

소프트웨어 품질 향상을 위한 테스트
자동화 도구에 관한 동향분석
Review of S/W Test Automation Tools for the
Improvement of the Software Quality

정승환* 이금석** 김종기***
Jeong, Seung-Hwan Lee, Keum-Suk Kim, Gi-Jong

ABSTRACT

Since the change of development methods and environment of the software is rapidly kept going in industry, previous manual test methods will not be able to guarantee the quality of software. Therefore the software test automation tools is being developed actively. And many of the tools which is reliable is applied and used in domestic organizations. Indirect test techniques currently are used For the improvement of the software quality, such as formal specification and design, structural programming. And also the various of direct test techniques are used to the measurement of software quality through the fault detection and modification, such as black box, white box and test automation tools. In this survey, the software test automation tools which predicate and access the software quality at the software development phase for the improvement of the software quality should be introduced

* 정회원, 동국대학교 컴퓨터공학 대학원, 석사과정
** 비회원, 동국대학교 컴퓨터공학 대학원 교수
*** 정회원, 한국철도기술연구원 신호통신 책임연구원

1. 서론

소프트웨어의 개발 방법 및 환경이 복잡해짐에 따라 기존의 프로그래머가 하던 수동의 테스트 작업으로는 더 이상 소프트웨어의 품질 테스트에 대한 신뢰성을 보장할 수 없게 되었다. 특히 Web 기반 및 분산/병렬 환경에서의 시스템 개발은 네트워크의 복잡성과 소프트웨어 및 장비 구성 등의 다양한 요건들 때문에 많은 위험요소와 공정 지연 요인이 산재하고 있다. 단기간에 적은 인력과 장비로 비순차성과 비결정성을 갖는 시스템을 충분하고 정확하게 테스트하여 소프트웨어의 품질을 보증하기 위해 소프트웨어 특성에 맞는 전문 테스트 도구의 검토 및 도입이 필요하게 되었다. 이 논문에서는 현재 우리나라에서 이와 관련되어 사용하고 있는 테스트 자동화 도구의 종류 및 특징에 대해서 소개하고 앞으로 테스트 자동화 도구에 대해 진행되어야 할 부분에 대해서 전망해보았다.

소프트웨어를 테스트하기 위해서는 그에 따른 품질이 정량적으로 측정 가능해야 한다. 여기에서 말하는 소프트웨어 품질은 주어진 요구사항을 만족시킬 수 있는 소프트웨어의 기능 및 특성이라고 IEEE에서 정의하고 있고 ISO 9126에서는 명시적이거나 묵시적인 필요를 만족시키는 능력과 관련된 소프트웨어 제품의 특성 및 특징이라고 정의하고 있다. 소프트웨어의 품질을 향상시키기 위해 여러 가지 측정기술이 필요하다. 현재 프로그램의 품질 중에 신뢰성이나 이식성, 유지보수성 등에 대하여 객관적인 수치를 측정하기 위해 매트릭스를 사용하고 있다. 1976년에 Thomas McCabe가 Cyclomatic Complexity(모듈내복잡도)와 소프트웨어 품질의 객관적인 관점을 제시하는 Essential Complexity 등의 매트릭스 요소들에 대해 소개하고 검증하였다. 이들 매트릭스 요소는 모듈의 크기와 구조를 결정하는 하나의 단위 모듈을 통해서 독립적인 경로의 수를 계산하여 소프트웨어의 품질을 측정한다. 이후의 많은 연구들에서도 Cyclomatic Complexity는 코드에서 발견된 결함의 수와 밀접하게 연관이 되어있다는 것이 증명되었다. 이는 모듈내부의 복잡도가 낮으면 코드를 이해, 테스트하기 쉽고 따라서 에러 발생률이 낮다는 것을 설명하고 있다. 이렇게 소프트웨어 내부의 복잡도를 측정함으로써, 시스템 내에 잠재되어있는 결함들이 있는 곳을 발견하여 소프트웨어의 품질을 정확하게 측정할 수 있게 된다. 일반적으로 수작업으로는 소프트웨어 품질에 대해 거의 모든 경우의 수를 검사하기에 너무 많은 자원과 시간이 필요하게 된다. 그러나 테스트 자동화 도구에 위에서 설명한 매트릭스 기법을 적용하게 되면 테스트를 집중해서 해야 할 부분을 알 수 있게 되기 때문에 테스트에 들어가는 시간과 자원을 절약 할 수 있게 된다. McCabe사의 McCabe 테스트 자동화도구에서 테스트하는 소프트웨어의 결함을 쉽게 발견할 수 있도록 모듈내 복잡도 측정 매트릭스를 적용하고 있다.

