

# 표준화를 통한 웹 어플리케이션 개발의 생산성 향상 방안

이보영

고려대학교 컴퓨터과학기술대학원 소프트웨어공학과

e-mail : bylee@lghitachi.co.kr

## Productivity Improvement of Web Application Development by Standardization

Bo-Young Lee

Dept. of Software Engineering, Computer Science Technology Graduate School  
Korea University

### 요 약

빠르게 발전하는 IT 기술을 기반으로 고생산성의 압박을 받고 있는 프로젝트는 보다 경쟁력 있는 생산성 확보가 프로젝트 성패의 관건이 되고 있다. 본 논문에서는 이러한 생산성 향상 압박에 대한 해결 방안으로 특히, 웹 어플리케이션 개발을 중심으로 Framework 의 적용이나 Tag Library 의 적용 등 간단한 표준화를 통한 생산성 향상 방안을 제시하였다.

### 1. 서론

21 세기 들어 빠른 속도로 IT 인프라가 발전하면서 바야흐로 인터넷 시대가 도래하였다. 그러다 보니 국내외를 막론하고 모든 시스템의 구축은 경이적인 통신 기술을 기반으로 장소와 시간에 구애 받지 않는 웹을 배제하고는 생각할 수 없게 되었다. 바야흐로 이제 웹은 시스템 구축 방식의 한 형태가 아니라 모든 시스템 구축의 기반으로 자리 잡게 된 것이다.

하지만 이렇게 빠르게 발전하는 IT 기술에 비해 인간의 시스템 개발 능력은 더디게 개선되고 있으며 신기술에 익숙해지기 전까지는 오히려 그 개발 속도나 품질이 현저히 떨어지는 경우까지 발생하고 있다. 게다가 웹을 비롯한 신기술의 발달로 좀더 빠르게 좀더 자동화로의 요구가 강해 지면서 시장 원리에 지배받고 있는 SI 프로젝트의 경우는 고객으로 부터 단기간, 고생산성, 고품질에 대한 압박을 더 강도 높게 받고 있다.

그로 인해 시스템 개발을 수행할 때에는 수많은 방법론, 개발 툴 등을 검토하고 있지만 해당 프로젝트에 최적화된 것을 선택하기도 어려울 뿐 더러 선택한 후

에도 원하는 고생산성, 고품질을 얻기 위해서는 상당한 기간이 걸린다는 어려움이 있다.

본 논문에서는 이런 과제를 해결하고 보다 경쟁력 있는 시스템 개발 능력을 확보하는 방법의 하나로 프로젝트 관리 능력, 생산성을 중요시하는 일본의 사례를 연구함으로써 생산성을 극대화할 수 있는 방안을 제시하고자 한다.

특히 방대한 SI 프로젝트의 시스템 중 웹 어플리케이션 개발을 중심으로 생산성 향상 수법 중 표준화의 수법을 통한 향상 방안을 제안하고자 한다.

### 2. 관련 연구

#### 2.1 소프트웨어공학에서의 생산성의 개념

본래 [생산성]이란 개념은 산업 사회로 발전하면서 투입된 Input 에 대해 어느 정도의 Output 이 나오는가를 정량적으로 측정하려는 의도로 만들어진 개념이다.

산출(Output) ← 생산량, 부가가치, 이익

투입(Input) ← 인원, 시간, 비용

이 생산성의 개념이 소프트웨어공학에도 적용이 되어 투입된 인력이 정해진 기간 내에 어느 정도의 산출물을 작성할 수 있는가를 공정별, 혹은 전 공정에 걸쳐 측정할 수 있는 지표로 소프트웨어 측정에 중요한 요소이다.

보통 step/人月의 단위로 측정한다.

2.2 시스템 개발에 있어서의 생산성 측정 방식

(1) 개발 작업의 구분과 공정의 대응

	업무 요건 설계 (RS)	소프트 웨어 방식 설계 (AS)	소프트 웨어 상 세 설계 (DS)	프로 그램 & 단체 테스 트 (P)	결합 테스 트 (CT)	통합 테스 트 (ST)
설계	■	■				■
제조			■	■	■	■

(2) 신규개발 생산성 측정 방식

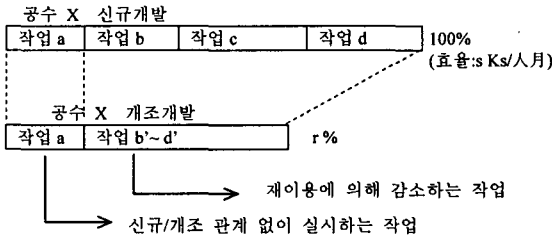
지표로서는 스텝수로 측정 한 개발량을 공수로 계산하는 방식을 사용한다.

