웨어 테스트 시간의 단축이 소프트웨어 품질관리 조직의 생산성 제고에 필수적인 요소 중의 하나라면 테스트 도구들의 자동화로의 변화는 필수적이라고 할 수 있다.

소프트웨어 개발에 CASE 도구의 사용은 이미 보편화되고 있다. 특히 상업적인 측면에서 소프트웨어의 테스트에 필요한 자동화 도구에 대한 연구는 90년대 이후로 급속하게 발전해 왔다. 그러나 현재 상용화된 소프트웨어 자동 테스트 도구들은 기능적인 측면에서 다음과 같은 세 가지 문제점을 가지고 있다.

첫째, 테스트 도구들의 통합(integration)이 어렵다. 상이한 업체들로부터 개발된 테스트 도구들을 통합할 수 있는 현실적인 방법은 거의 없으며, 심지어는 동일한 회사에서 개발된 제품들 간의 통합 가능성도 결여되고 있다 [1][2].

둘째, 기존 테스트 도구를 기반으로 새로운 자동 테스트 도구의 개발이 어렵다. 소프트웨어 테스트는 어플리케이션 영역별로 기술적인 난이도가 다르기 때문에 상용 도구만으로 모든 종류의 소프트웨어를 효과적으로 테스트할 수 없다. 따라서 기존 테스트 도구를 수정, 이용하거나, 새로운 테스

팅 도구를 개발, 이용할 수 있지만 현재의 테스트 도구들이 가지고 있는 기능상의 폐쇄적인 특성 때문에 도구들의 확장 및 변경이 매우 어렵다[1][2].

마지막으로, 기존 자동 테스트 도구들은 표준화된 테스트 케이스들의 개발에 도움을 주지 못한다. 테스트 케이스의 표준 양식을 이용하여 재사용 가능한 테스트 케이스들을 축적, 이용하는 것이 소프트웨어품질관리 조직에서 지식, 노하우(knowhow) 관리의 핵심요소가 되고 있지만 기존 테스트 도구에는 이러한 요소가 부족하다[1][2].

위와 같은 문제점을 해결하기 위해서는 다양한 테스트 도구들의 통합과 기존 도구를 기반으로 새로운 테스트 도구의 개발이 가능해야하며, 표준 테스트 케이스들의 축적을 지원할 수 있는 범용 테스트 프레임워크의 구축이 필요하다. 본 논문에서는 기존 소프트웨어 자동 테스트 도구들의 특성에 대한 광범위한 조사, 분석을 통하여 테스트 도구들의 발전 로드맵을 제시하고자 한다.

제 2장에서는 소프트웨어 자동 테스트 도구의 평가 기준에 대해 살펴보고, 이를 바탕으로 기존 테스트 도구들을 분류한다. 제 3장에서는 소프트웨어의 개발 및 운용환경 변화 추세, 테스트 도구의 자동화 정도 및 소프트웨어의 품질평가에 사용하고 있는 테스트 종류를 분석하여 자동 테스트 도구들의 발전 로드맵을 설정한다. 마지막 제 4장에서는 본 연구의 결과에 대해 기술한다.

II. 테스트 자동화 도구의 분류

1. 테스트 도구의 분류 기준

소프트웨어의 테스트 자동화에 따른 효과를 극대화하려면 자동 테스트 도구에 대한 정확한 기능 평가를 바탕으로 테스트 요구사항에 적합한 테스트 도구를 선택해야 한다. 현재 100개 이상의 자동 테스트 도구들이 상용화되고 있다 [3]. 각각의 도구들은 테스트 목적 및 운용환경이 다르기 때문에 각 도구들의 사용에 앞서 각 도구들에 대한 객관적인 기술 평가가 반드시 선행되어야 한다.

일반적으로 소프트웨어의 테스트 자동화에 영향을 미치는 요인은 크게 세 가지로 나누어 볼 수 있다.

첫째, 소프트웨어 개발 프로세스 관점에서 테스트를 수행하는 단계이다. 소프트웨어 테스트는 그 목적과 대상에 따라 단위(unit) 테스트(white-box 테스트 중심), 통합(integration) 테스트(하향식 또는 상향식 통합), 유효성(validation) 테스트(형상 검토, 수락 테스트) 및 시스템 테스트(회복, 보안, 강도, 성능 테스트)로 나누어진다. 각 테스트 단계에 따라 사용할 테스트 도구에 대한 분류와 평가가 필요하다.

