X-internet 환경에 사용되는 개발도구 테스팅 기술의 평가

이창섭*, 이승혁*, 조성환**, 배종순*
*주식회사 컴스퀘어
** Purdue University
e-mail : ds2shg@comsquare.co.kr

Development Tool in X-Internet Environment Evaluation on Testing Techniques

Lee Chang Sub*, Seung-Hyuk Lee*, Cho Seong Hwan**, Bae Jong Soon*
*Comsquare
**Dept of Computer Science, Purdue University

요 약

소프트웨어는 의도하지 않은 결함을 반드시 가지고 있다. 이러한 결함을 찾아내기 위해서는 테스트가 반드시 진행 되어야 한다. 수많은 소프트웨어 제품들이 나오게 되면서 테스팅 기법도 표준화 되어 가고 있다. 이러한 테스팅 기법들을 현업에 종사하는 테스터가 사용시 많은 시간과 비용이 들게 된다. 이를 좀더 효과적으로 테스트 하여 시간과 비용을 줄일 수 있는 방안을 찾고자 하는 것이 본 논문의 목적이다.

실행 가능한(eXecutable), 확장된(eXtended), X-internet 환경을 구축하는 UI(User Interface)개발 도구를 개발함에 있어 제품의 안정성을 향상 시키기 위하여 기존의 소프트웨어 테스팅 기법을 이용한 테스팅 방법론을 연구하고자 한다.

1. 서론

X-internet 이라는 용어는 2001 년 미국의 리서치 기관인 포레스터(Forrester)에서 차세대 인터넷이라고 발표하면서 사용되기 시작하였다.

기존의 웹은 인터넷을 통해 실행 가능한 응용프로그램의 개발과 배포에 적합하지 않고, 다양한 기기들을 인터넷에 연결하여 활용 할 수 없다는 한계점을 가지고 있었다. 이러한 문제점을 극복하는 방안으로 X-internet 개념이 도입 되었다. 이는 실행 가능한 인터넷(eXecutable Internet), 확장된 인터넷 (eXtended Internet) 이라고 정의 하고 있다[1]. 이처럼 X-internet 은 기존의 C/S(client/server) 환경과 같은 소프트웨어가 가졌던 풍부한 기능을 웹 상에 구현함으로써 차세대 어플리케이션을 실현 가능케 하고 있으며 사용자가 어느 디바이스에서나 구현 가능하도록 해주는 기술을 말한다[2].

웹의 변화에 따라 소프트웨어들도 다양하게 개발이 진행 되고 있으며, 표준화된 테스팅 방법을 통해 제품의 안정성을 보장하고 있다.

X-internet 환경에서 UI 를 개발할 수 있는 소프트웨어를 개발 하면서 진행된 테스팅 기법을 통해 제품의 안정성을 평가해 보고 현업에서 사용되는 테스팅 모델을 찾아보고자 한다.

본 논문에서는 소프트웨어 테스팅 기법 중 회기 테스트, 블랙박스테스트, 요구사항명세기반테스트, 데이터 테스트 기법들의 장단점을 파악해보고 제품의 안정성을 높이는 테스팅 기법을 찾아보고자 한다.

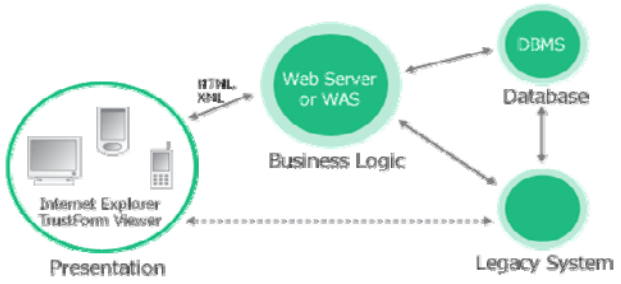
2. X-internet 에 사용되는 개발도구

X-internet 에 사용되는 개발도구들은 기존의 C/S 환경의 장점과 웹의 장점을 수용하며, 새로운 웹 개발 환경에 능동적으로 대응 할 수 있는 기능을 제공하고 있다.

본 논문에 사용된 X-internet 환경에 사용된 개발도

구는 TrustForm4Designer 라는 것이다. 이 개발도구는 기존의 C/S 화면을 쉽고 빠르게 개발 할 수 있는 XML / XForms 기반으로 구성이 되어있다.

이 개발도구는 다양한 비즈니스 UI 의 요구사항을 반영하고 있으며, Presentation Solution 으로서 UI(User Interface) 와 DATA(Business Logic 에서 처리된 Data) 를 완벽하게 분리함으로써, 운영 Performance 를 향상시키고 운용 비용을 절감하는 특징을 가지고 있다.



