

프레임워크 기반 온라인 서재 시스템의 효율적인 구현 및 품질 평가

박윤영[†], 변혜원^{**}

요 약

웹 2.0 환경에서 웹 서비스의 활용범위가 점차 확대됨에 따라서 웹 애플리케이션을 신속하고 편리하게 개발할 수 있는 방법이 요구되고 있다. 최근에 이러한 목적으로 도입되고 있는 프레임워크는 범용으로 설계되어 다양한 웹 애플리케이션에 적용되고 있지만 프레임워크를 이용하여 개발된 웹 애플리케이션에 대한 적합한 품질 평가 모델이 제시되지 않고 있다. 따라서 본 논문에서는 다양한 프레임워크를 이용하여 사용자 간에 도서 정보를 자유롭게 공유할 수 있는 온라인 서재 시스템을 개발하고, 이를 평가하기 위하여 프레임워크 기반의 웹 애플리케이션에 적합한 품질 평가 모델을 제시한다. 소프트웨어 품질 평가의 대표적인 국제 표준인 ISO/IEC 9126-2를 기반으로 기능성, 사용성, 효율성, 유지보수성, 안정성의 다섯 가지의 품질 특성을 정의하고 프레임워크 기반의 웹 애플리케이션의 품질 특성에 초점을 맞춘 평가 메트릭을 제시함으로써 온라인 서재 시스템에 대한 객관적이고 정량적인 평가를 수행한다.

Efficient Development and Quality Evaluation of Online Library System using Framework

Yoon-Young Park[†], Hae-Won Byun^{**}

ABSTRACT

In a web 2.0 environment, as a wide range of web services are being utilized, it requires quick and convenient way to develop web applications. For this purpose, the framework has been recently introduced and has been applied to web applications. However, software quality application model has not been presented for the web application using framework. In this paper, we develop web application, online library system using a number of frameworks. The system allows users to freely share book information. In order to evaluate the system, we present new software quality evaluation model for web applications using framework. The five quality attributes, functionality, usability, efficiency, maintainability, stability are defined based on the international standard for software quality evaluation, ISO/IEC 9126-2. We also propose the evaluation metrics for the web application using framework and perform quantitative evaluation for the online library system.

Key words: MVC(Model-View-Controller 모델), Framework(프레임워크), Web Application(웹 애플리케이션), Online Library System(온라인 서재 시스템), Software Quality Evaluation(소프트웨어 품질 평가), ISO/IEC 9126-2(ISO/IEC 9126-2 소프트웨어 품질모델)

※ 교신저자(Corresponding Author): 변혜원, 주소: 서울시 성북구 동선동 3가 249-1 동선동길 54-7(136-742), 전화: 02)920-7615, FAX: 02)920-7114, E-mail: hyewon@sungshin.ac.kr

접수일: 2010년 3월 2일, 수정일: 2010년 4월 21일
완료일: 2010년 4월 28일

[†] 준회원, 성신여자대학교 일반대학원 전산학과
(E-mail: luvchain@sungshin.ac.kr)

^{**} 정회원, 성신여자대학교 IT학부
(E-mail: hyewon@sungshin.ac.kr)

※ 본 연구는 2010년도 성신여자대학교 학술연구조성비(2010-1-11-049/1) 지원으로 수행되었음.

1. 서론

웹 애플리케이션은 인터넷이나 인트라넷을 통해서 웹 브라우저에서 이용할 수 있는 소프트웨어로서 사용자들의 PC에 일일이 소프트웨어를 설치해야 하는 번거로움을 덜어줄 뿐만 아니라 시스템의 관리가 용이하기 때문에 널리 사용되고 있다. 최근에는 웹 애플리케이션이 점차 대형화되면서 프레임워크의 필요성이 대두되고 있는데, 이는 프레임워크에서 제공하는 기본 구조와 핵심 소스코드를 이용함으로써 개발 비용을 줄일 수 있기 때문이다. 그러나 프레임워크를 이용한 웹 애플리케이션의 개발 사례가 급증하고 있음에도 불구하고 이를 평설계하고 이를 평가할 수 있는 소프트웨어 품질 평가 모델을 제안한다. 시스템의 효율적인 개발을 위해서 MVC(Model-View-Controller) 기반의 스프링 프레임워크를 이용하는데, 프레임워크를 이용하면 일관된 방향으로 개발을 주도할 수 있기 때문에 개발의 생산성과 유지보수성의 향상을 기대할 수 있다. Daum의 도서 검색 API를 통하여 방대한 양의 도서를 쉽고 빠르게 검색할 수 있도록 하고, 등록된 도서에 대한 리뷰, 별점, 인용구 등을 다른 사용자들과 공유하도록 한다. 정적 소스분석 도구를 이용한 디버깅 작업과 소스코드 버전 관리를 통하여 시스템의 결함을 최소화시키는 동시에 시스템을 최적화시키는 방안을 모색한다. 본 연구에서 제안하는 온라인 서재 시스템의 품질 평가를 수행하기 위하여 프레임워크 기반 웹 애플리케이션의 품질 특성을 분석하고, ISO/IEC 9126-2에 기초한 평가 모델을 제안한다. 각 품질 특성에 대한 평가 매트릭스를 제시하고 사용자들의 평가 데이터를 분석함으로써 시스템을 객관적이고 정량적으로 평가하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 MVC 패턴과 프레임워크, 소프트웨어의 품질 평가 모델, 웹 애플리케이션의 평가에 대해서 살펴본다. 3장에서는 제안하는 온라인 서재 시스템의 설계와 구현 내용을 설명하고, 4장에서는 프레임워크 기반의 웹 애플리케이션 품질 평가에 대해 기술한다. 마지막 장에서는 결론과 향후 연구 방향을 제시한다.

2. 관련연구

2.1 MVC 패턴과 프레임워크

MVC 패턴은 사용자 인터페이스와 비즈니스 로직

부분을 효과적으로 분리하기 위해서 고안된 아키텍처 패턴으로서 모델, 뷰, 컨트롤러로 구성된다[1]. MVC 패턴에 따라서 시스템을 설계하면 복잡한 애플리케이션의 관리와 유지보수가 용이해지며 각 컴포넌트들을 동시에 개발할 수 있기 때문에 개발의 생산성이 높아진다는 장점이 있다.

최근 웹 애플리케이션 개발에서 이용사례가 증가하고 있는 프레임워크는 개발에 필요한 구성 요소들을 미리 정의해 놓은 것이며, 프레임워크에서 제공하는 기본구조 및 핵심코드를 사용함으로써 개발에 드는 비용을 절감할 수 있다[2]. 또한, 검증된 개발환경을 통하여 신속하고 안정적인 프로그램의 개발이 가능하기 때문에 개발효율성을 높일 수 있다. MVC 패턴 기반의 대표적인 프레임워크로 대용량의 엔터프라이즈 애플리케이션에서 주로 사용되는 스프링(Spring)과 스트러츠(Struts)가 있으며, 그 밖에도 프로젝트의 효과적인 관리를 제공하는 메이븐(Maven)과 웹 페이지의 공통 레이아웃을 관리하기 위한 사이트메쉬(Sitemesh) 등 다양한 프레임워크가 있다.