둘째, 전체 테스트 프로세스 중에서 자동화 지원 범위에 따라 테스트 도구들의 분류가 가능하다. 일반적으로 테스트 과정은 테스트 목적의 설정, 테스트 케이스의 설계, 테스트 케이스의 구현, 테스트 수행 및 테스트 결과 분석으로 구성된다. 현재 상용화된 대부분의 테스트 도구들은 테스트 수행과 테스트 결과 분석 과정의 자동화를 지원하고 있으며, 일부 도구들만이 테스트 케이스의 자동설계 과정을 지원하고 있다. 따라서 테스트 과정의 자동화 범위에 따라서 테스트 도구들에 대한 평가가 필요하다.

마지막으로, 테스트 방법에 따라서 테스트 도구들의 분류 및 평가가 필요하다. 테스트 도구에 따라서 테스트 방법상에 장단점이 있으며, 또한 도구 상호간에 보완 관계를 갖는 경우가 많다. 따라서 자동 테스트 도구의 도입에 앞서 도구에서 지원하는 테스트 방법에 대한 평가가 필요하다(4).

(그림 1)은 소프트웨어 자동 테스트 도구에 대한 평가 모델을 나타낸다. 테스트 도구의 자동화 방법 과 테스트 도구의 결정은 테스트 단계, 테스트 프로세스의 자동화 범위 및 테스트 방법의 선택에 의존한다. 이를 달리 표현하면, 자동 테스트 도구의 평가를 위해서는 테스트의 지원 단계, 테스트 프로세스의 지원 범위 및 지원되는 테스트 방법에 대한 분석이 필요하다는 것을 의미한다.

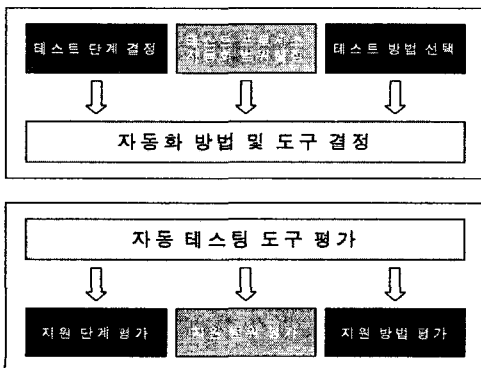


그림 1. 자동 테스트 도구의 평가 모델
Fig. 1 Evaluation Model of Automatic Testing Tools

2. 테스트 도구의 조사

〈표 1〉에서는 상용화된 테스트 도구들 중에서 대표적인 도구만을 나타내며, 이를 테스트 도구의 분석에 이용하였다.

3. 테스트 도구의 분석

상용화된 소프트웨어 자동 테스트 도구의 종류는 테스트 프로세스의 지원 범위에 따라서 다음과 같이 분류할 수 있다.

3.1 테스트 수행의 자동화 도구

테스트 수행 단계를 지원하는 도구이다. 테스트 케이스와 테스트 데이터가 주어진 환경에서 테스트 도구는 정해진 시나리오에 따라 테스트를 수행하며, 테스트 수행 결과를 분석한다. 즉, 가장 많은 시간이 소요되는 테스트의 수행 및 결과 분석 과정을 자동화함으로써 테스트 시간과 노력을 감소시킬 수 있다. 그러나 테스트 케이스 및 테스트 데이터의 자동생성이 불가능하므로 테스트 효과 측면에서 테스트 도구의 도입에 따른 반사 이익이 많지 않다.

3.2 테스트 케이스의 자동생성 도구

테스트 케이스의 설계와 구현을 지원하는 도구이다. 테스트 수행의 자동화와는 달리 테스트 케이스의 자동생성을 통해 테스트 효과를 향상시킬 수 있다. 그러나 이러한 도구들의 상용화는 아직 초기 단계에 있다.