[그림 1] TrustForm4 개발도구 개념

3. 소프트웨어 테스팅 정의

소프트웨어 테스트를 정의한 문헌들을 보면 그 정의들이 조금씩 다르다는 것을 알 수 있다. Glen Myers 는 “테스트란 오류를 발견하기 위한 의도로 프로그램을 수행하는 처리과정이다[3].” 라고 정의 하고 있으며, Pressman 은 “소프트웨어 테스트는 소프트웨어 품질 인증의 중요한 요소이고, 명세서, 설계, 코드 생성의 최종 검토를 나타낸다[4].” 라고 각각 정의를 하고 있다. 이러한 소프트웨어 정의를 종합하면 “소프트웨어 테스트는 프로그램에 있는 오류를 찾기 위해 프로그램을 실행하여 실행 결과와 예상 결과를 비교하고 검토하는 과정이다[5].” 라고 정리를 할 수 있다.

4. 소프트웨어 테스팅 기법

회기 테스트 기법은 개발도구를 테스트함에 있어 새로 추가 되는 테스트 케이스들을 자동화 샘플을 이용하여 테스트 샘플을 작성 하고, 이렇게 누적된 테스트 케이스를 테스트 함을 말한다. 회기 테스트의 장점은 수정된 개발도구를 테스트 할 때 기존에 되던 기능이 새로 추가된 기능 및 결함을 수정한 것에 대해 오동작 하는 기능을 찾아낼 수 있다. 그러나 테스트 케이스가 늘어나면 테스트 시간도 늘어나는 특성을 가지고 있다.

블랙박스 테스트는 소프트웨어의 내부 동작은 알 수 없는 상태에서 오직 소프트웨어가 값을 입력 하면 그에 상응하는 결과가 옳은지 판단하는 형태의 테스트를 말한다[6]. 기능이나 동작에 대한 테스트 케이스를 설계하여 테스트할 때 적합하다. 그러나 정확한 요구사항 명세서와 해당기능에 대한 기술문서가 필요하게 된다.

요구사항 명세기반 테스트는 명세에서 테스트 케이스를 유도할 수 있다. 명세는 소프트웨어 제품에 대한 속성이나 예상되는 행위(Behavior)를 서술하게 된다[7]. 요구사항 명세서는 고객의 요구사항에 의해 작성되며 이를 회의를 통하거나, 요구사항 명세 프로세스를 통해 DB 화 할 수 있다. 이 방법은 실제 사용자측면을 고려하여 개발 및 테스트가 진행이 되므로 고객의 요구사항을 충족시킬 수 있다. 그러나 수많은 요구사항으로 인해 의도하지 않은 결함을 낳게 할 수 있다.

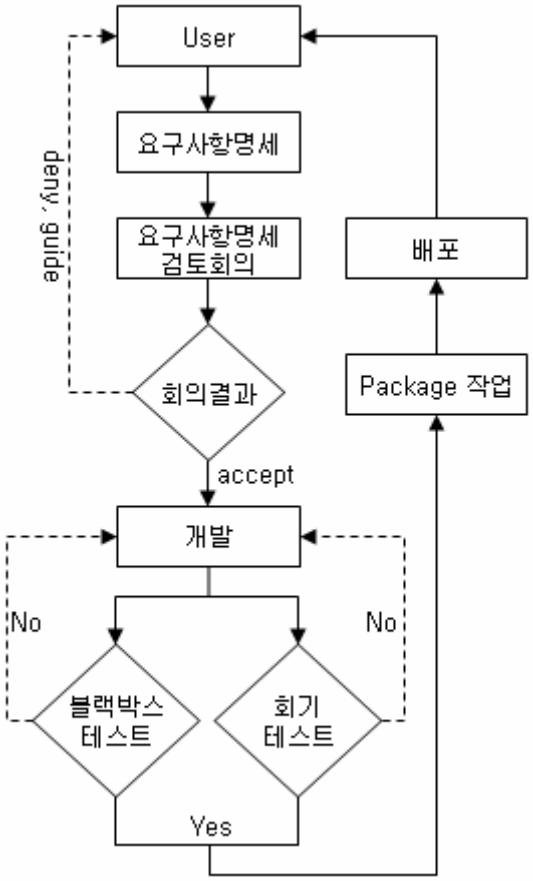
데이터 테스트 기법은 소프트웨어가 수행하는 작업 중 입력 값이 있을 때 이 값을 경계 값에 해당되는 값을 넣거나, 유효하지 않은(Invalid) 쓰레기(Garbage) 값을 입력하여 출력되는 결과를 테스트 하는 방법을 말한다. 이 방법은 예기치 않거나 잘못된 입력 값에 대한 프로그램의 견고성을 테스트 하기 위해 주로 사용 된다. 입력 값이 많게 되면 테스트 케이스가 많게 되므로 경계조건(boundary condition)과 하위 경계조건(sub-boundary condition) 등을 기초로 한 동등 분할에 의해 테스트 케이스 수를 줄이는 방법을 같이 사용해야 한다[6].

