

X-Forms 기반 UI 개발 소프트웨어에 테스트 프로세스 적용을 위한 연구

이승혁

(주) 컴스퀘어

e-mail : opseung@comsquare.kr

A Study on Testing Process Application About the Software for X-Forms base UI Development

Seung-Hyuk Lee

Corporation of Comsquare

요 약

X-Forms 기반의 웹 UI(User Interface)를 개발하는 소프트웨어는 개발이 완료되어 운영되는 과정에도 사용자의 요구, 성능향상, 기능개선, 기능추가 등의 이유로 변경이 가능하다. 이러한 소프트웨어의 유지보수도 일반적인 요구사항명세, 분석, 설계, 구현의 개발 프로세스를 따른다. 본 논문에서는 UI 개발 소프트웨어의 유지보수 단계에서 효율적인 테스트를 하기 위해 V-모델을 확장, 변형한 테스트 프로세스 제안한다. 제안한 테스트 프로세스의 주요 활동은 요구사항분석을 통한 테스트 계획, 테스트 데이터를 식별하고 환경을 구축하는 분석과 설계, 테스트 케이스를 명세화하고 테스트 방법론을 적용하는 구현과 실현, 리포팅과 산출물을 정리하는 테스트 마감 활동으로 진행된다. 웹 UI 개발 소프트웨어의 특징에 맞게 테스트 프로세스를 구축하고 실무에 적용하여 에러 검출률, 테스트 시간, 테스트 결과의 효율성을 높일 수 있는 방법을 제안한다.

1. 서론

웹 기반 어플리케이션은 다양한 서비스를 제공한다. 그러나 사용자의 눈 높이에 따른 요구사항이 많아지면서 풍부한 사용자 인터페이스를 웹으로 제공받길 원하고 있다. 이러한 시대의 흐름에 탄생한 것이 X-Internet 과 X-Forms 이다. 2000 년 IT 조사기관인 포레스터리서치가 처음 고안해 낸 개념으로 실행 가능한 (e-Xecutable), 확장 가능한(eXtended) 인터넷이라는 의미로 실시간 양방향 통신, 유비쿼터스, 클라이언트 서버 환경에서 풍부한 인터페이스를 제공하는 장점이 있다. 이와 관련된 국내외 업체는 X-Forms 기반의 웹 UI 를 개발하기 위한 소프트웨어들을 속속 개발 및 출시하고 있다[1].

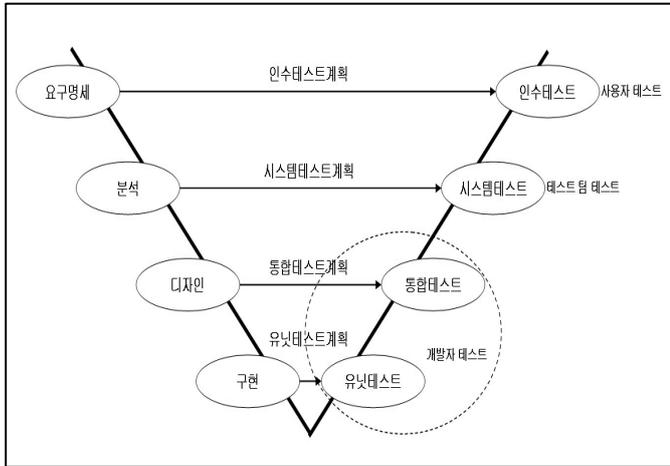
X-Forms 기반 웹 UI 개발 소프트웨어는 사용자의 요구사항변경, 성능향상, 기능개선, 기능추가 등의 다양한 이유로 변경될 수 있다. 변경된 소프트웨어는 이미 웹으로 구축이 완료되어 운영중인 사이트를 통해 사용자에게 배포되기 때문에 사용자가 에러를 발견하기 전에 철저하고 완벽한 테스트가 진행되어야 한다. 만약 운영 중인 사이트가 새로 개발된 소프트웨어로 인한 버그가 발생한다면 커다란 인적, 물적 손해를 발생 시킬 수 있다. 이러한 소프트웨어의 유지보수 활동에서 일어나는 일련의 개발 과정 또한 전체 개발 프로세스와 다르지 않다. 즉 개발 프로세스는 요구사항명세, 분석, 설계, 구현, 테스트의 과정을