고품질의 소프트웨어 생산하기 위한 여러 가지 도구와 방법들이 사용되고 있다. UML 도구와 정형적 명세 및 설계, 구조적 프로그래밍 등의 다양한 방법들을 소프트웨어 개발 시 사용하고 있다. 그리고 이렇게 개발된 소프트웨어를 테스트하여 품질을 향상시키기 위해 블랙박스 테스트, 화이트박스 테스트, 테스트 자동화 도구 등의 여러 테스트 기법들을 사용하고 있다. 이 논문에서는 이러한 소프트웨어의 품질향상을 위한 다양한 기법 및 도구들 중에서 국내에서 널리 사용되고 있는 소프트웨어 테스트 자동화 도구들의 상용화된 제품종류, 개발사례 및 최근동향에 대해서 소개하고자 한다.

2. 본 론

2.1. 테스트 자동화 도구 도입의 필요성

소프트웨어의 테스트에 있어서 테스트 자동화를 채택함으로써 수작업에 의해서 시간과 비용이 많이 소요되던 부분을 대체할 수 있다는 기술적 측면과 테스트의 환경적인 측면에서 자동화의 필요성이 발생하게 되었다. 90년대 이후 소프트웨어 테스트 자동화도구의 필요성은 다양한 환경의 변화에 따라 더 이상 인위적으로 테스트를 수행할 수 없게 됨에 따라 대두되게 되었다. 과거의 소프트웨어는 순차적 프로그램 기법을 따라 개발되었는데 현재는 분산, 병렬 프로그램의 개발이 보편화 되어 있고, 임베디드 소프트웨어, 유비쿼터스 환경의 소프트웨어, 클러스터 소프트웨어, 컴포넌트 기반 소프트웨어 등의 다양한 종류의 소프트웨어가 개발되고 있다. 이처럼 테스트 대상이 되는 소프트웨어가 다양하게 변하게 되어 그에 맞는 여러 다양한 테스트 방법 및 테스트 도구들이 개발되어야 한다. 또한 웹의 확산 및 네트워크의 발달로 인해 기존의 기능 및 성능 위주의 테스트에서 한 단계 나아가 최근에는 동시에 여러 사용자들의 요청을 지원하기 위한 병행성, 일관성 및 고성능 테스트에 대한 연구 또한 비중있게 다루지고 있다.

이와 같이 기술적인 측면과 환경적인 측면에서의 변화에 따라 소프트웨어 테스트 역시 다양한 플랫폼에서 다양한 품질 특성을 지원할 수 있도록 변화해야 한다. 이런 변화 추세는 테스트 기술, 테스트 환경, 테스트 관리 측면에서 기존에 사용했던 수동적인 테스트 도구 및 기법에서 감당할 수 없게 되었다. 현재 개발된 다양한 테스트 자동화 도구들은 테스트 준비, 수행, 분석 과정의 일부 혹은 전부를 자동화할 뿐만 아니라 분산 환경, 웹 환경, 내장형 환경을 지원함으로써 새로운 IT 환경에 맞는 테스트 요구사항을 지원하고 있다. 특히, 기존의 테스트 도구들은 주로 단일 컴퓨터 환경에서 동작하는 순차 프로그램에 국한된 경우가 많았으나 최근에는 테스트 생성 과정에 대한 자동화 도구들이 개발되고 있으며 분산 환경이나 클라이언트-서버 환경, 웹 기반 환경, 내장형 시스템에 대한 지원이 차츰 가시화하고 있다.