5. 테스팅 프로세스 환경 및 테스팅 절차

5-1. 테스팅 프로세스 사이트

본 논문에서는 테스트 프로세스 사이트를 구축하여 3 장에서 설명한 테스트 기법을 적용시켰다.

구성은 [그림 2] 와 같다.



[그림 2] 테스팅 프로세스 사이트환경

사용자는 [그림 3] 에서와 같이 테스팅 프로세스 사이트를 통해 기능 추가 요청 및 버그에 대한 사항

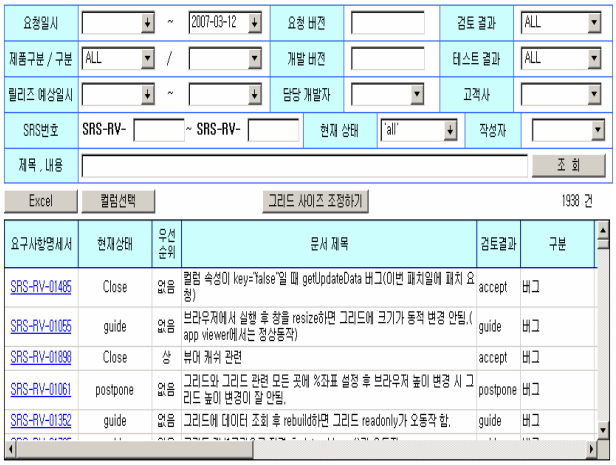
을 접수 할 수 있다. 접수된 요구사항 명세서는 검토 회의를 거쳐 개발도구에 반영 할 지의 여부를 판단하게 되는데, 이때 요구사항 명세기반 테스트 케이스에 따라 검토회의를 진행하며 회의를 통해 사용자의 요구사항이 무엇인지 정확하게 판단 후 개발 할지 의 여부를 결정하게 된다. 의미가 불명확하게 요청이 들어왔거나, 제품과 맞지 않는 스팩을 요구하게 될 경우는 요구사항 명세기반 테스트 케이스에 의해 반려 된다.

5-2. 테스팅 프로세스에 따른 테스팅 절차

요구사항 명세서기반 테스트 케이스에 의해 요구사항이 수락이 되면 실제 개발이 들어가게 된다. 이때 개발자는 요구사항 명세와 이를 검토한 회의 자료를 토대로 개발을 진행 하게 되고 개발이 완료가 되면 테스팅 프로세스 사이트에 개발완료 보고서를 작성하게 된다. 개발 완료 보고서가 작성되면 블랙박스 테스트와 회기 테스트가 진행된다.

블랙박스 테스트를 진행할 때에는 새로 개발된 제품의 요구사항 명세리스트가 무엇인지 파악 후 테스트 케이스를 설계를 하게 된다. 테스트 케이스가 설계되면 실제 테스트 샘플을 작성하고 테스터가 직접 운영자가 되어 테스트를 진행 한다. 테스트 시 확인사항은 요구사항 명세서의 내용에 적합한지와 검토회의 내용 및 개발 완료 보고서를 참조하여 기능이 정확하게 개발 되었는지 파악 한다. 블랙박스 테스트 조건에 만족하지 않으면 fail 처리가 되어 다시 개발자에게 재개발 요청을 보낸다.

회기 테스트는 새로 개발된 제품을 자동화 샘플을 이용해 테스트를 진행 한다. 새로 개발이 되거나 버그가 수정되었을 때 다른 기능이 오동작하는 것을 찾아내기 위한 테스트 방법이다. 이 부분에서도 결함이 발견 되면 개발자에게 결함보고를 하고 재개발이 진행이 된다.