거치는 것처럼 웹 UI 개발 소프트웨어의 유지보수에서도 위와 같은 과정을 거친다. 그러나 개발 프로세스의 마지막 단계에서 테스트를 수행하는 것이 아니다. 요구사항명세에서부터 구현에 이르는 개발 단계의 작업들에 대해 테스트를 포함하는 V-모델의 테스트 프로세스로 테스트한다[2]. V-모델은 개발자, 테스트 팀, 사용자 등이 테스트하지만 본 논문에서는 웹 UI 개발 소프트웨어라는 특징에 맞게 모든 테스트를 테스트 팀 또는 테스터를 기준으로 테스트를 계획하고 제어한다. 본 논문의 테스트 프로세스의 주요 활동은 요구사항분석을 통한 테스트 계획, 테스트 데이터를 식별하고 환경을 구축하는 분석과 설계, 테스트 케이스를 명세화하고 테스트 방법론을 적용하는 구현과 실현, 리포팅과 산출물을 정리하는 마감 활동으로 진행된다.

2. 관련 연구

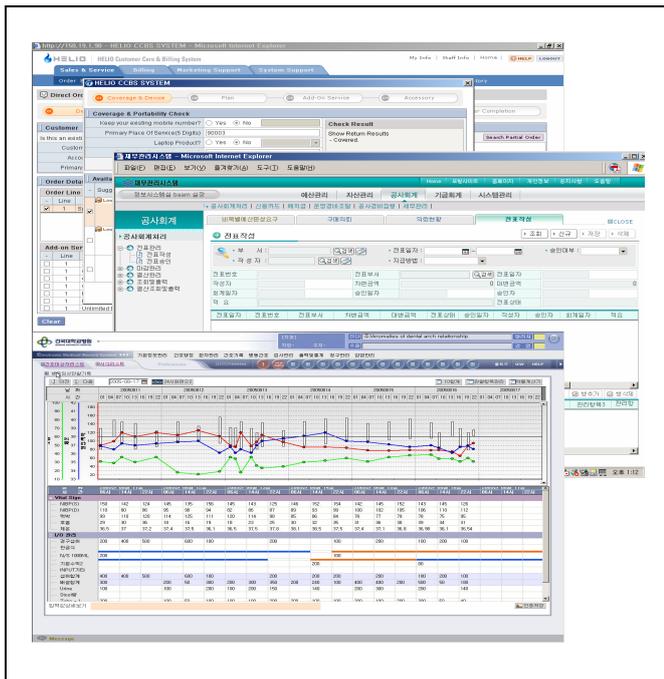
V-모델은 시스템이나 소프트웨어 프로그램을 테스트 할 때 단순히 코딩의 오류를 발견하는 것뿐만 아니라 요구사항명세에서부터 구현에 이르는 개발 단계에 테스트를 통합하여 체계화한 개념이다. 소프트웨어 생명 주기를 모형화하여 기존의 폭포수 모형(Waterfall Model)에서 시스템 검증과 테스트 작업을 명시하고 있다. 아래 (그림 1)은 V-모델을 도식적으로 보여주는 그림이다. 구현 단계에서 유닛 테스트를, 디자인 단계

에서 통합 테스트를, 분석 단계에서는 시스템 테스트를, 요구명세 단계에서는 사용자 인수 테스트를 계획, 설계, 수행한다[3].



(그림 1) V-모델

X-Forms 기반 웹 UI 를 개발하는 소프트웨어는 급변하는 기업환경에 대응하기 위해 개발된 W3C 의 XML/XFoms 등 표준 기술 기반으로 한다. 클라이언트/서버 환경의 장점과 웹의 장점을 수용한다. 이러한 소프트웨어들의 특징은 웹 UI 개발자가 개발 툴을 가지고 사용자 화면을 구성하여 파일로 저장한다. 파일 형태의 결과물은 클라이언트 상에서 웹 브라우저에 Plug-In 방식이나 단독 어플리케이션으로 운영된다[4]. 본 논문에서는 개발 툴을 테스트하는 것이 아니라 개발 툴로 개발이 완료되어 운영되는 웹 UI 화면을 구성하는 소프트웨어를 테스트하고자 한다. (그림 2)는 개발 툴로 개발되어 운영되는 웹 UI 화면이다.