2.2 ISO/IEC 9126

소프트웨어의 평가는 기본적으로 국제 표준으로 제정되어 있는 ISO/IEC 9126의 소프트웨어 품질 모델을 따르고 있다[3]. ISO/IEC 9126에서는 소프트웨어 제품에서 요구되는 품질을 정량적으로 평가하기 위하여 소프트웨어에 관한 품질 특성과 품질 부특성을 정의한다. ISO/IEC 9126-1은 소프트웨어의 품질을 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성의 여섯 가지 특성으로 규정하는데, 6개의 품질 특성은 다시 27개의 부특성으로 구성된다[4]. 그림 1은 ISO/IEC 9126-2 소프트웨어 품질 평가 모델로서 ISO/IEC 9126-2에서는 제품 개발 완료 후 실행코드 상태의 소프트웨어가 요구사항을 만족시키는 정도를 평가하는 외부 매트릭(External metrics)을 규정한다[5]. ISO/IEC 9126-3에서는 개발 단계에서의 내부 매트릭(Internal metrics)과 관련된 측정 기준을 제시하고[6], ISO/IEC 9126-4에서는 특정 환경에서 사용자의 작업 유효성, 효율성, 만족도 등 사용법(Quality in use metrics)에 대한 기준을 제시한다[7]. 그러나 앞서 설명한 품질 평가 모델들은 소프트웨어의 일반적인 특성을 기준으로 제안된 것이기 때문에 객관적이고 정량적인 품질 평가를 위해서는 소프트

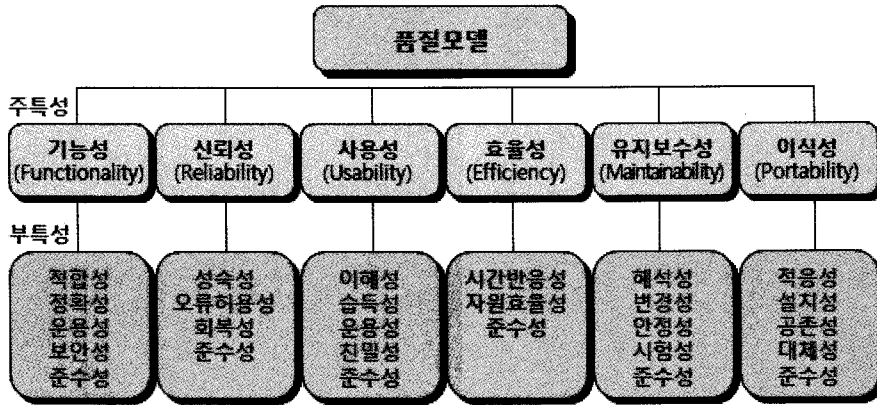


그림 1. ISO/IEC 9126-2 소프트웨어 품질 평가 모델

웨어의 특성을 반영한 평가 메트릭을 이용하여 평가가 이루어져야 한다.

2.3 웹 애플리케이션 평가 연구

질적으로 우수한 웹 애플리케이션에 대한 요구가 증대됨에 따라서 웹 애플리케이션의 품질 평가 모델과 평가 방법에 대한 연구가 진행되어 왔다. Mavroustakos 등은 웹 애플리케이션의 기능이 요구를 만족하는지 검증하기 위하여 구조적 측면과 사용성의 두 가지 기준을 바탕으로 효율적인 웹 애플리케이션 평가 모델을 제안하였다[8]. Tomita는 웹사이트의 평가 기준을 관리(Administration), 디자인(Design), 품질(Quality)의 세 가지로 구분하고, 정보의 정확성, 다른 사람의 정보에 대한 접근 가능성, 최신 정보의 업데이트 등을 평가 항목으로 이용하였다[9]. Santos는 박물관 웹사이트의 품질을 비교 평가하는 사례연구를 수행하기 위하여 웹 사이트의 목적 및 웹 사이트의 특성을 구체화하고 각 특성에 대한 평가 기준을 정의하는 등 웹사이트를 평가하기 위한 6단계의 평가 프로세스를 제안하였다[10].

박선영 등은 웹 서비스 테스트 단계에서의 기능성 평가에 초점을 맞추고 테스트 케이스 수, 요구사항 수, 결함 수 등의 평가지표를 기반으로 품질 평가 모델을 제시하였다[11]. 김용원은 웹 서비스의 품질 평가에 대한 기존 연구들을 분석하고 웹 서비스를 구조적인 측면과 데이터의 측면으로 분류하여 평가 모델을 설계하고, 웹 서비스의 사용 목적에 따라서 평가 항목에 가중치를 적용함으로써 사용 목적을 고려한 웹 서비스의 품질 평가를 수행하였다[12].

3. 온라인 서재 시스템의 설계 및 구현

3.1 시스템 개요

본 논문에서 제안하는 온라인 서재 시스템은 도서를 검색하여 원하는 도서를 자신의 온라인 서재에 담아 관리하고, 관련된 도서 정보를 다른 사용자들과 공유하는 서비스를 제공한다. 사용자들은 온라인 서재에 등록된 도서에 대하여 리뷰, 별점, 인용구, 방명록과 같은 다양한 정보를 추가할 수 있다. 그림 2는 시스템 개요도로서 크게 온라인 서재 시스템과 시스템 관리로 구성되며, 이를 구현하기 위하여 스프링 프레임워크에 메이븐, 사이트메쉬 등의 프레임워크가 결합된다. 온라인 서재 시스템은 도서 검색과 리뷰 등록으로 구성되고, 시스템 관리는 정적 소스분석 도구를 통한 시스템의 최적화와 SVN(SubVersion)을 이용한 소스코드의 버전 관리 부분으로 이루어진다.

3.2 데이터베이스 설계

본 시스템에서는 도서 검색 API를 통하여 방대한 도서 정보에 자유롭게 접근할 수 있다. 따라서 모든 도서 정보를 데이터베이스에 가지고 있을 필요 없이 사용자가 서재에 등록된 도서 정보에 대해서만 데이터베이스에 저장하여 관리한다. 표 1, 2는 주요 테이블인 서재, 도서 테이블의 레코드이며 그 밖에도 사용자, 별점, 인용구, 방명록 등의 테이블이 있다.

3.3 구현 결과

시스템 구현 환경으로 이클립스 3.14 JEE 버전을

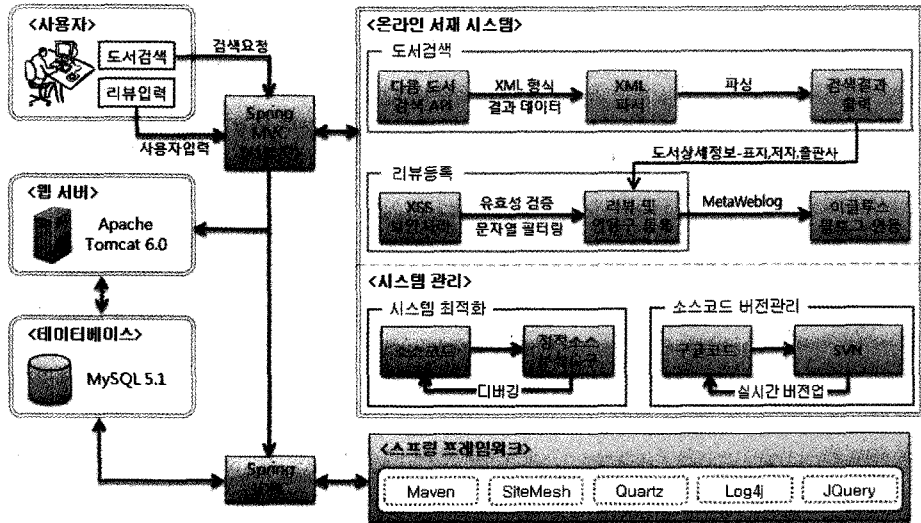


그림 2. 온라인 서재 시스템 개요도

표 1. 서재 테이블 레코드

| 필드 | 타입 | 설명 | 필드 | 타입 | 설명 |
|------------|---------|--------------|--------|---------|-----------|
| idNum | int(11) | 서재 고유 id(PK) | notice | text | 공지사항 |
| user_idNum | int(11) | 사용자 id(FK) | count | int(10) | 사용자 방문 횟수 |
| isopen | int(1) | 서재 공개/비공개 설정 | | | |

표 2. 도서 테이블 레코드

| 필드 | 타입 | 설명 | 필드 | 타입 | 설명 |
|--------------|--------------|--------------|-------------|--------------|-------|
| idNum | int(11) | 도서 고유 id(PK) | price | int(5) | 가격 |
| name | varchar(100) | 도서명 | cover | varchar(100) | 책 표지 |
| isbn | char(13) | ISBN | category | varchar(50) | 카테고리 |
| publish_date | varchar(20) | 출판일 | description | text | 도서 설명 |

이용하고, 스프링 프레임워크 2.5와 Tomcat 6.0 웹 서버, MySQL 5.1 데이터베이스를 연동한다. 온라인 서재 시스템은 크게 도서 검색과 리뷰 등록, 보안 및 시스템 관리로 이루어진다.