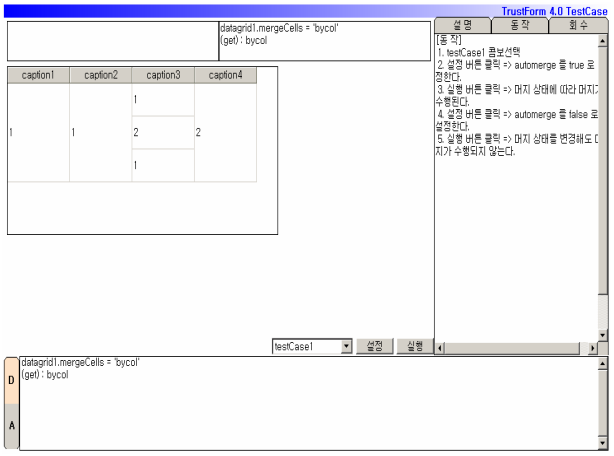


[그림 3] 테스팅 프로세스 사이트

데이터 테스트는 경계 값에 대한 리스트를 선정하고 해당되는 파라미터에 값을 연속적으로 입력 하여 출력되는 데이터를 확인하게 된다.

이 테스팅 방법은 완성된 기능에 대해 경계 값을 결정 하고 하나의 화면에서 적게는 몇 가지 많게는 수백 가지의 테스트를 하나의 샘플을 통해 진행을 할 수 있다. 데이터 테스팅 기법은 블랙박스 테스트에서 동시에 진행이 된다.

테스트 케이스에 의해 새로 작성된 샘플들은 다음 제품 버전이 나오기전 자동화 샘플을 이용해 회기 테스트가 될 수 있도록 진행 한다.



[그림 4] 데이터 테스팅 샘플

6. 테스팅 기법을 통한 결함 률 분석 및 제품안정성 평가

회기테스팅을 통한 테스트 기법은 새로 추가된 기능 및 버그를 수정하여 기존에 잘 동작하던 기능이 안 되는 현상을 감지해내는데 효과적일 것으로 예측을 하였다.

실제 회기테스팅을 수행을 하면 기존에 잘 동작하던 기능이 안 되는 현상을 찾아낼 수 있었다. 그러나 테스트 샘플이 늘어남에 따라 시간과 비용이 비례함을 알 수 있다.

요구사항 명세서 테스팅 기법은 요구사항 명세서가 접수가 되면 아래 [표 1] 요구사항 명세기반 테스트 요건[8]에 따라 검토회의가 진행이 되고 검토 결과가 [accept, deny, guide, postpone] 4 단계로 구분이 된다.

요구사항 명세 검토 회의는 일주일에 2 회에 걸쳐 진행하였고, 하루 평균 20 건을 검토회의를 통해 처리하였다. 20 건 검토회의당 평균 2 시간이 소요되었다.

[표 1] 요구사항 명세기반 테스트 요건

요구사항 명세기반 테스트 케이스	
1. 완결성	명세서 만으로 조작할 수 있는 모든 것이 포함 되었는가?
2. 정확성	목적이 적절하게 정의되어있는가?
3. 정밀함, 모호하지 않고 명백함	설명이 애매하지 않고 정확한가? 읽고 이해하기 쉬운가?
4. 일관성	각 기능이 다른 기능과 서로 위배되는 경우가 있지 않은가?
5. 연관성	해당 기능을 토대로 고객의 요

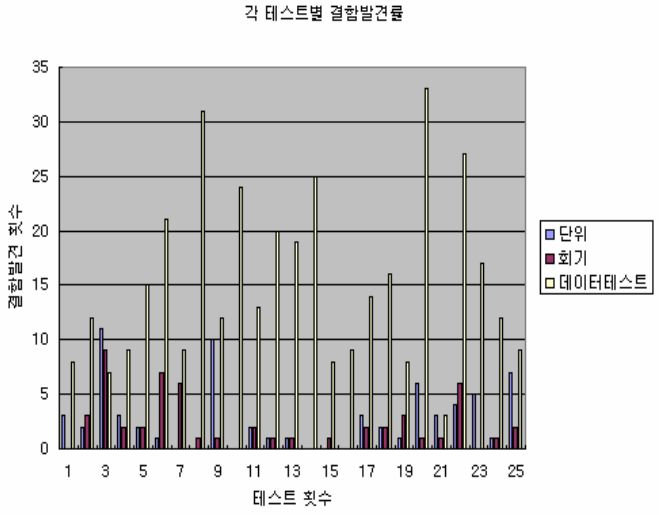
	구 사항으로 추적이 가능한가?
6. 실행 가능성	자원을 이용하여 제시된 비용과 일정을 맞추어 구현될 수 있는 기능인가?
7. 코드와 무관	명세서가 소프트웨어 설계, 아키텍쳐, 코드와 연관되지 않고 제품 정의에만 집중하고 있는가?
8. 테스트 가능성	테스트 케이스를 만들고 작동을 확인할 수 있는 충분한 정보가 제공되는가?

