소프트웨어 개발 품질 향상을 위한 회귀테스트 자동화 도구 설계

서강복, 이우진
경북대학교 IT대학 컴퓨터학부
e-mail:seokang13@naver.com, woojin@knu.ac.kr

Design of Automated Regression Testing Tool for Improvement of Software Development Quality

Kang-Bok Seo, Woo-Jin Lee
School of Computer Science and Egineering, Kyungpook National University

요 약

최근 소프트웨어가 사회에서 하는 일이 점점 더 증가함에 따라 소프트웨어의 개발비용도 함께 증가하고 있다. 소프트웨어의 개발비용 중 에러를 찾아내는 소프트웨어 테스팅이 상당한 부분을 차지하고 있는데 테스팅에 소요되는 비용 때문에 테스팅을 등한시 하는 경우가 많다. 그럼에도 불구하고 소프트웨어의 기능 추가나 변경이 이루어질 때마다 테스트를 진행하여야 소프트웨어의 에러를 최소화할 수 있다. 하지만 개발이 진행될수록 소요되는 비용이 증가하는 경우가 많아 실제 소프트웨어 개발에선 개발 기한이 다가올수록 회귀 테스트를 기피하게 된다. 본 논문에서는 이러한 문제를 해결하기 위해 소프트웨어의 개발을 진행하면서 형상관리를 통해 소프트웨어의 변경이 있을 때마다 회귀 테스트를 자동으로 수행해주는 도구를 제안한다.

1. 서 론

일상생활에서 소프트웨어의 중요성이 점점 늘어 가고 소프트웨어의 영향력이 높아지고 있다. 그에 따라 소프트웨어의 개발비용 또한 증가하고 있다. 소프트웨어의 개발비용 중에서 상당한 부분을 차지하는 요소는 소프트웨어에 있는 에러를 찾아내는 작업인 소프트웨어 테스팅[1]인데 소프트웨어 테스팅을 하지 않았을 시 소프트웨어에 대한 신뢰성이나 품질을 기대하기 어렵다.

소프트웨어 테스팅 기법 중 하나인 회귀 테스트는 소프트웨어에 수정이 가해졌을 때 이전에 실행하였던 테스트를 다시 수행하는 것이다. 소프트웨어는 개발 도중에 수많은 수정이 일어나고 여러 개발자가 같이 개발할 때 더욱 많은 수정이 일어날 수 있다. 하지만 이러한 소프트웨어의 수정 도중 의도치 않게 시스템 내에 버그를 유발하는 경우가 종종 일어나는데 회귀 테스트를 주기적으로 수행하면 이 버그를 효과적으로 찾아낼 수 있다. 하지만 회귀 테스트의 경우 개발이 진행될수록 더욱더 많은 비용이 소요되고 이러한 부담은 프로젝트 종료 시점이 다가올수록 회귀 테스트를 기피하게 되는 큰 요인이 된다. 실제 우리나라 소프트웨어 산업에서는 회귀 테스트뿐만 아니라 여러 소프트웨어 테스팅에 관한 인식이나 시행이 많이 등한시 되고 있다[2][3]. 본 논문에서는 이러한 문제점을 해결하기 위해 프로젝트를 진행하면서 회귀 테스트를 자동으로 수행할 수 있는 회귀 테스트 자동화 도구를 제안한다. 이 도구는 개발자가 소프트웨어를 개발하면서 형상관리 서버에 소스코드를 업로드 하면 변경된 소스코드를 자동으로 테스트를 진행하고 테스트의 결과를 보고서 형태로 작성하여 개발자에게 제공하여 소스코드에 있는 에러 유무를 확인할 수 있다. 결과적으로 개발자가 수정한 소스코드는 회귀테스트를 통해 신뢰성이 증가하고 개발자는 회귀테스트에 드는 비용을 온전히 개발에 사용할 수 있어 소프트웨어의 품질을 향상시킬 수 있다.