(그림 2) UI 소프트웨어로 개발된 화면

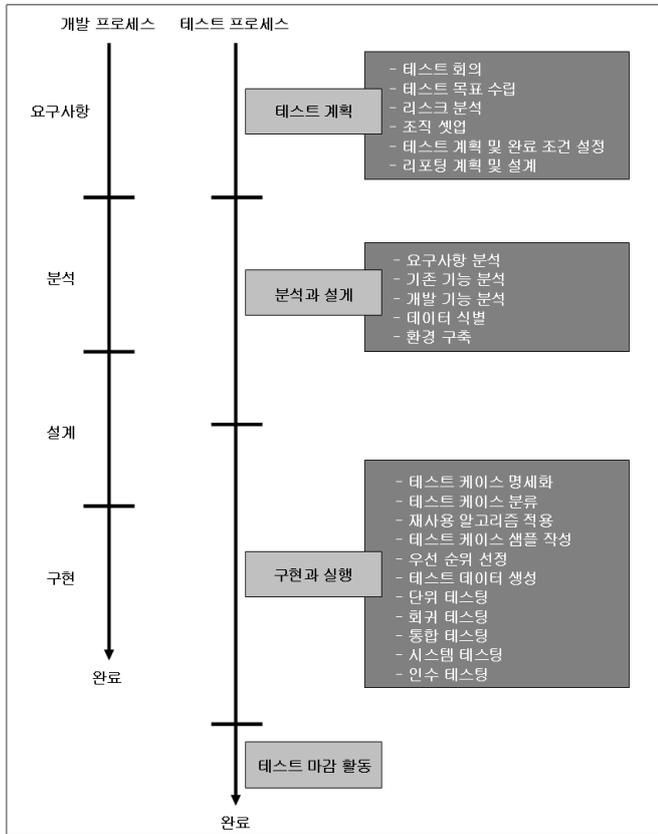
하나의 웹 UI 화면을 구성하는 소프트웨어는 여러 개의 컨트롤들로 구성된다. 컨트롤들은 속성과 메소드를 가지고 있으며 속성은 컨트롤의 모양을 정의하고 메소드는 컨트롤의 동작을 수행하도록 명령한다. 각각의 컨트롤들은 독립적인 동작을 수행하지만 개발 내용에 따라 유기적으로 상호 연동을 한다. 예를 들어 “button” 컨트롤을 클릭하였을 때 서버와의 통신을 하고 그 결과가 “datagrid” 컨트롤에 보여진다는가 하는 동작을 할 수 있다. 하나의 컨트롤에 속성들과 관계도 일정한 규칙을 가진다. 예를 들어 “input” 컨트롤은 문자나 숫자 형태의 데이터를 입력할 수 있다. 따라서 Format 속성을 설정할 수 있다. 숫자 Format 이 설정되면 반올림을 할 수 있는 속성도 설정할 수 있지만 문자 Format 을 설명하면 반올림을 할 수 있는 속성을 설정할 수 없다. 속성은 컨트롤의 모양을 설정하기 때문에 동작을 명령하는 메소드도 속성에 따라 다르다. 이렇듯 컨트롤에 설정하는 속성들과의 관계, 속성과 메소드의 관계, 컨트롤과 컨트롤들과의 유기적인 관계 등을 모두 테스트해야 한다.

개발된 기능들은 클라이언트에게 바로 배포되기 때문에 배포된 후에 발견되는 에러는 운영 중인 사이트에 커다란 손실을 줄 수 있다. 따라서 배포 전에 확실하고 완벽한 테스트가 테스트 팀에서 수행되어야 한다. 클라이언트의 환경은 운영체제와 브라우저에 따라 개발된 기능이 다르게 동작할 수 있다. 따라서 테스트 환경도 클라이언트 환경과 동일한 상태에서 테스트해야 한다. 또한 복잡하고 다양한 UI 환경이기 때문에 사용자의 요구명세에서부터 테스트의 관여가 필수적이다. 테스트는 개발되는 기능과 기존의 기능에 대해 충분히 숙지해야 명세기반, 구조기반, 경험기반 테스트가 가능하다. 본 논문에서는 이러한 X-Forms 기반의 웹 UI 소프트웨어에서 V-모델을 적용한다. 그러나 V-모델에서 사용자 테스트, 테스트 팀 테스트, 개발자 테스트로 나누어진 테스트 역할을 모두 테스트 팀에서 수행한다. 다만 프로그램의 구조상 구현의 유닛 테스트는 개발자가 한다. 따라서 개발과 독립된 테스트 조직이 분리한다.