2.2. 테스트 자동화 도구의 도입효과

테스트 도구를 이용한 소프트웨어 품질관리 자동화는 소프트웨어의 결함을 조기에 발견하고 수정하는 테스트 작업의 생산성 향상과 비용 절감으로 소프트웨어 품질관리 비용의 투자대비 효과를 2가지 측면에서 극대화 시킬 수 있다.

1)정성적 효과

- 어플리케이션 개발공정 중 40% 이상을 차지하는 테스트 공정의 기간을 단축 (3배 이상 가능)
- 테스트 기간의 단축 및 비용 절감으로 충분한 종합 테스트를 반복 시행함으로써 오류 수정 및 성능 검증을 사전에 실시 가능
- 테스트 과정 및 결과물의 자동화를 통해 테스트의 정확성과 생산성을 향상
- 개발자의 수작업이 감소하므로 개발자의 생산성이 증대
- 향후 운영 단계에서 시스템의 변경(H/W,S/W VERSION UPGRADE등)이나 프로그램의 수정이 발생하였을 경우 정확한 테스트를 사전에 실시하여 오류 발생을 최대한 감소
- 단기간에 최소의 인원과 장비로 충분한 사전 모의 테스트를 할 수 있기 때문에 전체적인 소프트웨어의 품질을 향상

- 소프트웨어 품질 향상으로 고객 만족도 향상 및 대외 경쟁력 극대화
- 테스트 신기술을 도입함으로써 소프트웨어 품질 보증 분야의 경쟁우위를 확보

2)정량적인 효과

- 대부분의 테스트 부서와 QA(Quality Assurance) 부서는 개발과 유지보수 인원에 비해 절대적으로 인원이 부족해서 기한을 지키기 어려운 문제가 있었으나 테스트 자동화가 이런 병목 현상을 해소 시켜 줄 수 있다.

2.3. 국내에서 사용하는 S/W 테스트 도구의 종류

현재 세계적으로 기업에서 사용되고 있는 소프트웨어 테스트 자동화 도구는 100개가 넘는다. 이 도구들은 각각 지원 기능, 지원 환경 등에 따라서 서로 다른 특징들을 가지고 있다. 따라서 이 도구들의 주요 특징을 파악하는 것은 향후 소프트웨어 시험 자동화 과정에서 중요한 단서가 될 것이다.

아래의 표는 국내에서 상용화된 소프트웨어 시험 도구들에 대해 간략하게 요약한 내용이다.

<도표 1. 국내에서 사용되고 있는 소프트웨어 테스트 자동화도구의 종류 및 특징>

회사명 / 제품군	제품명	주요기능 및 특징	분야
SureSoft / CodeScroll	Concurrency Tester	- 소스코드 기반 테스트 도구로서 별도의 명세가 없더라도 테스트를 수행할 수 있음. - 테스트의 모든 과정을 자동화하여 테스트의 도움 없이 수행가능	코드기반테스트생성
McCabe & Associate	McCabe QA	- metrics를 통해 위험요소를 추적	커버리지 분석
McCabe & Associate	McCabe Test	- 구조화된 단위테스트 기법제공. - 테스트 작업에 대한 품질을 보증해주는 동적 분석기법제공	커버리지 분석
Rational / Function Testing	Robot	- Capture & Playback에 의한 기능 시험도구	Capture & Playback
Rational / Performance Testing	Test Studio	- 가상 유저를 생성하여 시스템의 성능/부하테스트	성능 및 로드
Segue	Silk Test	- Capture & Playback에 의한 기능 시험도구	Capture & Playback
TeleLogic	Tau Logiscope	- static testing, coverage analysis.	커버리지 분석
Mercury Interactive / Monitoring	Topaz	- 성능 모니터링 도구. - 기능 테스트 및 성능 테스트를 거쳐 시스템이 배포된 이후 실제 운영환경에서의 시스템 성능 모니터링	테스트관리
Mercury Interactive / Testing	LoadRunner	- 가상 유저를 생성하여 시스템 성능/부하 테스트	성능 및 로드
Mercury Interactive	TestDirector	- 웹 기반 테스트 케이스 관리도구	테스트관리