2. 관련 연구

2.1 회귀 테스트 기법

회귀 테스트는 소스코드가 수정되었을 때 시스템 내에 의도치 않게 포함된 버그를 찾아낼 수 있도록 이전에 수행했던 모든 테스트를 수행하는 기법인데 이 테스트는 많은 비용이 필요하다. 그래서 회귀 테스트에 관한 연구는 프레임워크에 따른 테스팅 도구의 개발도 연구가 되고 있지만 회귀 테스트에 드는 비용을 감소시키는 연구도 많이 진행되고 있다. 회귀 테스트 기법은 크게 4가지로 나눠지는데 지금까지 했던 모든 테스트를 다시 수행하는 Retest All, 테스트를 선택적으로 수행하는 Regression Test Selection, 테스트 케이스에 우선순위를 매겨 수행하는 Test Case Prioritization, Regression Test Selection과 Test Case Prioritization을 적절히 섞어서 수행하는 Hybrid Approach가 있다[4]. 이 논문의 도구에서는 모든

테스트를 다시 수행하는 Retest All을 이용하여 회귀 테스트를 수행한다.

2.2 회귀 테스트 자동화

최근 많은 소프트웨어가 GUI환경으로 개발되어 회귀 테스트 자동화도구 또한 GUI를 대상으로 개발 및 연구가 많이 진행되고 있으며 우리나라도 네이버에서 개발한 GUITAR(GUI Testing Automation fRamework)[5]라는 GUI자동 테스트 도구가 있다. GUI에서 수행하는 테스트는 수행하는 시나리오를 저장하여 저장되어 있는 시나리오대로 수행하여 테스트 자동화를 진행한다. GUI이외에도 회귀 테스트를 지원하는 테스트 도구들이 있는데 수행방식은 테스트케이스를 엑셀이나 다른 파일들로 저장하고 테스트를 수행하면 저장되어 있는 모든 테스트케이스를 수행하는 것으로 회귀 테스트를 진행한다.

2.3 Maveryx

Maveryx[6]는 이클립스 플러그인 형태로 지원되는 테스팅 도구이다. 자바와 안드로이드를 대상으로 한 소스코드와 GUI를 모두 지원하는 테스팅 도구인데 엑셀이나 XML로 테스트 데이터만을 입력하면 자동으로 테스팅을 수행할 수 있다. 이 도구의 회귀 테스트는 XML이나 엑셀 형태로 입력 하여 두었던 이전의 테스트 데이터들을 모아서 한번에 진행하는 형태로 회귀 테스트를 수행할 수 있다. 하지만 이 도구도 개발자가 직접 수행하여야 한다는 점에서 앞에서 지적한 개발 기한이 촉박할 때는 개발자가 수행하지 않을 수 있다는 문제점이 남아 있다.

2.4 Visual Studio

비주얼 스튜디오 에서도 MTM(Microsoft Test Manager)을 통해 테스팅을 지원하고 있다. MTM은 비주얼 스튜디오에서 지원하는 여러 가지 개발언어에 대하여 테스트를 지원하는데 회귀 테스트 또한 지원하고 있다. MTM을 이용한 회귀 테스트는 개발자가 테스트 케이스를 엑셀이나 비주얼 스튜디오에서 지원하는 테스트 케이스 형태로 작성하여 저장되어 있는 테스트 케이스들을 한번에 수행하는 형식으로 회귀 테스트가 진행이 된다. MTM은 회귀 테스트를 수행하는 시점이 소스코드가 수정된 시점이 아닌 시간을 기준으로만 회귀테스트를 자동으로 수행할 수 있다는 단점이 있다.

3. 회귀 테스트 자동화 도구의 구조

회귀 테스트 자동화 도구의 구조는 크게 소스코드를 개발할 수 있는 통합개발환경과 소스코드를 저장, 회귀 테스트를 진행하는 서버로 나눌 수 있다. 이중에 서버는 DBMS, SVN, 테스트 모듈, 테스트 결과 보고서 생성 모듈 이렇게 4개의 모듈로 구성되어 있다. IDE 모듈은 소스코드를 개발할 수 있는 환경을 제공하고, SVN은 IDE에서 업로드된 소스코드를 관리하고 테스트 모듈은 서버에 업로드되어 있는 소스코드를 가져와서 데이터베이스에 입력되어 있는 테스트 케이스를 기반으로 회귀테스트를 수행하고, DBMS는 수행할 테스트 케이스와 테스트를 수행하고 난 결과를 저장하고, 테스트 결과 보고서 모듈은 데이터베이스로부터 데이터를 가져와 회귀 테스트를 진행하고 난 뒤의 결과를 출력해준다. 시스템의 전체적인 구성도는 그림 1과 같다.