3.3.1 OpenAPI를 이용한 도서 검색

사용자는 자신의 온라인 서재를 할당받아서 도서를 검색하고 원하는 도서를 자유롭게 분류하여 관리할 수 있다. 그림 3은 MVC 패턴을 따르는 도서 검색 프로세스로서 클라이언트의 요청에 대한 내부 처리에 관해 설명한다. 사용자가 도서 검색을 요청하면 BookController 도메인 클래스는 BookService 인터페이스를 통하여 입력받은 검색 키워드를 포함하는 URL을 호출하는데, 이때 Daum의 도서 검색 API를

이용한다. 도서 검색 요청에 대한 응답으로 Daum의 서버로부터 XML 형태의 결과 데이터를 전송받으면 시스템의 SqlParser 인터페이스를 통하여 파싱된 결과를 사용자 화면에 출력한다. 사용자들이 원하는 도서를 서재에 등록할 때, BookDao 인터페이스를 통해서 도서 정보를 전달받아 데이터베이스에 저장한다.

3.3.2 리뷰 및 인용구 등록

사용자는 도서에 대해 리뷰를 작성하고 기억에 남는 인용구를 등록하여 다른 사용자들과 공유할 수 있다. 리뷰를 편리하게 작성할 수 있도록 OpenAPI를 통하여 도서의 기본 정보를 가져옴으로써 도서의 표지 그림이나 저자, 출판사, 가격과 같은 정보를 일일이 입력해야 하는 번거로운 작업을 줄여준다. 또한,

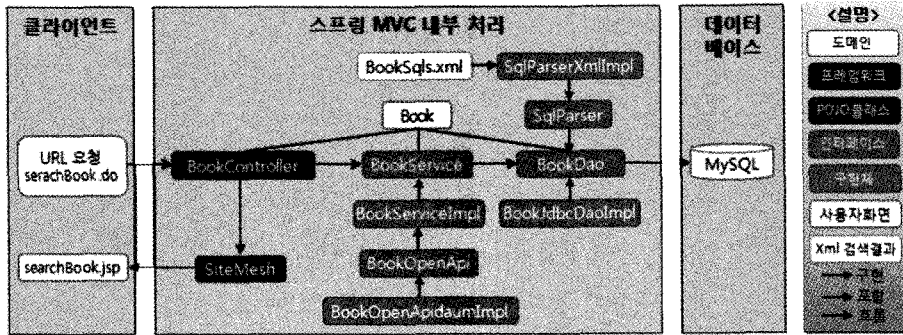


그림 3. MVC 패턴을 따르는 도서 검색 프로세스

사용자가 온라인 서재에 등록된 리뷰를 자신의 다른 블로그에 즉시 포스팅하는 기능을 지원하기 위해서 MetaWeblog API를 이용하여 온라인 서재 시스템과 블로그를 연동한다. 그림 4는 온라인 서재 시스템의 메인 화면으로 가장 최근에 등록된 도서 정보와 리뷰, 별점 리스트를 보여준다.

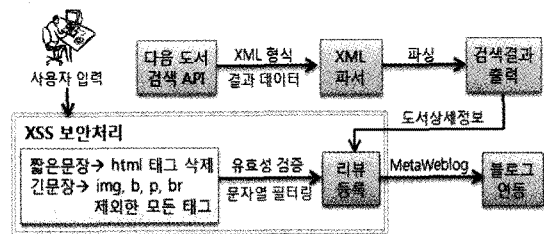


그림 5. 리뷰 등록에서의 XSS 보안 처리

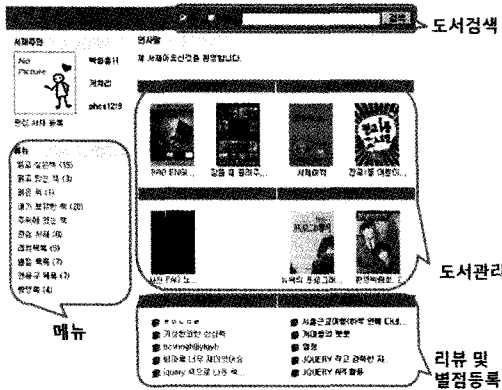


그림 4. 온라인 서재 시스템의 메인 화면

사용자에게 정상적인 데이터가 아닌 악성 스크립트가 포함된 데이터를 입력받는 경우, 서버에서 실행된 악성 스크립트를 통해서 사용자의 개인 정보가 유출될 수 있기 때문에 XSS(Cross Site Scripting) 공격에 대한 보안처리가 요구된다. 본 시스템에서는 정규표현식(Regular Expression)을 이용하여 입력받은 문자열의 패턴을 검증하고, 유효한 경우에만 데이터베이스에 저장한다. 위의 그림 5는 도서 리뷰가 등록되는 과정이며, 제목이나 이름과 같이 HTML 태그가 불필요한 짧은 문장에서는 태그를 삭제하고, 리뷰나 인용구처럼 긴 문장을 입력받는 경우에는 일부 태그를 허용하여 리뷰 작성을 돕는다.

3.3.3 시스템 관리

시스템을 효율적으로 관리하기 위해서 개발 과정에 정적 소스 분석을 통한 디버깅과 SVN을 이용한 소스코드 버전 관리를 포함한다. 정적 소스 분석을 통하여 초기화되지 않거나 중복으로 정의된 변수와 같이 오류를 유발시키는 요소에 대한 정보를 얻을 수 있다[13,14]. 본 시스템에서는 메릴랜드 대학에서 개발한 정적 소스분석도구 FindBugs를 이용하여 구문이나 객체 오류를 검출한다[15].

버전 관리는 공동 개발에서 발생할 수 있는 소스코드의 충돌을 방지할 뿐만 아니라 개발의 히스토리를 남길 수 있기 때문에 대규모의 시스템을 개발하는데 반드시 필요하다. 본 시스템을 개발할 때 SVN과 구글 코드(Google Code)를 연동하여 버전을 관리하였는데, 구글 코드는 프로젝트 호스팅을 제공하는 동시에 개발 이슈와 스케줄 등을 관리할 수 있는 기능을 제공하여 보다 원활한 공동 작업을 가능하게 한다.

4. 프레임워크 기반 웹 애플리케이션의 품질 평가

ISO/IEC 9126-2 품질 평가 모델은 소프트웨어의 일반적인 특성을 기준으로 제안된 것이므로 프레임워크 기반 웹 애플리케이션의 특성들을 분석하여 이

를 바탕으로 새로운 품질 평가 기준과 평가 메트릭을 제안하여 평가를 수행한다.

4.1 프레임워크 기반 웹 애플리케이션의 품질 특성

프레임워크 기반 웹 애플리케이션의 품질 평가 방법에 관해 연구하기에 앞서 먼저 웹 애플리케이션의 사용 경험이 많은 사용자들을 대상으로 웹 애플리케이션 사용 시 자주 발생하는 오류들에 대해서 수집하였다. 사용자들이 공통적으로 응답한 오류 내용은 표 3과 같다.

위의 설문조사에 기초하여 웹 애플리케이션의 특성을 분석해보면 웹 애플리케이션은 불특정 다수의 사용자가 접속할 수 있으며 사용자의 권한에 따라서 서로 다른 수준의 서비스를 제공한다는 특성을 가진다. 따라서 네트워크상에서 발생 가능한 많은 문제들과 사용자의 개인 정보 보안 문제를 고려해야 하며, 사용자에게 권한을 부여할 수 있어야 한다.

웹 애플리케이션 프레임워크는 웹 애플리케이션 또는 웹 서비스를 쉽게 개발할 수 있도록 구성요소를 모아놓은 것으로서 데이터베이스 연동, 세션 관리, 표준 템플릿, 소스 코드의 재사용 등을 지원한다. 웹 애플리케이션의 개발에서 프레임워크를 이용하는 경우, 기본 구조와 공통의 모듈을 사용함으로써 개발 기간을 단축시킬 뿐만 아니라 검증된 개발 환경을 제공받을 수 있어서 코드의 품질을 향상시킬 수 있다. 특히, MVC 기반의 프레임워크는 애플리케이션의 구성 요소를 모델, 뷰, 컨트롤러로 구분하고, 일관된 패턴에 따라서 구현하도록 유도하기 때문에 클래스 간의 의존성을 줄일 수 있다. 이는 유지보수성의 향상과 밀접한 관련이 있으며 기존의 시스템과 신규 시스템의 안전한 통합이 가능하고 요구사항의 변경에 대해서 빠르게 대처할 수 있다는 것을 의미한다. 그러나 다수의 프레임워크들을 결합하여 사용하기 위해서는 구성요소 간의 호환성을 고려하고 표준을 준수함으로써 일관성을 유지해야 한다.