3.3 테스트 관리의 자동화 도구

테스트 관리업무를 지원하는 도구이다. 테스트 일정과 테스트 시나리오 결정, 테스트 케이스와 프로그램 명세간의 관계성 유지, 테스트 결과의 기록 및 재시험 관리 등을 지원한다. (그림 2)는 위에서 제시한 테스트 도구의 평가 모델에 따라 상용화된 소프트웨어 자동 테스트 도구들을 분류한 결과이다. 총 25개의 회사에서 상용화한 104개의 제품을 분석 대상으로 하였다.

표 1. 상용 테스트 도구
Table. 1 Commercial Testing Tools

회사명/제품군	제품명	주요 기능 및 특징
McCabe & Associates	McCabe QA	Metrics
	McCabe Test	Structured unit testing
Mercury Interactive /Testing	ActiveTest	load testing, web application
	Astra LoadTest	Multiple browser, capture and playback, load testing, web testing
	Astra QuickTest	Database verification, capture and playback, web

Mercury Interactive /Testing	LoadRunner	load testing, data-driven testing, EJB testing, XML	
	TestDirector	test plan/case/defect management	
	WinRunner	Function testing with capture and replay, WAP	
	Xrunner	X window GUI testing, record and playback	
Compuware /QACenter	QARun	test execution, e-commerce/ERP/client/server	
	QALoad	load testing, client/server	
	TestPartner	Functional testing, visual scripting, MS/Java/web	
	QADirector	test process management	
	Reconcile	test planner	
	File-AID/CS	data management, GUI for database, client/server	
	QAHyperstation	record-based test creation/execution/analysis, unit, concurrency, integration, migration, capacity, performance testing, S/390 applications	
	QAHyperstation +	extend QAHyperstation with GUI-based test analysis, mainframe application	
	Segue	SilkTest	DB access and validation, 4test script language, multiple platform/browser, web/java/client/server applications
		SilkPerformer	load and performance testing
SilkPilot		test middle-tier server, record and playback, CORBA/Java/EJB server	
SilkRadar		defect tracking	
Econfidence Scale		Scalability testing (consulting service)	
Econfidence Program	Functional/performance testing (consulting)		

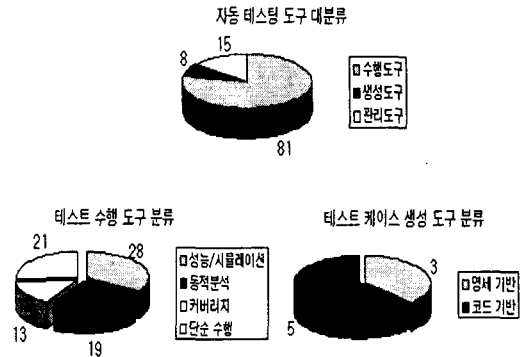


그림 2. 자동 테스트 도구의 분류
Fig. 2 Taxonomy of Automatic Testing Tools

표 2. 주요 자동 테스트 도구
Table. 2 Major Automatic Testing Tools

분야	소분류	기능 및 주요 도구
테스트 수행 도구	성능 및 시뮬레이션	<ul style="list-style-type: none"> simulate virtual user LoadRunner, TestStudio
	동적분석 (Capture and Playback)	<ul style="list-style-type: none"> intrusive/non-intrusive/web-type SilkTest, Robot, QARun, WinRunner, etc.
	커버리지 분석	<ul style="list-style-type: none"> language specific Panorama, McCabe, Insure++
	메모리 테스트	<ul style="list-style-type: none"> detect memory defects BoundChecker, Purify, etc.
	클라이언트/서버	<ul style="list-style-type: none"> provide centralized control LoadRunner, QARun
테스트 케이스 생성 도구	명세 기반 테스트 생성	<ul style="list-style-type: none"> requires formal specification QualityArchitect(Rational)
	코드 기반 테스트 생성	<ul style="list-style-type: none"> test statements and decisions C++Test, JTest (Parasoft)
	ANSI/IEEE template 지원	<ul style="list-style-type: none"> use IEEE 829 as templates in test repository
	테스트 데이터 생성	<ul style="list-style-type: none"> permutation of specified input Datamacs(CA), Datatext
테스트관리 도구	테스트 관리	<ul style="list-style-type: none"> record and track the defects ClearQuest, DevTrack, etc.