요구사항 명세기반 테스트 요건 중 8 가지 요건이 모두 만족하면 accept 였으며, 이중 하나의 요건이라도 만족하지 않으면 deny 처리가 되었고, 요구사항 명세서를 재 요청 하였다.

[표 2] 요구사항 검토회의 따른 처리율

구 분	accept	deny	guide	postpone
도입초기	62%	27%	8%	3%
도입후기	70%	19%	9%	2%

12 개월 동안 진행된 요구사항 명세기반 테스트를 분석한 결과 테스팅 개발 프로세스 도입 초기에는 요구사항 명세기반 테스트 요건에 맞지 않는 내용이 많아 deny 건수가 많이 발생 됨을 확인 할 수 있었으며, 도입후기에는 요건에 맞는 요구사항 명세서가 올라오면서 deny 건수가 줄어들고 accept 건수가 늘어나게 됨을 확인할 수 있었다.



[그림 5] 각 테스트 별 결함 발견 률

테스트 횟수당 요구사항 검토에 의해 개발된 건수는 평균 30 건 이었다. 이때 블랙박스테스트(단위테스트) 및 회기 테스트는 평균 3~4 개의 결함을 발견 하였다. 데이터 테스팅의 경우 해당되는 기능의 파라미터에 경계 값을 설정하고 테스트를 진행한 결과 예상치 못하던 결함들을 90% 이상 찾아내게 되었다.

7. 결론 및 향후 연구과제

본 논문은 X-internet 환경에서 사용되는 개발도구를 테스트 하는데 있어 제품의 안정성을 높이기 위한 소프트웨어 테스팅 기법을 이용하여 테스트를 진행 하고 각 테스트 별 장단점을 파악해 낼 수 있었다.

회기테스팅 기법의 경우는 제품의 안정성 중 base, 즉 이미 수행중인 기능에 대한 과거에 이미 개발된 기능의 안정성을 보장해주는데 적합한 테스트임을 확인할 수 있었다.

요구사항 명세기반 테스팅 기법은 형식에 맞는 요건에 의해 정확한 명세요건을 주어 개발하는데 도움이 되었으며, 또한 테스트 케이스를 유추해내는데도 효과적이었다.

블랙박스 테스팅은 정확한 요구사항 명세에 의해 개발이 되었는지 파악하는데 도움이 되는 테스팅 기법이었으며, 요구사항 명세에 의한 테스트 케이스 설계로 단위테스트에는 효과적이었다.

데이터 테스팅 기법은 예측하지 못한 데이터를 입력하여 소프트웨어가 출력하는 값에 대한 오류 여부를 판단하는데 적합한 테스팅 기법이었으며, 본 테스팅에 사용된 개발도구인 TrustForm4Designer 에 대한 안정성을 높이는데 효과적이었다.

향후 연구 과제로서 위의 테스팅 기법들을 응용하여 각 테스팅의 장점을 수용하고 단점은 보완하여 특화된 환경에서 사용되는 소프트웨어의 테스팅 방법론을 제시해야 할 것이며, 제품의 안정성을 체크할 수 있는 지표를 마련해야 한다. 또한, 테스팅을 통해 개발 비용을 줄일 수 있는 방안도 고려해야 할 것이다.

참고문헌

[1] IT 동향 X-Internet 입문

[2] 이명호 “X-인터넷 환경에서 디자인 패턴을 활용한 개발 프레임웍의 구축” 한국경영정보학회 추계학술대회, Vol.2004, No.0, pp. 555~558

[3] Myers, G., Sandler, C., Badgeet, T., and Thomas, T., The Art of Software Testing, Second Edition, John Wiley & Sons, 2004.

[4] Pressman, R., Software Engineering: A Practitioner’s Approach, McGraw-Hill, 2003.

[5] 소프트웨어 테스트 전문기술 기초과정 2006 한국정보통신기술협회 TTA-06002-IB pp.1~2

[6] Ron Patton 저 소프트웨어 테스팅 제 2 판 pp.75~79

[7] 한규정 정연대 “형식명세에 의한 소프트웨어 테스팅 자동화” 정보처리 제 4 권 제 4 호 1997.7 pp.46~56

[8] Walkthroughs, Inspections, and Technical Reviews, 3rd Edition Copyright 1990, 1982 by D.P. Freedman and G.M. pp. 294~295, 303~308