4.2 프레임워크 기반 웹 애플리케이션의 평가 모델

위에서 분석한 품질 특성들에 따라서 ISO/IEC 9126-2 품질 평가 모델을 토대로 하여 프레임워크 기반의 웹 애플리케이션에 특성화된 품질 평가 모델을 제시하는데, 품질 특성으로 기능성, 사용성, 효율성, 유지보수성, 안정성을 정의한다. 소프트웨어 품질 평가에서 기능성은 가장 기본이 되는 평가 항목이며, 웹 애플리케이션에서는 사용자들이 직접 조작하는 부분이 많으므로 높은 사용성이 요구된다. 웹 기반의 서비스에서는 실시간으로 데이터를 주고받으며 동작하기 때문에 시스템 효율성의 평가가 중요한데, 특히 프레임워크를 이용했을 때 향상될 수 있는 개발 생산성을 정량적으로 평가할 수 있어야 한다. 또한, 웹 애플리케이션의 특성상 기능이 추가되거나 사용자가 증가하면 시스템의 유지 보수 작업이 불가피하므로 시스템의 유지보수성에 대해서도 평가되어야 한다.

따라서 본 연구에서는 기존의 품질 평가 항목에서 기능성, 사용성, 효율성, 유지보수성의 주특성을 도출하고 웹 애플리케이션과 프레임워크의 사용에서 기대할 수 있는 몇 가지의 세부 특성을 추가한다. 웹 애플리케이션에서 제공하는 기능과 인터페이스가 편리하게 구성되어 있는지 평가하는 기준으로서 사용성 아래 편리성을 정의한다. 프레임워크와 Open-API를 이용함으로써 효율적인 개발과 소스코드, 클래스의 재사용이 가능한지 평가하는 기준으로서 효율성 아래 개발효율성을 정의하고, 유지보수성 아래 재사용성을 정의한다. 프레임워크를 이용하면 일관된 패턴에 따라서 체계적인 개발이 가능하기 때문에 웹 애플리케이션 동작의 안정성을 기대할 수 있다. 웹 애플리케이션의 안정성을 평가하기 위한 기준으로서 호환성, 일관성, 준수성을 정의하는데, 호환성은 프레임워크 및 구성 요소들 간의 호환성을 평가하기 위한 기준이며, 일관성은 개발방식과 시스템 동작의 일관성을 평가하는 기준이다. 준수성은 웹 애플리

표 3. 웹 애플리케이션 사용 시 자주 발생하는 오류들

| No | 내 용 | No | 내 용 |
|----|---------------------------|----|---------------------------|
| 1 | 네트워크 오류(접속 끊김/서버오류 등) | 5 | 웹 페이지 다운(처리 지연/시스템 과부하 등) |
| 2 | 이미지 또는 동영상 깨짐 현상 | 6 | 레이아웃 문제(원하는 기능을 찾을 수 없음) |
| 3 | 보안 문제(아이디 도용/해킹/XSS 문제 등) | 7 | 권한 오류(로그인 실패/페이지 접근 불가 등) |
| 4 | 링크 오류(잘못된 페이지 이동/링크 없음 등) | 8 | 기타 원인을 알 수 없는 오류 |

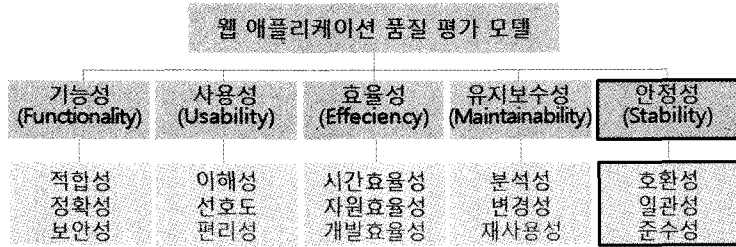


그림 6. 프레임워크 기반 웹 애플리케이션의 품질 평가 모델

케이션에 있어서 지켜져야 할 표준과 규약에 대한 준수성을 평가하는 기준이다. 그림 6은 본 논문에서 제안하는 프레임워크 기반 웹 애플리케이션의 품질 평가 모델이다.

4.2.1 기능성 평가

기능성은 시스템의 기능이 내재된 요구를 만족하는가를 평가하는데, 세부 특성으로 적합성, 정확성, 보안성을 가진다. 적합성은 표 4, 5와 같이 시스템에서 요구된 기능이 완전하게 구현되었는지, 기능의 구성이 적절한지에 대해서 평가한다.

기능의 정확성은 구현된 기능이 정확한 결과 데이터를 제시하는지, 사용자가 입력한 데이터와 데이터베이스에 저장된 데이터가 정확하게 일치하는지, 링크에 적용된 기능이 정확하게 동작하는지에 관한 항목을 평가한다(표 6-8).

보안 통계 보고서에 따르면 약 12,186개의 웹 애플리케이션에서 97,554개의 보안 취약점이 진단되었는데, XSS에 대한 사례가 가장 많았으며 정보 누출, SQL 인젝션의 순서로 나타났다[16]. 이처럼 웹 애플리케이션에서는 보안성을 높이기 위하여 데이터 암호

표 4. 기능 구현 완전성에 대한 평가 매트릭

| | |
|-------|-----------------------------|
| 평가매트릭 | 요구사항에 따라 기능이 얼마나 구현되었는가? |
| 측정 항목 | A 구현된 기능 수 |
| | B 요구 명세서에 정의된 전체 기능 수 |
| 계산식 | 기능 구현 완전성 = (A/B) * 100 (%) |

표 5. 기능 구성 적합성에 대한 평가 매트릭

| | |
|--------|---------------------------------------|
| 평가 매트릭 | 기능 구성이 적절한가? |
| 측정 항목 | 요구명세서에 정의된 기능에 대해서 구성이 적절하게 이루어졌는지 평가 |

표 6. 결과 데이터의 정확성에 대한 평가 매트릭

| | |
|--------|-------------------------------------|
| 평가 매트릭 | 구현된 기능이 정확한 결과 데이터를 제시하는가? |
| 측정 항목 | A 잘못된 결과 데이터가 출력된 케이스 수 |
| | B 사례연구에서 제시한 전체 테스트 케이스 수 |
| 계산식 | 결과 데이터의 정확성 = {1 - (A/B)} * 100 (%) |

표 7. 데이터 일치성에 대한 평가 매트릭

| | |
|--------|---|
| 평가 매트릭 | 사용자의 입력 데이터와 데이터베이스에 저장된 데이터가 정확하게 일치하는가? |
| 측정 항목 | A 데이터가 일치하지 않는 케이스 수 |
| | B 사례연구에서 제시한 데이터 입력에 대한 전체 케이스 수 |
| 계산식 | 데이터 일치성 = {1 - (A/B)} * 100 (%) |

표 8. 기능 적용의 정확성에 대한 평가 매트릭

| | |
|--------|------------------------------------|
| 평가 매트릭 | 기능이 정확하게 적용되었는가? |
| 측정 항목 | A 비정상적인 기능이 적용된 케이스 수 |
| | B 버튼 또는 링크에 적용된 전체 기능 수 |
| 계산식 | 기능 적용의 정확성 = {1 - (A/B)} * 100 (%) |

호화, 세션 관리 등의 접근 통제성과 접속한 사용자와 핵심 정보 변경, 잘못된 접근에 대한 로그 기록 등 접근 감시성에 대한 평가가 요구된다. 표 9-11은 보안성을 평가하기 위한 매트릭이다.