〈표 2〉는 자동 테스트 도구의 분류기준에 따라 분석한 대표적인 자동화 도구와 기능을 나타낸다.

〈그림 2〉에서와 같이 기존 테스트 도구의 약 80%는 테스트 수행의 자동화를 지원하고 있으며, 약 15%는 테스트 관리의 자동화를 지원하고 있다. 총 104개의 자동 테스트 도구 중에서 단지 8개만이 테스트 케이스의 자동 생성을 지원하고 있다.

따라서 지금까지 테스트 과정의 자동화는 주로 테스트 수행 단계 위주로 진행되었음을 의미한다[3].

일반적으로 테스트 케이스의 생성과정은 기술 집약적인 특징을 소유하며, 테스트 수행과정은 노동 집약적인 특징을 가지고 있으므로 지금까지 테스트 도구의 자동화는 테스트

수행과정 중심으로 진행되어 왔다[2][5]. 그러나 소프트웨어 개발자나 테스트 엔지니어가 직접 테스트 케이스를 생성하는 데는 많은 어려움이 따르기 때문에 테스트 케이스의 자동 생성에 대한 좀 더 많은 연구가 필요하다[3].

III. 소프트웨어 자동 테스트 도구의 발전 로드맵

제 2장에서 수행한 기존 소프트웨어 자동 테스트 도구들의 특성분석 결과를 바탕으로 테스트 도구들의 발전 로드맵을 추정함으로써 향후 자동 테스트 도구들의 발전 추세를 예측할 수 있다.

기존 테스트 도구들의 자동화 단계를 평가하기 위해 다음과 같은 세 가지 평가요소를 이용하였다.

- 지원 플랫폼
- 도구의 자동화 정도
- 소프트웨어 품질의 테스트 종류

1. 지원 플랫폼

기존 자동 테스트 도구들은 주로 개인용 PC, 일반 서버 및 메인 프레임에 개별적으로 설치, 운용되고 있다. 그러나 최근 웹기반 소프트웨어, 클라이언트-서버 소프트웨어 및 내장형 소프트웨어 등의 테스트를 위해 테스트 도구들이 이질적인 플랫폼(heterogeneous platform)에 설치되고 있다. 또한 개인용 PC나 일반 서버는 Java 플랫폼, .NET 플랫폼 등의 이용 확산에 따라 새로운 테스트 환경을 지원할 수 있도록 개선(upgrade)되고 있다[3]. 따라서 소프트웨어의 개발 및 운용 환경의 변화가 자동 테스트 도구의 발전에 많은 영향을 미치게 될 것이다.

2. 도구의 자동화 정도

소프트웨어 자동 테스트 도구들의 평가 자료에 의하면 기존 테스트 도구들은 테스트 수행과 테스트 결과분석 과정을 자동화했을 뿐 테스트 준비 과정(테스트 케이스 설계, 테스트 데이터의 생성 및 테스트 드라이버 구현 등)은 지원하지 못하고 있다[3]. 또한 현재 대부분의 자동 테스트 도

구들이 사용하고 있는 녹화 후 테스트(capture and replay) 기법은 녹화되는 테스트 케이스의 유연성과 재사용성에 따라서 제 1세대 도구, 제 2세대 도구 및 제 3세대 도구로 구분되고 있다[5]. 따라서 도구의 자동화 정도에 따라서 테스트 도구의 발전 추세를 예측할 수 있다.

3. 소프트웨어 품질의 테스트 종류

테스팅 종류는 테스트하고자 하는 소프트웨어의 품질평가 요소에 따라 기능, 성능, 이식성(portability), 신뢰성(reliability), 가용성(availability) 및 확장성 테스트 등으로 분류되고 있다[6]. 지금까지는 주로 기능 테스트와 성능 테스트 위주로 테스트 자동화가 진행되고 있으나, 최근 소프트웨어의 활용분야가 확대됨에 따라 신뢰성 및 이식성 테스트 등에 대한 요구가 증가되고 있다. 따라서 향후 자동 테스트 도구들은 소프트웨어의 신뢰성, 이식성 및 가용성 등을 테스트할 수 있을 정도로 확장되어야 한다.