(3) FP 생산성 측정 방식

IFPUG 방식으로 계산된 FP를 사용한다. 단, 미들 또는 부품만의 개발에는 적용 않는다.

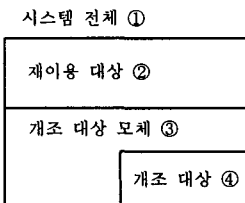
(4) 개조 생산성 측정 방식

각 공정의 성과물을 재이용한다는 전제로 계산한다.



① 개조 작업 효율(Ks/人月) =  $s \times r / 100$   
 $s$ : 신규개발의 생산성 기준치  
 $r/100$ :  $a + b' + c + d'$  (1~0)  
 $b' = b(1-n)$  (n: 재이용율)

② 재이용율 정의



[그림 1] 재이용율 정의

공정	재이용율
RS, AS, ST	②/① X 100 (%)
DS, P, CT	③/(③+④) x 100 (%)

2.3 웹 어플리케이션의 생산성 향상 수법

(1) 개발 프로세스의 개선

① 기존 Methodology 의 개선

J2EE, .NET 와 같이 설계 지향의 아키텍처의 등장으로 기존의 Structured 방법론 이나 객체지향 방법론만으로는 새로운 개발 환경에 적합한 프로세스를 완전하게 확립하기 어렵게 되었다. 그래서 기존의 개발방법론에다 새로운 기술의 특성과 개발 절차를 어느 정도 고려하여 변형 혹은 개선한 방법론을 적용함으로써 생산성 향상을 도모한다.

• RAD(Rapid Application Development)

우수한 개발 툴의 도입으로 개발 생산성을 극대화하고 프로토타입의 개발로 사용자의 요구를 조기에 확정하며 설계과정을 개발에서 그대로 적용함으로써 전체 개발 기간을 단축하는 것을 목적으로 개선된 방법론이다.

• CBD(Component-Based Development)

이미 검증된 컴포넌트를 재사용, 조합함으로써 고객의 요구에 맞는 소프트웨어를 빠르고 품질 좋게 구축하는 것을 목적으로 하는 방법론이다..

② 획기적인 신규 Methodology 적용

기존의 개발 절차나 패러다임을 무너뜨리는 획기적인 개발방법론의 적용으로 생산성의 향상을 도모한다.

• Agile 방법론

도큐먼트의 최소화, 고객 요구 사항에 대한 기민한 대응, Pair Programming, Test Driven 개발 등 기민하게 소프트웨어를 개발하는 것을 목적으로 하는 새로운 개발 방법론이다.

(2) 표준화

① Framework 적용

A. Framework 이란

시스템 개발시에 강제적으로 맞춰야 하는 표준화된 짜여진 구조(틀)이다.

B. Framework 의 장점.

- 시스템 구조 설계의 부담 경감
- 제공되는 공통 클래스의 재사용 가능
- 패턴화가 가능함으로 유지 보수에 용이

C. Framework 의 종류

- Jakarta Project 의 <Struts>
- <Pet Store>
- 일본 EC-One 의 <cFramework>

D. J2EE 와 .NET 에서의 Framework

J2EE 는 각 벤더들이 Framework 제품을 제공하여 생산성 향상을 지원하고 있는 반면 .NET 은 그 자체가 단일 Framework 구조로 이루어져 있어 그 기반에서 개발하는 것으로

바로 Framework 을 적용하게 되어 있다.

② TagLibrary 적용

Java 기반의 웹 어플리케이션 개발의 경우, JSP(Java Server Page) 기술이 MVC 모델의 프레젠테이션 층을 전담하는 방향으로 가고 있다. 그러나 JSP는 비즈니스 로직과 화면을 구성하기 위한 코드가 혼재되어 있어 유지보수 및 품질을 저하 시키는 요인을 가지고 있다. 그 해결책으로 SUN은 JSP1.1부터 Custom Tag를 지원하고 JSP1.2부터는 JSTL(JSP 표준 태그 라이브러리)의 사용을 권장하고 있다.

A. Custom Tag

<jsp:useBean />과 같은 JSP 상의 Action Tag는 SUN이 정해 놓은 표준 Tag이며 Custom Tag란 개발자가 필요할 경우 JSP에서 제공하지 않는 Tag를 정의하여 사용할 수 있는 Tag의 확장을 의미한다.

B. Custom Tag의 장점

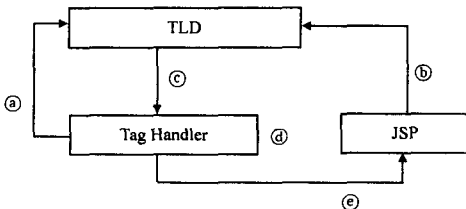
- 프레젠테이션 로직과 비즈니스로직의 분리
- Tag Library를 패키지 형태로 배포할 수 있으므로 다른 프로그램에서 재사용 가능.
- 웹페이지의 반복 작업을 Custom Tag화 함으로써 코드 작성의 생산성을 향상.