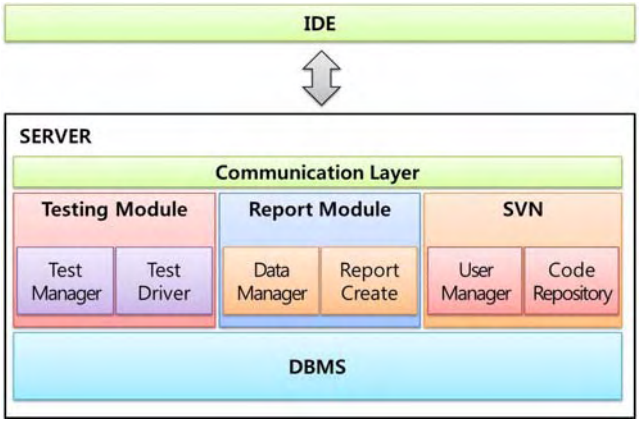

그림 1. 시스템 구성도

3.1 통합개발환경(IDE)과 SVN

개발환경은 C/C++를 기준으로 하며 이클립스 환경을 가져와 기능을 추가하는 방식으로 구현한다. SVN은 SVNkit[7]라는 오픈소스 API를 이용하여 각 기능들을 커스터마이징할 수 있는데 서버에 소스코드가 업로드 되면 데이터베이스에 저장되어 있는 테스트 케이스를 출력하여 개발자가 테스트 케이스를 수정할 수 있도록 구현한다.

3.2 테스트 모듈

테스트 모듈은 SVN에 업로드되어 있는 소스코드와 DB에 저장되어 있는 테스트 케이스를 가져와 테스트 드라이버를 이용하여 회귀 테스트를 진행할 수 있도록 한다. 또한 개발자가 기존에 데이터베이스에 저장되어 있는 일부분의 테스트 케이스도 수행하는 기능도 추가한다. 소프트웨어의 테스팅 이외에도 ISO/ANSI 표준과 MISRA-C와 같은 표준들의 대해 개발되어진 소스코드들이 권고사항들을 얼마나 충족시키고 있는지에 대한 소스코드에 대한 분석 또한 수행할 수 있도록 개발한다.

3.3 테스팅 결과 보고서

테스팅 결과 보고서생성모듈은 데이터베이스에 저장되어 있는 테스트 결과를 개발자가 쉽게 참고할 수 있도록 보고서 형태로 수행한 테스트 케이스와 테스트 횟수, 테스트 결과 등을 출력하여 개발자가 보고서를 참고하여 개발을 진행하면서 수행된 테스트의 결과를 지속적으로 개발하고 있는 소프트웨어에 적용할 수 있다.

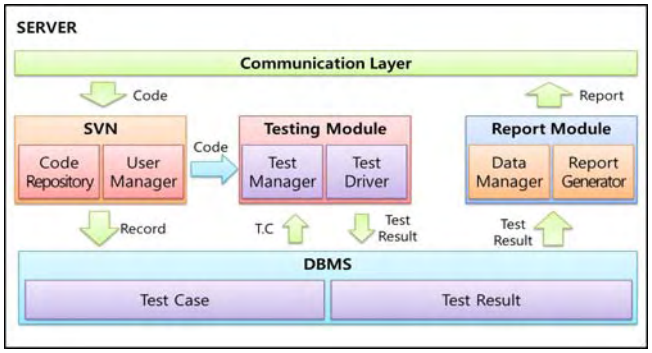

그림 2. 서버 내부 구조

그림 2는 서버에서 각 모듈들이 동작하는 구조를 나타낸 것이다. 개발자가 소스코드를 업로드하면 소스코드는 SVN에 있는 코드 저장소에 저장되고 테스팅 모듈은 SVN에 있는 소스코드와 데이터베이스에 저장되어 있는 테스트 케이스를 가져와 테스트 드라이버에서 회귀 테스트를 수행한다. 회귀 테스트가 수행되고 나면 데이터베이스에 수행 결과가 저장되고 보고서 모듈은 데이터베이스로부터 수행결과를 가져와 보고서를 작성하여 통신모듈을 통하여 개발자에게 전송된다.