3. 테스트 프로세스

테스트 프로세스는 개발 프로세스와 동일한 시점에서 시작한다. 사용자의 요구사항을 파악하는 최초의 단계에서부터 개발 팀과 테스트 팀이 같이 요구명세를 분석하고 회의를 통해 개발 계획과 테스트 계획을 수립한다. 테스트 계획은 테스트 회의를 통해 목표와 리스크 분석을 하고 조직을 셋업한다. 또한 완료 조건을 설정하고 리포팅 계획과 설계를 한다. 개발 프로세스의 분석과 설계 단계에서 테스트 프로세스도 분석과 설계를 진행하되 개발 프로세스의 설계 중간 단계에서 테스트 프로세스의 분석과 설계를 조속히 완료한다. 이것은 개발 프로세스의 모든 구현이 완료되기 전에 일부 완료된 기능을 테스트하므로 개발 및 테스트 시간을 단축하기 위함이다. 구현과 실행 단계에서 테스트 케이스를 명세화하고 샘플을 작성한다. 작성된 테스트 케이스는 기능별, 중요도별, 에러 발생

를 등을 기준으로 분류하고 테스트 케이스 알고리즘을 이용하여 테스트 케이스 샘플을 작성한다[5]. 이렇게 작성된 테스트 케이스를 단위, 회귀, 통합, 시스템, 인수 테스트를 한다. 마지막 단계로 테스트 마감 활동을 한다.



(그림 3) 테스트 프로세스

4. 테스트 계획

테스트 계획은 테스트의 목표를 달성하기 위해 필요한 활동 내역을 정의하는 것이다. 테스트 목표는 웹 UI 를 구성하는 컨트롤의 버그를 발견하고 사용명세에 의해 정의된 내역대로 개발이 완료되었음을 확인하는 것이다. 테스트 목표와 계획을 수립하기 위해서는 개발되는 기능에 대한 정확한 이해가 필요하다. 이를 위해 사용자의 요구사항을 파악해야 한다. 일반적으로 유지보수 단계의 요구사항은 사용자가 요구사항문서를 작성하여 개발팀에 전달한다. 전달된 내용은 개발팀의 개발여부, 개발계획 등을 정의하기 위해 회의를 갖는다. 이때 테스트 팀도 개발 팀과 같이 회의를 진행하여 개발될 기능에 대한 명세를 하고 상호의사를 전달 한다. 명세화된 요구사항에 테스트 범위를 설정하고 운영상 리스크를 분석한다. 리스크 분석은 해당 요구사항뿐만 아니라 운영되는 사이트들 간의 충돌이 없는지 확인해야 한다. 서로 다른 환경에서 개발된 사이트들은 같은 웹 UI 개발 소프트웨어를 사용하기 때문에 서로간의 충돌이 발생할 수 있다. 이렇게 테스트 범위와 리스크를 결정한 후 요구 수준,

테스트 레벨 별 목적을 커버하고자 하는 품질 특성 등에 대한 목적을 정한다. 테스트 목적이 완료되면 테스트 추정, 접근법, 전략에 따라 테스트 팀의 인원과 조직을 셋업한다. 셋업된 조직은 테스트 계획과 완료 조건을 설정하고 리포팅 계획 및 설계를 한다.