표 9. 접근 통제성에 대한 평가 매트릭

| | |
|--------|--------------------------|
| 평가 매트릭 | 접근 통제에 대한 기능이 있는가? |
| 측정 항목 | A 비정상적인 접근이 통제된 케이스 수 |
| | B 사례연구에서 접근통제에 대한 케이스 수 |
| 계산식 | 접근 통제성 = (A/B) * 100 (%) |

표 10. 접근 감시성에 대한 평가 매트릭

| | | |
|--------|--------------------------|-----------------------|
| 평가 매트릭 | 접근 감시에 대한 기능이 있는가? | |
| 측정 항목 | A | 접근 감시 기능이 동작한 케이스 수 |
| | B | 사례연구에서 접근통제에 대한 케이스 수 |
| 계산식 | 접근 감시성 = (A/B) * 100 (%) | |

표 11. 데이터 보안성에 대한 평가 매트릭

| | | |
|--------|--------------------------------|--------------------------------|
| 평가 매트릭 | 사용자 개인 정보 등 중요한 데이터가 보호되고 있는가? | |
| 측정 항목 | A | 사용자의 중요한 데이터가 암호화되어 보호된 케이스 수 |
| | B | 사례연구에서 사용자의 중요한 데이터에 접근한 케이스 수 |
| 계산식 | 데이터 보안성 = (A/B) * 100 (%) | |

4.2.2 사용성 평가

사용성은 사용자가 시스템을 이해하고 사용하는 것을 평가하며 기존의 이해성, 선호도에 편리성의 세부 특성을 추가한다. 웹 애플리케이션 사용 시 자주 발생하는 오류에 대한 설문을 토대로 하여 시스템이 제공하는 기능, 인터페이스, 오류 메시지에 대한 이해성과 시스템의 친밀성, 선호도를 평가 항목으로 설계한다(표 12-14).

편리성은 사용자 인터페이스와 화면 레이아웃이 사용하기에 편리하도록 구성되어 있는가에 초점을 맞추어 평가하며 사용되는 평가 매트릭은 표 15, 16과 같다.

표 12. 기능 이해성에 대한 평가 매트릭

| | | |
|--------|--------------------------------|------------------------|
| 평가 매트릭 | 제공하는 기능을 이해하고 사용할 수 있는가? | |
| 측정 항목 | A | 사례연구에서 이해하지 못한 기능 수 |
| | B | 사례연구에서 제시한 전체 기능 테스트 수 |
| 계산식 | 기능 이해성 = {1 - (A/B)} * 100 (%) | |

표 13. 오류 메시지 이해성에 대한 평가 매트릭

| | | |
|--------|------------------------------------|----------------------------|
| 평가 매트릭 | 오류 메시지를 이해하고 처리할 수 있는가? | |
| 측정 항목 | A | 오류 메시지를 이해하지 못하여 대처하지 못한 수 |
| | B | 사례연구에서 제시한 오류 메시지 수 |
| 계산식 | 오류 메시지 이해성 = {1 - (A/B)} * 100 (%) | |

표 14. 시스템 친밀성에 대한 평가 매트릭

| | | |
|-------|---------------------------|-----------------------------------|
| 평가매트릭 | 기존의 시스템 사용 방식과 유사한가? | |
| 측정 항목 | A | 사례연구에서 기존의 시스템 사용 방식과 유사하다고 응답한 수 |
| | B | 사례연구에서 친밀성에 대한 평가 항목 수 |
| 계산식 | 시스템 친밀성 = (A/B) * 100 (%) | |

표 15. 기능 접근 용이성에 대한 평가 매트릭

| | | |
|-------|-----------------------------------|-------------------------|
| 평가매트릭 | 원하는 기능에 쉽게 접근할 수 있는가? | |
| 측정 항목 | A | 원하는 기능에 접근하지 못한 케이스 수 |
| | B | 사례연구에서 제시한 전체 테스트 케이스 수 |
| 계산식 | 기능 접근 용이성 = {1 - (A/B)} * 100 (%) | |

표 16. 인터페이스 편리성에 대한 평가 매트릭

| | | |
|-------|-----------------------------|------------------------|
| 평가매트릭 | 사용자 인터페이스가 편리하게 구성되어 있는가? | |
| 측정 항목 | A | 사용자 인터페이스가 편리하다고 응답한 수 |
| | B | 사례연구에 참여한 참가자 수 |
| 계산식 | 인터페이스 편리성 = (A/B) * 100 (%) | |

4.2.3 효율성 평가

효율성은 시스템의 자원이 적절하게 사용되고 있는지를 평가하는 항목으로서 기존의 세부 특성인 시간효율성, 자원효율성에 개발효율성을 추가한다. 웹 애플리케이션은 특정 요청에 대한 시스템의 내부적인 처리 및 응답이 실시간으로 이루어져야하기 때문에 처리 속도와 시스템 동작에 필요한 메모리, CPU의 점유율, 데이터베이스의 접근 및 데이터 조회 등이 중요한 평가 항목으로 고려되어야 한다. 표 17-20은 시간효율성과 자원효율성에 대한 평가 매트릭이다. 개발효율성은 프레임워크나 OpenAPI에서 제공

표 17. 시스템 처리 시간에 대한 평가 매트릭

| | | |
|--------|----------------------------|-------------------------|
| 평가 매트릭 | 시스템 처리가 원활하게 이루어지는가? | |
| 측정 항목 | A | 각 테스트 케이스별 수행 시간의 총합 |
| | B | 사례연구에서 제시한 전체 테스트 케이스 수 |
| 계산식 | 시스템 평균 처리 시간 = (A/B) (sec) | |

표 18. 서버-클라이언트 응답 시간에 대한 평가 메트릭

| | | |
|-------|--------------------------------|------------------------------|
| 평가메트릭 | 서버-클라이언트의 응답이 원활하게 이루어지는가? | |
| 측정 항목 | A | 각 테스트 케이스별 서버-클라이언트 응답시간의 총합 |
| | B | 사례연구에서 제시한 전체 테스트 케이스 수 |
| 계산식 | 서버-클라이언트 평균 응답시간 = (A/B) (sec) | |

표 19. 메모리 사용률에 대한 평가 메트릭

| | | |
|--------|----------------------------------|-------------------------|
| 평가 메트릭 | 시스템이 동작하기 위해 요구되는 메모리 점유율이 적절한가? | |
| 측정 항목 | A | 각 테스트에서 요구되는 메모리 사용률 총합 |
| | B | 사례연구에서 제시한 전체 테스트 케이스 수 |
| 계산식 | 메모리 사용률 = (A/B) (MB) | |

표 20. 데이터베이스 데이터 사용률에 대한 평가 메트릭

| | | |
|--------|--------------------------------|-----------------------------------|
| 평가 메트릭 | 시스템이 동작하기 위한 데이터베이스 사용률이 적절한가? | |
| 측정 항목 | A | 각 테스트에서 사용된 데이터 중 데이터베이스에 저장된 데이터 |
| | B | 사례연구에서 사용된 전체 데이터 |
| 계산식 | 데이터베이스 데이터 사용률 = (A/B) (%) | |

하는 프로그램의 기본 구조나 공개된 소스코드를 이 용함으로써 개발 생산성이 얼마나 향상되었는가를 평가하는 항목이다. 표 21, 22는 소스코드 작성의 편이성과 소스코드의 중복 회피성에 대한 평가 메트릭이며, 그 밖에도 시스템의 설계 및 개발 부담 절감성, 개발의 신속성, 개발 시 적용된 코딩 규약의 일관성, 테스트의 용이성, 오류 정보에 대한 체계적인 관리 등을 평가 항목으로 사용한다.