/ Testing			
Mercury Interactive / Testing	WinRunner	- 윈도우 어플리케이션의 기능테스트 자동화 도구	클라이언트/서버
Compuware	QARun	- 윈도우 어플리케이션의 기능테스트 자동화 도구	클라이언트/서버
Compuware	QALoad	- 가상 유저를 생성하여 시스템 성능/부하 테스트	성능 및 로드
Compuware	Application Expert	- 시스템 성능 모니터링	테스트관리
Compuware	QACenter	- Capture and Replay에 의한 기능 시험	Capture & Playback
Compuware	QADirector	- 자동화된 테스트 프로세스 관리도구	테스트관리
Parasoft	Jtest	- 소스 코드 분석	코드기반테스트생성

3. 결 론

이전의 소프트웨어는 그 규모가 작고 각 모듈의 수행순서가 순차적인 프로그램들이 대부분이었으나 지금은 코드의 라인 수 면에서 수천만 줄에 이르는 매머드 급 소프트웨어(e.g. window series)가 개발되어있고 지금도 그 이상의 크기의 소프트웨어가 개발되고 있다. 지금의 소프트웨어는 그 수행 순서가 순차적이기보다는 한번에 수천이상의 사용자의 요구를 처리해야하는 병행적이면서 비결정적인 성격을 띠고 있다. 여기에는 분산/병렬 소프트웨어, 웹 기반의 소프트웨어, 내장형 소프트웨어, 유비쿼터스 기반의 소프트웨어 등 그 종류가 다양한 소프트웨어의 구조상의 차이에 맞게 테스트하는 방법 및 절차 또한 달라야한다. 이 논문에서는 소프트웨어 테스트 자동화 도구의 필요성 및 장점과 테스트 대상이 되는 소프트웨어의 종류에 따라 적용되어야하는 테스트 자동화 도구에 대해 현재 국내에서 사용되고 있는 동향 및 특징에 대해 소개하였다. 그리고 현재 국내에 사용되고 있는 테스트 툴의 도입현황에 대한 객관적인 자료가 마련되어 있지 않고 대부분의 자동화 도구가 고가인 점에 비해 활용적인 부분은 아직까지 많이 미흡한 실정이다. 앞으로 테스트 도구의 수행성 측면에 있어서 활용방안에 대한 연구가 활발히 진행되어야 할 것이다.

참고문헌

1. E. Ferneley, "Design Metrics as an Aid to Software Maintenance: An Empirical Study", *J. Softw. Maint: Res. Pract*11, 55-72, 1999
2. 최은만(2001), "소프트웨어공학론", 사이텍미디어.
3. 제니시스기술(주)(2003), "품질 향상을 위한 머큐리인터랙티브 시험 솔루션", 2003
4. 배현섭(2003), "소프트웨어 시험 자동화 환경 구축", 2003 소프트웨어공학연구회 튜토리얼
5. SureSoft Technologies Inc.(2002), "CodeScroll : API Tester", White Paper
6. CodeScroll Concurrency Tester, <http://www.suresofttech.com/>
7. Telelogic Tau, <http://www.telelogic.com/>
8. Compuware <http://www.compuware.co.kr/>
9. Mercury Interactive <http://www.mercury.co.kr/>
10. G-T. Daich, G Price, B Ragland, M Dawood(1994), "Software Test Technology Report", US Air force
11. K-J. Ross(1998), "Practical Guide to Software System Testing", 2 days Tutorial