C. Tag Library의 구성

Custom Tag를 모아 전체적으로 관리하는 방식을 Tag Library라 한다.

- Tag Handler Class : Tag 정의
- Tag Library Descriptor(TLD): JSP와 Tag Handler의 연결 담당. XML 형식.
- JSP

D. Tag Library의 동작 원리



[그림 2] Tag Library의 동작 원리

- ㉑: Tag Handle 작성 후 TLD에 등록
- ㉒: Tag Library에 uri를 지정하고 해당 Tag를 호출
- ㉓: 해당하는 Tag의 Handler Class가 호출
- ㉔: Tag 내용 처리
- ㉕: 처리한 결과를 JSP에 리턴

3. 표준화 수법을 적용한 생산성 향상 사례 연구

3.1 Framework을 적용한 N社의 사례

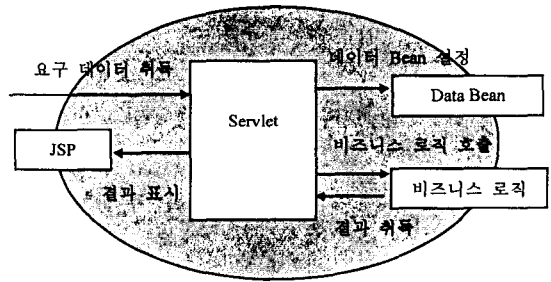
(1) Framework 적용 배경

여행업계에서 일본 시장의 세위 1위를 차지하는 N社는 전면적인 웹 시스템 구축을 앞두고 경쟁 업체의 시스템 오픈 일정이 알려지면서 보다 단기간에 대규모의 시스템을 구축한다는 전제 조건 하에 프로젝트

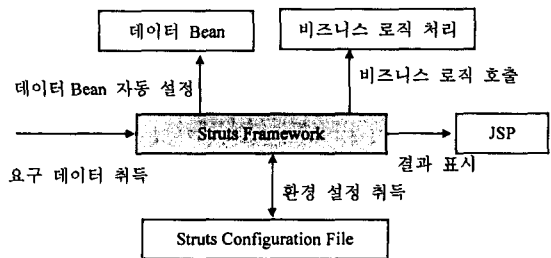
발주가 되었다. 이에 프로젝트를 수주한 PM은 충분하지 못한 설계 기간으로 인해 개발 중에 사양 변경이 다발할 수 있다는 리스크를 감안하여 범용 Framework인 Struts의 적용을 개발 방침으로 정하고 프로젝트를 수행했다.

(2) Struts의 적용 방안

Strut와 같은 Framework 구조를 가지지 않은 웹 어플리케이션을 작성할 경우 개발자는 클라이언트 요구에 대한 처리 흐름을 제어할 수 있는 소스 컨트롤러에 해당하는 Class(주로 Servlet)에서 일일이 정의/구현해 주어야 한다. 또한 업무 및 사양 변경에 따라 화면 천이 등이 변경될 경우 관련 Class를 직접 수정하고 재컴파일해야 한다. 이에 반해 Struts를 이용할 경우 Servlet, Bean, JSP의 흐름 구조를 제어할 소스를 코딩할 필요가 없으며 필요한 데이터는 Action Form Bean으로, 비즈니스 로직은 Action Class로 구현하여 만들고 Action Configuration File에서 흐름을 제어할 수 있도록 정의하는 것만으로 웹 어플리케이션 처리가 가능해졌다.



[그림 3] Framework를 사용하지 않는 구조



[그림 4] Struts Framework를 사용한 구조

(3) 적용 효과

Struts 도입 초기에는 Struts 등의 구조 등을 이해하고 개발자들이 적용하는 데 시간이 걸려 생산성에 별다른 효과를 얻지 못하였으나 Struts를 고려한 설계 공정이 끝나고 프로그램 제조 공정에 들어가면서 사양 변경 등에 대한 대응에 효율적으로 대응 가능하게 되었으며 신규 인원이 투입되어도 어느 정도 패턴화된 프로그램으로 인해 쉽게 적용하였다. 그로 인해 프로젝트 평균 생산성은 일본 SI 프로젝트에서 적용되고

있는 생산성 기준보다 0.4 Ks/人月 정도 향상되는 효과를 가져왔다.

### 3.2 Tag Library 를 적용한 Q社의 사례

#### (1) Tag Library 적용 배경

Q社의 경우, 고객 요건이라는 것이 없는 상태에서 설계-개발-테스트를 반복함으로 고객으로부터 피드백을 받아 가며 시스템을 구축해야 했기에 개발방법론으로 Spiral 방법론이 채택되었다.