3.4 시스템의 개발 적용

이 도구를 이용하여 프로젝트의 개발을 진행할 시 프로젝트가 주기를 가지고 진행될 수 있다. 개발자가 소스코드를 작성하여 SVN에 업로드 하면 테스트 모듈은 데이터베이스에 저장되어 있는 테스트 케이스와 소스코드 저장소에 있는 소스코드를 가져와 회귀 테스팅을 수행하고, 테스트를 수행한 결과를 데이터베이스에 저장한다. 그러면 테스팅 결과 보고서생성모듈에서 데이터베이스로부터 테스트 결과를 가져와 개발자가 손쉽게 참고 할 수 있는 보고서 형태로 출력하여 주고 개발자는 이 보고서를 참고하여 현재 개발 중인 소스코드에 적용할 수 있다. 이 도구의 장점은 회귀 테스팅을 자동으로 수행하므로 소스코드를 수정할 때에도 테스팅으로 인한 개발자의 부담을 줄여 주고 개발에 소요되는 비용도 줄일 수 있을 것으로 예상된다. 또한 테스팅을 수차례 수행 하였으므로 소프트웨어의 품질과 신뢰성 또한 보장 받을 있다. 그림 3은 위의 과정을 시퀀스다이어그램으로 나타낸 것이다.

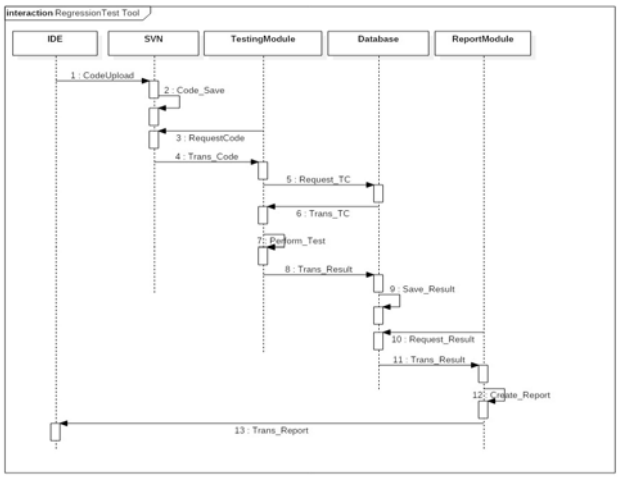

그림 3. 회귀테스트 자동화 도구 시퀀스 다이어그램

5. 결 론

본 논문은 소프트웨어 테스팅 기법 중 상대적으로 많은 비용이 소요되는 회귀 테스팅의 비용을 절감하고 자동화할 수 있는 도구를 제안하였다. 위의 도구를 이용하여 많은 비용이 소요되는 회귀 테스팅을 자동으로 수행하여 소프트웨어의 수정을 통해 추가 될 수 있는 버그를 최소화할 수 있다. 향후 연구로는 위의 시스템의 좀 더 확장하여 분산 개발환경에서 위의 시스템을 적용하여 통합적으로 프로젝트를 관리하는 시스템에 대한 연구가 필요하다.

Acknowledge

본 논문은 중소기업청에서 지원하는 2015년도 산학연협력 기술개발사업(No. C0332987)과 2014년도 정부(교육부)의 재원으로 한국과학재단의 지원을 받아 수행된 기초연구사업(No. NRF-2014R1A1A2058733)의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

[1] B. Beizer, “Software Testing Techniques”, Second Edition, Van Nostrand Reinhold Company Limited, 1990
[2] 김진형, “우리는 소프트웨어 중심사회를 준비하고 있는가?“, 소프트웨어정책연구소, 2015년 1월12일
[3] 권호열, “소프트웨어테스팅 및 IT감리 고급인력 양성에 관한 수요 조사 및 분석“, 대한전자공학회 하계학술대회 논문집, 대한전자공학회, 2015년 6월
[4] S. Kadry, "A New Proposed Technique to Improve Software Regression Testing Cost", International Journal of Security and Its Applications, Vol. 5, July, 2011
[5] GUITAR, http://dev.naver.com/projects/guitar
[6] Maveryx, http://www.maveryx.com/
[7] SVNkit, http://svnkit.com/