5. 분석과 설계

테스트 분석과 설계는 추상적인 테스트 목적을 실제적이고 구체적인 테스트 상황과 설계로 변환하는 과정이다. 이를 위해 명세화된 요구사항을 분석하고 개발되는 기능과 기존의 기능에 대한 정확한 이해가 필수적이다. 기존의 기능은 컨트롤들과 컨트롤의 속성, 메소드 등에 대한 정확히 이해하고 개발되는 기능이 기존의 기능에 어떠한 영향을 미칠 것인지 판단한다. 기능 분석을 통해 테스트 데이터를 식별하고 환경을 구축한다. 테스트 환경은 운영체제와 브라우저를 기준으로 분류하고 최대한 웹 UI 소프트웨어를 사용하고 있는 사이트와 동일하게 구축한다.

6. 구현과 실행

테스트 구현과 실행은 테스트 상황을 테스트 케이스나 테스트웨어로 변환하는 과정이다. 테스트 케이스는 소프트웨어에 대한 테스트 항목을 명세화하고 체크리스트를 작성한 것이다. 본 논문에서 테스트 케이스는 블랙박스 테스트 기법에 맞게 작성한다. 그러나 모든 기능에 대한 테스트 케이스를 작성하는 것은 아니다. 일반적으로 오류는 전체 테스트 케이스에 골고루 분포되지 않고 특정 영역에 집중되는 Pareto 형상을 따른다. 웹 UI 개발 소프트웨어의 유지보수 단계에서 기능 개발은 완전히 새로운 것이 아니기 때문에 재사용성을 위해 기존의 테스트 케이스를 분류한 후 선택적으로 사용한다. 분류 기준은 기능별, 중요도별, 에러 발생률로 한다. 1 차적으로 X-Forms 를 기반으로 기능을 분류하고 요구 사항에 따른 중요도를 계산한다. 중요도는 사용자의 요구사항에 해당하는 기능에 대해 점수를 부여하는 방식이며, 요구사항이 1 건 접수 시 1 점을 추가한다. 마지막으로 기존의 테스트 케이스에서 발생한 에러 발생률을 추가한다. 이렇게 분류된 테스트 케이스 중에서 점수가 높고 새로 개발되는 기능에 대한 테스트 케이스를 선별적으로 테스트한다. <표 1>은 재사용을 위한 테스트 케이스를 분류한 목록이다.

<표 1> 재사용을 위한 테스트 케이스 분류

기능	케이스의 수	중요도	에러 발생률
Bind	136	5	5%
Body	252	0	3%
Bool	132	0	6%
Browser	65	0	12%
Button	148	2	6%
Calendar	58	2	10%
Caption	71	1	5%
Case	65	3	6%
Checkbox	186	0	15%
Combo	325	5	25%
Datagrid	658	12	32%
Group	120	2	5%
Image	34	0	3%

Import	79	0	6%
Input	192	0	8%
Instance	164	0	8%
Iviewer	201	0	8%
Listbox	116	2	7%
Multilinegrid	359	9	24%
Output	267	0	6%
Radio	410	1	5%
Scrollbar	193	0	6%
Switch	320	0	6%
Textarea	123	0	7%
Treeview	162	6	6%

선택적으로 분류된 테스트 케이스로 개발된 기능에 대한 테스트가 부족할 경우 새로운 테스트 케이스를 작성한다. 이때 테스트 케이스는 X-Forms 스펙을 따르는 컨트롤들을 정적으로 구현하기 위해 HTML 을 사용하고 동적으로 구현하기 위해 JavaScript 나 VBScript 를 사용한다[5]. <표 2>는 테스트 케이스 알고리즘을 보여주고 있다.

<표 2> 테스트 케이스 알고리즘

	<pre><xhtml:html> <xhtml:head> <xhtml:title> 테스트케이스 </xhtml:title> </xhtml:head> <xhtml:body> <model> <instance> <root> <!--UI 에서 사용하는 데이터(XML)--> </root> </instance> </model> </xhtml:body> </xhtml:html></pre>
XML 데이터 선언	<pre><!--UI 에서 사용하는 데이터(XML)--></pre>
정적 기능 구현	<pre><controlname attribute="<!--정적 control 제어-->" Style="<!--정적 control 제어-->"</pre>
동적 기능 구현	<pre><script type="javascript" event="<!--동적 control 제어-->" <![CDATA[// 동적 control 제어를 위한 javascript 구문]]> </script> <script type="vbscript" event="<!--동적 control 제어-->" <![CDATA[// 동적 control 제어를 위한 vbscript 구문]]> </script></pre>
	<pre></xhtml:body> </xhtml:html></pre>