게다가 요건 설계와 방식 설계에만 100 인/월 정도의 공수가 투입되는 대형 프로젝트였기에 PM 은 개발을 최대한 공통화 하는 방안을 검토, 그 중 하나로 화면 구성에 Tag Library 를 적용하기로 하였다.

#### (2) Tag Library 적용 방안

- ① 표시계의 화면이 많고 유사한 화면이 많은 점을 고려하여 반복적인 헤더 출력을 Custom Tag 화하기로 했다.
- ② 화면 공통화의 방안으로 Tag Library 를 적용, 화면레이아웃을 공통화하기로 했다.
- ③ 공통 스타일 시트와 스크립트 호출 부분을 Custom Tag 화하기로 하였다.

#### (3) 적용 효과

화면 레이아웃 공통화의 경우는 표준화의 효과로 반복 개발 시에 작업량을 상당히 많이 줄여 주었고, 재사용의 효과가 탁월하였다. 특히 Custom Tag 를 사용한 코딩으로 인해 JSP 의 가독성을 향상 시켜 화면계의 디버깅 작업이 보다 쉽게 이루어졌다. 그 결과 3 차 Spiral 개발 시에는 2 차 Spiral 개발 시보다도 1.2 배 정도 생산성이 향상되는 효과를 가져왔다.

```
<%@ page contentType="text/html; charset=euc-kr" %>
<%@ page import = "MyClasses.*" %>
<html>
<head><title> 工事件名検索条件 </title></head>
<body>
<%
MyBean mb = new MyBean() ;
mb.setUserId((String)request.getParameter("userId"));
mb.setUserName((String)request.getParameter("userName"));
...
%>
<%= mb.getUserId() %>
</body>
</html>
```

[그림 5] 스크립팅 요소를 적용한 JSP

```
<%@ page contentType="text/html; charset=euc-kr" %>
<jsp:useBean id="myBean" scope="page"
class="MyClasses.MyBean">
<jsp:setProperty name="myMember" property="*" />
</jsp:useBean>
<html>
<head><title> 工事件名検索条件 </title></head>
<body>
<jsp:getProperty name="myBean" property="userId" />
</body>
```

[그림 6] Action Tag 를 적용한 JSP

### 4. 결론 및 향후 연구

본 논문에서는 실제 프로젝트 사례에서 보여주고 있듯이 표준화된 Framework 의 적용이나 Custom Tag 등의 적용을 통한 웹 페이지의 공통화 만으로도 웹 어플리케이션의 생산성 향상에 상당한 효과가 있었음을 보여주었다.

특히 웹 기술은 하루가 다르게 발전하고 있는 중이어서 본 논문에서 제시한 방안 중 Tag Library 의 경우, JSP2.0 Spec 이 등장하면서 JSP 로 부터의 완전히 스크립팅 요소의 배제가 가능하도록 다양한 위치의 데이터에 쉽게 접근하도록 하는 언어 EL(Expression Language)이나 표준 Tag Library 인 JSTL 등이 추가되어 개발의 부담을 더욱 감소시킬 전망이다.

물론 이런 개발의 표준화를 통한 생산성 향상 방안은 그 범위가 설계, 제조 공정에 그치고 효과도 한계가 있을 수 밖에 없다. 따라서 경쟁력 있는 개발력 향상을 위해서는 프로세스 개선을 통한 프로젝트 전반에 적용할 수 있는 생산성 향상 방안을 확보하는 것이 절실하다. 특히 SI 프로젝트의 경우에는 다양한 업무와 인프라를 Integration 해야 하는 시스템인 만큼 프로젝트의 특성에 최적화된 생산성 향상 방식을 쉽고 빠르게 의사 결정할 수 있는 체크 포인트 및 가이드라인을 제시할 수 있도록 계속 연구해야 한다는 과제를 남기고 있다.

#### 참고문헌

- [1] 日立製作所情報・通信グループ生産技術本部 “システム開発における生産性ガイドライン” (2003)
- [2] 와세다 비즈니스 스쿨 “MOT 입문” (2004)
- [3] 처크개버네스 “자카르타 스트럿츠 프로그래밍” (2003)
- [4] MagungRydin 외 “JSP 태그 라이브러리” (2002)
- [5] Katharine Whitehead, "Component-Based Development: Principles and Planning for Business Systems" (2002)
- [6] 윤현식 “XP 사례를 통해 본 올바른 프로젝트 방법론 모색” (2003)
- [7] 연세대 김기영 “웹사이트 개발 방법론에 관한 연구” (2002)
- [8] Roger Fournier (유해영 역) “웹 어플리케이션 개발방법론” (2002)