기존 테스트 도구들의 자동화 단계를 지원 플랫폼, 도구의 자동화 정도 및 소프트웨어 품질의 테스트 종류 중심으로 평가하고, 평가결과를 바탕으로 향후 자동 테스트 도구들의 발전 로드맵을 (그림 3)과 같이 설정하였다. (그림 3)에서 설정한 3개의 도구발전 축으로부터 2개의 축을 조합, 선택하여 x, y축을 구성하고 (그림 4)에서와 같은 대표적인 상용 테스트 도구들을 좌표 상에 표시하면 현재 이용되고 있는 자동 테스트 도구들의 특징을 분석할 수 있다.

(그림 5)는 지원 플랫폼과 도구의 자동화 정도를 축으로 하는 테스트 도구들의 분석 결과를 나타내며, (그림 6)은 도구의 자동화 정도와 테스트 종류를 축으로 하는 테스트 도구들의 분석 결과를 나타낸다. (그림 7)은 지원 플랫폼과 테스트 종류를 축으로 하는 테스트 도구들의 분석 결과를 나타낸다.

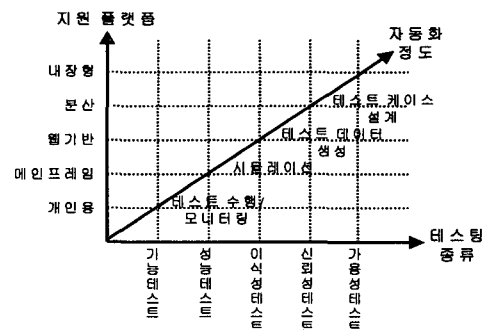


그림 3. 자동 테스트 도구의 발전 로드맵
Fig. 3 Development Roadmap of Automatic Testing Tools

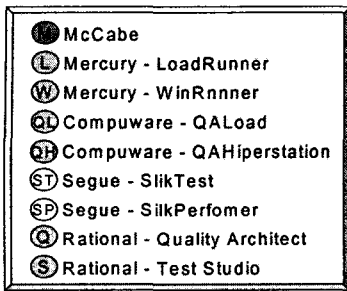


그림 4. 대표적인 상용 자동 테스트 도구
Fig. 4 Major Commercial Automatic Testing Tools

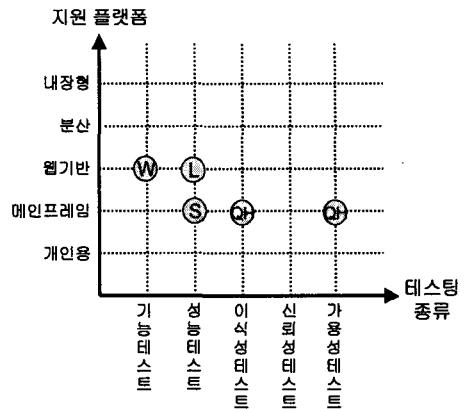


그림 7. 테스트 도구의 분석(지원 플랫폼-테스팅 종류)
Fig. 7 Analysis of Testing Tools (Supporting Platform-Testing Step)

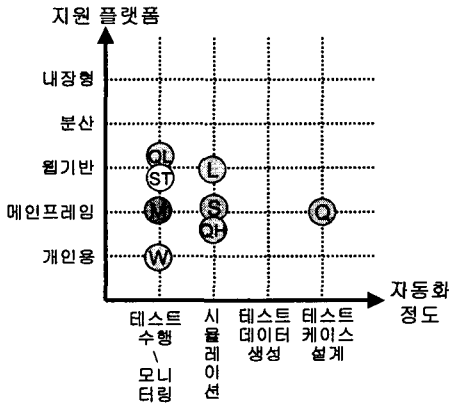


그림 5. 테스트 도구의 분석(지원 플랫폼-자동화 정도)
Fig. 5 Analysis of Testing Tools (Supporting Platform-Automatic Level)