4.2.4 유지보수성 평가

유지보수성은 요구사항에 따라 시스템의 변경이 가능한지를 평가하며 유지보수성의 세부 특성은 분

표 21. 소스코드 작성의 편이성에 대한 평가 메트릭

| | | |
|--------|--------------------------------|------------------------|
| 평가 메트릭 | 소스코드 작성의 편이성을 제공하는가? | |
| 측정 항목 | A | 프레임워크/API에서 제공해주는 소스코드 |
| | B | 시스템 개발 시 작성된 전체 소스코드 |
| 계산식 | 소스코드 작성의 편이성 = (A/B) * 100 (%) | |

표 22. 소스코드의 중복 회피성에 대한 평가 메트릭

| | | |
|--------|--------------------------------------|----------------------|
| 평가 메트릭 | 소스코드의 중복을 피할 수 있는가? | |
| 측정 항목 | A | 시스템 개발 시 중복된 소스코드 |
| | B | 시스템 개발 시 작성된 전체 소스코드 |
| 계산식 | 소스코드의 중복 회피성 = (1 - (A/B)) * 100 (%) | |

석성, 변경성, 재사용성으로 구분된다. 분석성과 변경성은 기존의 평가 모델에서 정의한 세부 특성이며, 본 연구에서는 프레임워크의 사용에 초점을 맞추어 재사용성을 추가한다. 분석성은 표 23과 같이 오류의 발생 여부를 발견하고, 그에 대한 원인을 분석할 수 있는지를 평가하며, 사용자의 잘못된 접근이나 입력 범위 초과, 필수 입력 값의 누락과 같은 예외 상황에 대해서 대처할 수 있는지에 대해서도 평가한다.

표 23. 오류 분석 가능성에 대한 평가 메트릭

| | | |
|--------|-----------------------------------|-------------------------|
| 평가 메트릭 | 오류 발생에 대한 분석이 가능한가? | |
| 측정 항목 | A | 오류 발생에 대한 정보가 제시되지 않은 수 |
| | B | 오류 발생에 대한 전체 테스트 케이스 수 |
| 계산식 | 오류 분석 가능성 = {1 - (A/B)} * 100 (%) | |

변경성은 시스템의 설정이나 구성 요소를 변경할 수 있는 가능성을 평가하는데, 설정 값의 변경 가능성, 기능 변경성, 소스코드 명료성에 대한 평가 메트릭을 포함한다(표 24, 25).

표 24. 설정 변경 가능성에 대한 평가 메트릭

| | | |
|--------|-----------------------------------|-----------------------------------|
| 평가 메트릭 | 설정 값의 변경이 가능한가? | |
| 측정 항목 | A | 사례연구에서 설정 값의 변경이 불가능한 테스트 항목 수 |
| | B | 사례연구에서 설정 값의 변경이 요구되는 전체 테스트 항목 수 |
| 계산식 | 설정 변경 가능성 = {1 - (A/B)} * 100 (%) | |

표 25. 소스코드 명료성에 대한 평가 메트릭

| | | |
|--------|----------------------------------|---------------------------------------|
| 평가 메트릭 | 시스템의 유지보수가 용이하도록 소스코드가 명료한가? | |
| 측정 항목 | A | 중복코드 및 사용되지 않는 변수 선언 등 불필요하게 작성된 소스코드 |
| | B | z작성된 전체 소스코드 |
| 계산식 | 소스코드 명료성 = {1 - (A/B)} * 100 (%) | |

사이트메쉬와 같은 프레임워크를 이용하면 웹 페이지의 레이아웃 설정에서 공통으로 사용되는 소스 코드를 한번만 작성하여도 모든 페이지에 적용할 수 있다. 이처럼 프레임워크를 이용하면 소스코드 뿐만 아니라 클래스 컴포넌트의 재사용이 가능하기 때문에 불필요한 소스 코드의 중복을 줄일 수 있다. 소스 코드가 중복되는 경우 유지보수성이 떨어지므로 재사용성을 평가 항목으로 고려하여 변경 사항에 대해서 빠르고 효율적으로 대처할 수 있는지 평가해야 한다. 표 26, 27은 재사용성에 대한 평가 메트릭이다.

표 26. 소스코드 재사용성에 대한 평가 메트릭

| | | |
|--------|--------------------------------------|---|
| 평가 메트릭 | 시스템 기능 추가/변경 시 이전 소스코드를 사용할 수 있는가? | |
| 측정 항목 | A | 시스템 기능 추가/변경 시 이전 소스코드를 사용하지 않고 새로 작성한 소스코드 |
| | B | 시스템 기능 추가/변경 시 작성한 소스코드 |
| 계산식 | 소스코드 재사용성 = $(1 - (A/B)) * 100 (\%)$ | |

표 27. 클래스 컴포넌트 재사용성에 대한 평가 메트릭

| | | |
|--------|------------------------------------|---------------------------------|
| 평가 메트릭 | 클래스 컴포넌트의 재사용이 가능한가? | |
| 측정 항목 | A | 재사용된 클래스 컴포넌트 수 |
| | B | 각 기능을 구현하기 위해 필요한 전체 클래스 컴포넌트 수 |
| 계산식 | 클래스 컴포넌트 재사용성 = $(A/B) * 100 (\%)$ | |

4.2.5 안정성 평가

안정성은 본 연구에서 새롭게 정의한 품질 특성으로서 시스템이 안정적으로 동작하는가에 대해서 평가하며 호환성, 일관성, 준수성의 세 가지 세부 특성을 정의한다. 웹 애플리케이션은 웹 서버, 데이터베이스, 클라이언트로 구성되며 상호 간에 정보를 주고받으면서 동작하기 때문에 각 요소 간의 호환성과 안정적인 데이터 전송이 중요하다. 프레임워크를 이용하여 웹 애플리케이션을 개발하는 경우, 프레임워크에서 제공하는 최적화된 소스코드를 바탕으로 구현할 수 있기 때문에 보다 안정적인 시스템 개발이 가능하다. 특히, 스프링과 같은 경량 컨테이너 기반의 프레임워크는 단위 테스트가 용이하다는 이점이 있어서 지속적인 테스트를 통하여 웹 애플리케이션의 품질 향상을 도모할 수 있다. 또한, 스프링은 자바

를 이용하여 작성된 기존의 프레임워크와의 통합을 지원하기 때문에 개발 중에 다른 프레임워크로 대체하는 것이 가능하다. 프레임워크 및 시스템 구성 요소들 간의 호환성을 평가하기 위하여 다양한 프레임워크 및 상용도구, OpenAPI와의 연계가 가능한지, 웹 서버와 클라이언트, 웹 서버와 데이터베이스 간의 정보 처리가 정상적으로 이루어지는지 점검한다. 호환성에 대한 평가 메트릭은 표 28-30과 같다.

표 28. 프레임워크 간의 호환성에 대한 평가 메트릭

| | | |
|--------|--------------------------------------|--|
| 평가 메트릭 | 연동된 프레임워크들이 호환되는가? | |
| 측정 항목 | 연동된 프레임워크들 간의 정보 처리가 정상적으로 이루어지는지 점검 | |

표 29. 웹서버-클라이언트 간의 호환성에 대한 평가 메트릭

| | | |
|--------|--|--|
| 평가 메트릭 | 웹서버-클라이언트 간에 안정적으로 정보를 주고 받는가? | |
| 측정 항목 | A | 사례연구에서 웹서버-클라이언트의 접속 및 정보처리에 실패한 테스트 케이스 수 |
| | B | 사례연구에서 웹서버-클라이언트 간에 정보를 주고받는 전체 테스트 케이스 수 |
| 계산식 | 웹서버-클라이언트 호환성 = $(1 - (A/B)) * 100 (\%)$ | |

표 30. 웹서버-데이터베이스 간의 호환성에 대한 평가 메트릭

| | | |
|--------|---|---|
| 평가 메트릭 | 웹서버-데이터베이스 간에 안정적으로 정보를 주고 받는가? | |
| 측정 항목 | A | 사례연구에서 웹서버-데이터베이스의 접속 및 정보처리에 실패한 테스트 케이스 수 |
| | B | 사례연구에서 웹서버-데이터베이스 간에 정보를 주고받는 전체 테스트 케이스 수 |
| 계산식 | 웹서버-데이터베이스 호환성 = $(1 - (A/B)) * 100 (\%)$ | |

일관성은 시스템의 개발 방식과 동작에서 일관성을 유지하는가를 평가한다. 스프링 프레임워크는 MVC 패턴에 따라서 일관된 방향으로 애플리케이션을 개발하도록 주도하며, 인터페이스를 구현하는 하위 클래스들과 비즈니스 계층 간의 의존성을 낮춤으로써 하위 클래스를 수정하여도 비즈니스 계층에 영향을 주지 않는다. 일관성을 평가하기 위하여 표 31, 32에서 설명하는 시스템 개발 일관성과 시스템 동작 일관성 외에도 기존 시스템에 특정 기능을 추가, 변경, 삭제하여도 전체 시스템에 미치는 영향을 최소화

표 31. 시스템 개발 일관성에 대한 평가 매트릭

| | | |
|--------|------------------------------|------------------------|
| 평가 매트릭 | 시스템이 일관성 있게 개발되었는가? | |
| 측정 항목 | A | MVC 패턴을 따라 작성된 클래스 파일 |
| | B | 시스템 개발 시 작성된 전체 클래스 파일 |
| 계산식 | 시스템 개발 일관성 = (A/B) * 100 (%) | |

표 32. 시스템 동작 일관성에 대한 평가 매트릭

| | | |
|--------|------------------------------------|--|
| 평가 매트릭 | 시스템의 기능이 추가/변경/삭제되어도 일관성 있게 동작하는가? | |
| 측정 항목 | A | 시스템의 기능을 추가/변경/삭제 시 일관성 있게 동작하지 않는 기능의 수 |
| | B | 시스템이 제공하는 전체 기능 수 |
| 계산식 | 시스템 동작 일관성 = {1 - (A/B)} * 100 (%) | |

시킬 수 있는지에 대해서 종합적으로 평가한다.