작성된 테스트 케이스를 테스터가 모두 테스트하는 것은 불가능하다. 단 하나의 기능을 테스트하기 위해서는 적게는 수백 개 많게는 수천 개의 테스트 케이스가 있기 때문이다. 따라서 새로 추가된 테스트 케이스만 테스터에 의해 단위 테스트를 진행하고 기존의 테스트 케이스는 테스트 자동화 도구를 이용하여 회귀 테스트한다. 회귀 테스트는 기존의 기능에 새로 추가된 기능으로 인한 충돌이 있는지 테스트하는데 목적이 있다. 단위와 회귀 테스트를 완료한 후 통합과 시스템 테스트를 한다. 통합 테스트는 운영체제, 파일 시스템, 통신 시스템과 상호 연동을 하는 동작을 테스트한다. 또한 기능적 특성을 물론 비기능적

특성을 테스트한다. 시스템 테스트는 실제 UI 소프트웨어를 사용하는 사이트와 동일한 환경이 구축된 테스트 환경에서 환경 제한적 장애 관련 리스크를 최소화하기 위한 테스트이다. 마지막 단계의 인수 테스트는 일반적으로 고객이나 사용자가 전담하여 수행하지만 웹 UI 개발 소프트웨어의 특성 상 테스트 팀에서 진행한다. 인수 테스트에서 UI 소프트웨어를 배포하고 실제 사용 가능한지 평가한다. 테스트 결과는 테스트 중인 소프트웨어, 테스트 도구, 테스트웨어의 식별과 버전을 수시로 기록한다. 테스트가 완료되면 예상 결과와 실제 결과를 비교한다. 예상 결과와 실제 결과간의 차이에서 오는 불일치를 인시던트(Incidents) 또는 결함(Defects)으로 보고한다. 보고된 내용이 재개발되었을 때 불일치 조치한 결과를 확인하기 위해 테스트 활동을 반복 수행한다.

7. 테스트 마감 활동

테스트 마감 활동의 첫 번째는 테스트 목표에 비하여 어느 정도 실제 테스트가 실행되었는지 평가해야 한다. 평가는 일반적으로 테스트 실행 결과가 테스트 계획에 명시된 완료 조건을 만족하는지 확인하고 추가적인 테스트가 필요한지 확인한다. 테스트 마감의 주된 활동은 데이터를 수집하여, 테스트에서 발견된 사실, 수치, 테스트 경험, 테스트웨어를 종합하고 축적한다.

8. 결론 및 향후 연구 과제

본 논문에서는 웹 UI 를 개발하는 소프트웨어를 테스트하기 위한 테스트 프로세스를 제안한다. 이를 위해 테스트하고자 하는 소프트웨어를 분석하고 V-모델을 확장 변형한 프로세스를 구축한다. 테스트 프로세스는 크게 테스트 계획, 분석과 설계, 구현과 실행, 테스트 마감활동으로 구분하고 세부활동을 명시한다. 구현과 실행에서 테스트 케이스를 분류하고 테스트 케이스 알고리즘을 개발하였으며, 단위, 회귀, 통합, 시스템, 인수 테스트를 한다.

향후 제안한 테스트 프로세스를 실제 업무에 적용하기 위하여 테스트 프로세스 사이트를 구축하고 시험 성숙도 모델등으로 제안한 테스트 프로세스를 평가할 필요가 있다.

참고문헌

- [1] 한국소프트웨어진흥원, <http://www.software.or.kr>
- [2] 권원일, 박은영, 조현길, “개발자도 알아야 할 소프트웨어 테스트 실무”, STA, 2006.
- [3] 김홍구, “소프트웨어 테스트 전문기술”, 한국정보통신기술협회, 2007.
- [4] 이창섭, 이승혁, 김응민, 조성환, 성락민, “TrustForm System User annual”, 컴스퀘어, 2007.
- [5] 이승혁, 이창섭, “X-Internet 환경에서 X-Forms 기반 UI 소프트웨어의 효율적인 테스트 케이스 작성을 위한 연구”, 제 27 회 한국정보처리학회 춘계학술발표대회 논문집, 05.2007.