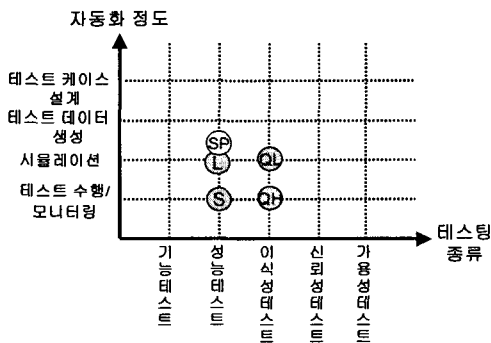


그림 6. 테스트 도구의 분석(자동화 정도-테스팅 종류)
Fig. 6 Analysis of Testing Tools (Automatic Level-Testing Step)

IV. 결론

소프트웨어의 품질보증에 대한 관심이 높아짐에 따라 향후 소프트웨어의 테스트에 자동화된 도구의 도입은 더욱 증가할 것이다. 소프트웨어의 개발과 개발된 소프트웨어의 운영 환경이 변화함에 따라 소프트웨어에서 발견되는 오류의 원인과 증상(symptom)도 다양화되고 있다. 따라서 자동 테스트 도구들의 개발도 이러한 변화에 신속히 대처할 수 있어야 할 것이다.

본 논문에서는 기존 소프트웨어 자동 테스트 도구들에 대한 특징을 광범위하게 조사, 분석하였으며, 이를 바탕으로 향후 소프트웨어 자동 테스트 도구들의 발전 로드맵을 제시하였다. 이를 통해 새로운 자동 테스트 도구들의 발전방향을 파악하고, 발전방향에 적합한 도구 개발을 촉진시키고자 하였다. 본 논문에서 제시한 테스트 도구들의 발전 로드맵은 기존 소프트웨어 테스트 도구들과 상호 보완 관계를 가지므로 현재 사용하고 있는 테스트 도구의 평가와 새로운 테스트 도구의 도입시 활용될 수 있다.

현재 상용화된 테스트 도구들은 회사에 따라 상이한 프레임워크 상에서 구축되고 있기 때문에 본 논문에서 제시한 도구 발전 로드맵 상에 존재하는 다양한 테스트 도구들의 통합, 활용은 매우 어렵다. 따라서 테스트 도구들의 통합 프레임워크 구축에 대한 좀 더 많은 연구가 필요하다.

참고문헌

- [1] E. Dustin, J. Rashka, and J. Paul, *Automated Software Testing*, Addison Wesley, 1999
- [2] M. Fewster, D. Graham, *Software Testing Automation: Effective use of test execution tools*, ACM Press, Addison Wesley, 1999E. Dustin, J. Rashka, and J. Paul, *Automated Software Testing*, Addison Wesley, 1999
- [3] OVUM, *Software Testing Tools*, OVUM, 1999
- [4] B. Beizer, *Software Testing Techniques*, 2nd Ed., International Thomson Computer Press, 1990
- [5] L. Hayes, *The Automated Testing Handbook*, 1995.
- [6] ISO/IEC, "ISO/IEC 9126:2001(E)", 2001.



저자소개

정 창 신

1983년 2월 홍익대학교 전자
계산학과 졸업
1987년 8월 홍익대학교 전자
계산학과 석사
1984년 3월 ~ 1992년 2월
ETRI 연구원
1992년 3월 ~ 2000년 7월
ETRI 선임연구원
2000년 8월 ~ 2000년 11월
ETRI 책임연구원
2000년 12월 ~ 2001년 11월
ETRI S/W품질인증팀장
2001년 12월 ~ 현재
TTA IT시험연구소 S/W시험
인증센터 평가2팀장/책임연구원
<관심분야> S/W제품품질평가,
S/W프로세스개선, 통신S/W,
DBMS기술



정 순 기

1982년 8월 Uni. of Dortmund,
Informatik, Dipl. Inform 취득
1994년 2월 Uni. of Groningen,
Computing Science, Dr. 취득
1985년 5월 ~ 현재
충북대학교 컴퓨터공학과 교수
1994년 8월 충북대학교
전자계산소장
1998년 11월 한국과학재단
한독기초과학협력위원회
정보분과위원장
2000년 4월 충북대학교 도서관장
<관심분야> 데이터베이스 시스템,
소프트웨어공학, 소프트 실시간
시스템