준수성은 웹 애플리케이션 개발에 있어서 지켜져야 할 표준과 규약을 준수하고 있는지 평가하는 항목으로서 HTML, XHTML, CSS, JSTL 등의 웹 표준에 따라 소스코드가 작성되어 있는지, 시스템에서 요구되는 데이터 표준에 따라서 사용자 입력이 이루어지는지, 그리고 보안 정책을 준수하는지에 대한 평가 매트릭을 포함한다(표 33, 34). 시스템에서 요구되는 데이터 표준에 부적합한 데이터가 입력되는 경우, 예상치 못한 오류나 시스템 다운으로 연결될 수 있으므로 사용자 입력의 준수성을 중요한 평가 항목으로 고려해야 한다.

표 33. 소스코드 표준 준수성에 대한 평가 매트릭

| | | |
|--------|-------------------------------------|----------------------|
| 평가 매트릭 | 소스코드가 웹 표준에 따라서 작성되었는가? | |
| 측정 항목 | A | 웹 표준에 위배되는 소스코드 |
| | B | 시스템 개발 시 작성된 전체 소스코드 |
| 계산식 | 소스코드 표준 준수성 = (1 - (A/B)) * 100 (%) | |

표 34. 사용자 입력 준수성에 대한 평가 매트릭

| | | |
|--------|---|---|
| 평가 매트릭 | 시스템에서 요구되는 데이터 표준에 따라서 사용자의 입력이 이루어지는가? | |
| 측정 항목 | A | 사례연구에서 데이터 표준에 부적합한 입력에 대해 경고 없이 무시된 경우 |
| | B | 사례연구에서 사용자 입력이 요구되는 전체 테스트 케이스 수 |
| 계산식 | 사용자 입력 준수성 = (1 - (A/B)) * 100 (%) | |

4.3 시스템 품질 평가 수행 및 결과

본 연구에서 제시한 품질 평가 모델과 평가 매트릭에 따라서 프레임워크 기반 온라인 서재 시스템의 품질 평가를 수행한다. 시스템의 품질 평가는 웹 애플리케이션의 사용 빈도가 높은 34명의 컴퓨터 관련 전공자들을 대상으로 진행하며, 회원 가입, 로그인, 도서 검색, 서재 관리, 리뷰 및 별점 등록 등 구현된 총 64개의 기능을 종합적으로 사용하는 시나리오를 제시한다. 참가자들은 시나리오에 따라서 일정 시간 동안 시스템을 사용하며 사용성, 기능성, 효율성에 해당하는 총 48개의 항목에 대해서 평가한다. 또한, 웹 애플리케이션을 사용할 때 자주 발생하는 오류들을 취합하여 작성한 30개 항목의 오류 체크 리스트에서 시스템 품질 평가 수행 시 발생하는 오류를 점검하도록 함으로써 온라인 서재 시스템이 정확하게 동작하는지에 대해서도 평가한다.

사용자들의 시스템 품질 평가 결과를 정량적으로 나타내기 위하여 각각의 평가 항목에 적합한 평가 매트릭을 대응시켜서 측정값을 계산한다. 표 35는 시스템 품질 평가에서 이용한 오류 항목들과 대응되는 평가 매트릭들을 나타낸 것으로서 오류 항목은 사용자들이 오류 체크 리스트에서 체크한 항목을 기준으로 한다.

유지보수성은 사용자 평가를 통하여 측정할 수 없는 품질 특성이다. 따라서 유지보수성을 평가하기 위하여 기존의 시스템에 Quartz 프레임워크를 이용한 스케줄 관리와 다른 사용자의 리뷰에 공감하기 등의 크고 작은 기능 5가지를 추가하여 시스템의 확장이 용이한지, 그리고 이전에 작성한 소스코드 및 클래스 컴포넌트를 재사용할 수 있는지 평가한다.

표 36은 프레임워크 기반 온라인 시스템에 대한 사용자들의 품질 평가 결과를 정량적으로 나타낸 결과표이다. 각 품질 특성에 대해서 평가 점수를 합산하여 시스템의 전체 품질 평가 결과를 도출할 수 있다. 본 연구에서 제안하는 프레임워크 기반의 온라인 서재 시스템의 품질을 평가한 결과, 기능성 98.7%, 사용성 94%, 유지보수성 97.1%, 안정성 99.4%로 나타났으며 전체 평가 점수의 평균은 97.3%로 대체적으로 품질이 우수한 것으로 확인되었다(그림 7). 또한, 평균 0.54초의 빠른 처리 속도와 2~7%의 낮은 CPU 점유율을 보였으며, 특정 작업을 수행하더라도 메모리의 사용률이 크게 증가하지 않는 것으로 나타났다.

표 35. 시스템 품질 평가에서 이용한 오류 항목과 대응되는 평가 매트릭

| No | 오류 항목 | 평가 매트릭 |
|----|--------------------------------|---|
| 1 | 사이트에 접속되지 않는다. | 접근 통제성/접근 감시성, 오류 메시지 이해성, 웹서버-클라이언트 평균 응답 시간, 오류 분석 가능성, 웹서버-클라이언트 호환성 |
| 2 | 링크 오류가 발생하여 원하는 페이지로 이동할 수 없다. | 기능 구현 완전성/기능 구현 정확성/결과 데이터의 정확성/기능 적용의 정확성, 오류 메시지 이해성/기능 접근 용이성, 웹서버-클라이언트 응답시간/웹서버-DB 응답시간, 오류 분석 가능성 |
| 3 | 이미지가 깨져서 정상적으로 보이지 않는다. | 결과 데이터의 정확성/데이터 일치성, 웹서버-클라이언트 응답시간/웹서버-DB 응답시간, 웹서버-클라이언트-DB 호환성 |
| 4 | 사이트가 다운된다. | 결과 데이터의 정확성/데이터 일치성, 시스템 처리 평균 시간/서버-클라이언트 응답시간/메모리 사용률/CPU 사용률/DB 사용률, 오류 분석 가능성, 프레임워크 간의 호환성/웹서버-클라이언트-DB 호환성/사용자 입력 준수성 |
| 5 | 원하는 기능에 해당하는 메뉴를 찾을 수 없다. | 기능 구성 적합성/기능 적용의 정확성, 기능 이해성/시스템 친밀성/시스템 흥미도/기능 접근 용이성/인터페이스 편리성 |

표 36. 프레임워크 기반 온라인 서재 시스템의 평가 결과표품질특성

| 품질특성 | 평가항목 | 측정결과 | 품질특성 | 평가항목 | 측정결과 |
|------|--------------|-------------------|------------|---------------|-------|
| 기능성 | 구현 완전성 | 98.9% | 사용성 | 기능/인터페이스 이해성 | 95% |
| | 구성 적합성 | 97.4% | | 오류 이해성 | 96.8% |
| | 데이터 정확성 | 95% | | 시스템 친밀성 | 93.6% |
| | 데이터 일치성 | 100% | | 시스템 흥미도 | 92.5% |
| | 적용 정확성 | 98.5% | | 기능 접근 용이성 | 92.6% |
| | 접근 통제성 | 100% | | 인터페이스 편리성 | 93.5% |
| | 접근 감시성 | 100% | | 오류 분석 가능성 | 98.7% |
| 효율성 | 데이터 보안성 | 100% | 유지 보수성 | 설정 변경 가능성 | 100% |
| | 시스템 처리시간 | 평균 0.54초 | | 소스코드 명료성 | 96.2% |
| | 구성요소 간 응답 시간 | 평균 0.21초 | | 소스코드/클래스 재사용성 | 93.5% |
| | 메모리 사용률 | 843~5MB | 안정성 | 프레임워크 호환성 | Y(제공) |
| | CPU 사용률 | CPU 2~7% | | 구성요소 간 호환성 | 98.8% |
| | DB 데이터 사용률 | 낮음(37.4%) | | 시스템 개발 일관성 | 100% |
| | 소스코드 작성 편리성 | 높음(47.3%의 기본코드제공) | | 시스템 동작 일관성 | 100% |
| | 소스코드 중복 회피성 | 87.3% | | 소스코드 표준 준수성 | 100% |
| | | | 사용자 입력 준수성 | 98.3% | |

5. 결론 및 향후 연구

본 논문에서는 다양한 도서 정보를 공유할 수 있는 온라인 서재 시스템을 설계하고 MVC 기반의 다양한 프레임워크를 이용하여 구현하였다. 프레임워크를 이용함으로써 개발 비용을 줄이는 동시에 시스템의 확장과 유지보수에 유연하게 대처할 수 있도록 한다. 또한, 시스템의 보안성을 높이기 위하여 웹 애플리케이션에서 발생할 수 있는 XSS 공격에 대비하고, 정적 소스분석도구를 이용한 디버깅을 통하여 시

스템의 결함을 최소화시키고자 하였다.

제안하는 온라인 서재 시스템의 품질을 객관적이고 정량적으로 평가하기 위하여 프레임워크 기반의 웹 애플리케이션에 적합한 품질 평가 모델을 제시하였다. 기존의 ISO/IEC 9126-2 평가 모델을 기반으로 기능성, 사용성, 효율성, 유지보수성, 안정성의 평가 항목을 정의하고, 프레임워크 기반의 웹 애플리케이션에 특화된 평가 매트릭을 제시하였다. 프레임워크를 사용함으로써 기대할 수 있는 소스코드 작성의 편리성, 중복 회피성, 재사용성과 프레임워크 간의

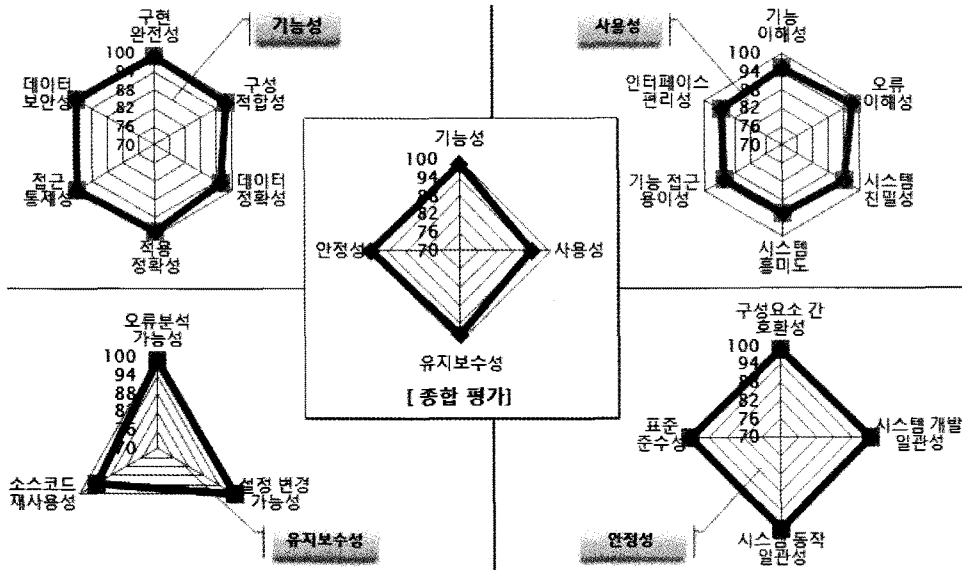


그림 7. 프레임워크 기반 온라인 서재 시스템의 평가 결과 그래프

호환성, 개발 방식의 일관성 등에 초점을 맞추어 평가 메트릭을 설계하고 프레임워크 기반의 온라인 서재 시스템에 대한 사용자 평가를 수행하였다. 그 결과, 다섯 가지의 평가 항목에서 모두 우수한 결과를 보였으며, 특히, 기능성의 보안성 측면과 안정성의 일관성 측면에서 높이 평가되었다.

향후 프레임워크 기반의 웹 애플리케이션에 특화된 품질 평가 모델 연구가 체계적이고 지속적으로 진행되어야 할 것이며, 이러한 연구를 통하여 보다 높은 품질의 웹 애플리케이션을 제공할 수 있는 기반을 마련할 수 있을 것으로 기대한다.

참 고 문 헌

[1] S. Burbeck, "Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller," <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>, 1987.

[2] J. Arthur and S. Azadegan, "Spring Framework for Rapid Open Source J2EE Web Application Development: A case study," IEEE AGIS Conference, pp. 90-95, 2005.

[3] 송지훈, 이시진, "오픈소스 소프트웨어 특성을 고려한 소프트웨어 평가모델," 한국인터넷정보학회 학술발표대회 논문집, 제9권, 제1호, pp.

347-352, 2008.

[4] Software Engineering-Product Quality-Part 1: Quality Model, ISO/IEC, 2001.

[5] Software Engineering-Product Quality-Part 2: External Metrics, ISO/IEC TR 9126-2, 2003.

[6] Software Engineering-Product Quality-Part 3: Internal Metrics, ISO/IEC TR 9126-3, 2003.

[7] Software Engineering-Product Quality-Part 4: Quality in use metrics, ISO/IEC TR 9126-4, 2003.

[8] S. Mavromoustakos and A. S. Andreou, "WAQE: a Web Application Quality Evaluation model," *International Journal of Web Engineering and Technology*, Vol.3, No.1, pp. 96-120, 2007.

[9] Mark Tomita, "Administration Design Quality Web site Evaluation Method," <http://steinhardt.nyu.edu/hepr/resources/online/adq.pdf>.

[10] L. O. Santos, "Web-site Quality Evaluation Method: a Case Study on Museums," ICSE 2nd Workshop on Software Engineering over the Internet, 1999.

[11] 박선영, 한효섭, 정기원, "웹 서비스 테스트 단계에서의 기능성 품질평가 모델," 한국인터넷정보

- 학회 학술발표대회 논문집, pp. 469-472, 2009.
- [12] 김용원, "사용자 측면을 고려한 웹 서비스의 품질 평가 모델 설계," 한국정보기술학회 논문지, 제6권, 제5호, pp. 138-144, 2008.
- [13] 이종민, "보안소프트웨어 제품을 위한 평가 매트릭스 연구," 한국정보과학회 가을 학술발표 논문집, 제33권, 제2호, pp. 427-432, 2006.
- [14] N. Ayewah, W. Pugh, J. D. Morgenthaler, J. Penix, and Y. Zhou, "Evaluating Static Analysis Defect Warning On Production Software," ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering, pp.1-8, 2007.
- [15] <http://findbugs.sourceforge.net>
- [16] Web Application Security statistics 2008, <http://www.projects.webappsec.org>



박 윤 영

2006년 성신여자대학교 미디어정보학부 학사
2006년 (주)TG CORP 기술개발팀 사원
2010년 성신여자대학교 일반대학원 전산학과 이학석사

관심분야: Computer Graphics, Crowd Simulation, Game, HCI, User Perception, Web Application



변 혜 원

1990년 연세대학교 전산학과 이학사
1992년 KAIST 전산학과 공학석사
2004년 KAIST 전산학과 공학박사
2004년 KBS 방송기술연구소 선임연구원
2006년~현재 성신여자대학교 IT학부 교수

관심분야: Computer Graphics, Virtual Reality, HCI, Game, User Perception