

## 소프트웨어 프로세스 내 부산물을 통한 다양한 산출 보고 문서 생성

<sup>1</sup>문소영, <sup>\*2</sup>김영철

<sup>1</sup>홍익대학교 소프트웨어공학연구실, [msy@selab.hongik.ac.kr](mailto:msy@selab.hongik.ac.kr)

<sup>\*2</sup>홍익대학교 소프트웨어공학연구실, [bob@hongik.ac.kr](mailto:bob@hongik.ac.kr)

### Generating Diverse Report Documents with Byproducts on Software Process

<sup>1</sup>So Young Moon, <sup>\*2</sup>R. Young Chul Kim

<sup>1</sup>SELab., Sejong Campus, Hongik University, [msy@selab.hongik.ac.kr](mailto:msy@selab.hongik.ac.kr)

<sup>\*2</sup>. SELab., Sejong Campus, Hongik University, [bob@hongik.ac.kr](mailto:bob@hongik.ac.kr)

#### 요약

소프트웨어의 중요성이 증가하며 소프트웨어 제품 품질과 소프트웨어 프로세스 품질에 대한 이슈도 높아지고 있다. 국내에서는 NIPA의 SP 인증 제도가 국내 중소/벤처 기업의 소프트웨어 프로세스 품질 향상을 위한 지원과 이를 보증하고 있다. 특히 국내 소프트웨어 개발 업계가 어려움을 겪고 있는 것 중의 하나가 소프트웨어 개발 프로세스에 맞춰 일할 수 있는 1)시간과 비용이 충분하지 않다는 것, 2)각종 산출물 작성에 투자할 인원부족, 3)잦은 요구사항 변경 등이 있다. 본 논문에서는 국내 중소/벤처 기업의 어려움을 해소하고자, 소프트웨어 개발 주기 전체에서 산출되는 문서를 자동화하고 체계적인 프로세스 관리를 통해 소프트웨어 품질을 향상시키는 방안을 제시한다. 또한 개발한 시스템을 통해 개발자들이 잦은 요구사항 변경에 유연한 대처를 하고, 개발자가 다양한 보고 문서들을 작성하는데 효율적 시간 투자가 가능하고, 더 많은 시간을 구현에 투자할 것으로 기대된다.

#### Abstract

*As the importance of software increases, issues on software quality and software process quality are also growing up. In Korea, Software Process Certification (SP Certification) system of NIPA helps to improve quality of software processes, and guarantees it for Korean small/venture companies. In particular, Korean software development industries will have been experiencing difficulties, including: 1) insufficient time and cost to work according to the software development process, 2) lack of personnel to invest in creating output, and 3) frequent changes in requirements. This paper proposes a method to improve the software quality by automatically creating diverse report documents being generated and managing the processes systematically during the whole cycle of software development in order to resolve the difficulties of Korean small, medium-sized (SMEs) and venture enterprises. In addition, it is expected that this system proposed and developed in this paper enables developers flexibly to cope with frequent changes in requirements, to be able to spend efficient time writing report documents and to spend more time implementing software.*

**Keywords:** Software Process Quality, Document Artifact, Improvement of Software Process Quality, Automatic Generation System for Document Artifact, Software Process Tool-chain System

---

\* Corresponding Author

Received: Nov. 15, 2017, Revised: Dec. 4, 2017, Accepted: Dec. 21, 2017

## 1. 서론

소프트웨어 개발은 잦은 요구사항 변경과 요구사항의 불확실성으로 예산이 초과되고 납기가 지연되어 소프트웨어의 품질 저하가 불가피하다. 건축이나 제조업과 같은 하드웨어 분야에서는 개발과정 중 발생하는 변경사항에 관련없이 계획대로 개발을 진행하지만, 소프트웨어 개발의 경우 잦은 요구사항 변경이 발생하고 이로 인해 예산 초과와 납기 지연이 발생하고 기대 이하의 품질을 갖게 된다. 특히 중소·벤처 기업의 경우 개발을 하면서 문서작성까지 완벽하게 소화한다는 것은 현실적으로 어려움이 많다. 소프트웨어의 중요성이 높아지면서 품질에 대한 관심도 함께 증가하고 있다. 소프트웨어의 품질 측정 및 평가를 위한 국제 품질 기준으로 가장 많이 사용되는 모델은 CMMI, SPICE 가 있고, 국내에서는 GS 인증[1]과 SP 인증[2]이 중소·벤처 기업의 소프트웨어 품질 인증을 돕고 있다. SP 인증은 소프트웨어 기업 및 개발 조직의 프로젝트 수행 역량 수준을 평가하고 개선하는 것이다. SP 인증 심사는 프로젝트 관리, 프로세스 개선, 개발, 조직관리, 지원 영역으로 구성되어 각 영역에서 생산되는 산출물들을 검토하고 인터뷰를 통해 품질 인증을 한다.

본 논문에서는 소프트웨어 개발 주기 전체에서 산출되는 문서를 자동화하고 체계적인 프로세스 관리를 통해 소프트웨어 품질을 향상시키는 방안을 제안한다. 요구사항 변경에 유연한 대처와 문서 자동화를 통해 개발자를 돕고 이는 소프트웨어 품질향상과 직결한다. 개발자들은 요구사항 변경에 영향을 받는 소스코드를 쉽게 알 수 있고, 관리자는 요구사항 변경이 전체 시스템에 미치는 영향을 파악할 수 있다. 또한 테스터들은 요구사항 변경에 영향 받는 테스트케이스, 테스트 시나리오들을 쉽게 수정할 수 있다.

본 논문의 2 장은 관련 연구로 소프트웨어 품질과 소프트웨어 문서화에 대해 설명하고, 3 장에서는 산출물 자동 생성에 대한 과정과 방법에 대해 기술한다. 그리고 마지막으로 4 장에서는 결론 및 향후 연구에 대해 서술한다.

## 2. 관련 연구

### 2.1. 소프트웨어 품질

IEEE [3]가 정의하는 소프트웨어 품질은 다음과 같다.

- 주어진 요구 사항을 충족시킬 수 있는 능력을 갖춘 소프트웨어의 총체적인 특징과 제품의 특성들.
- 소프트웨어가 요구하는 특성이나 속성들을 조합할 수 있는 소프트웨어 프로세스의 정도.
- 소프트웨어가 고객 또는 사용자의 기대치를 충족시키고, 이것을 인식할 수 있는 정도.
- 사용중인 소프트웨어가 고객의 기대를 충족시키는 정도를 결정하는 소프트웨어의 복합적 특성.

또한 IEEE에서는 소프트웨어 제품은 소스코드, 문서로 구성된다고 한다.

ISO 9126 [4]에서 정의하는 소프트웨어 품질을 다음과 같다.

- 고객이 만족해야 한다.
- 최종 소프트웨어가 어떠한 오류도 없이 소프트웨어 요구사항에 맞게 개발되어야 한다.

### 2.2. 소프트웨어 문서 산출물

드웨인 필립[5]은 사람(people), 프로세스(process), 제품(product) 이 소프트웨어 프로젝트에 많은 영향을 미친다고 했다. 소프트웨어는 눈에 보이지 않는 특성으로 안정적이며 지속적으로 동일한 수준의 품질을 보장할 수 없다. 근래에는 소프트웨어의 품질 보장을 위해 소프트웨어 개발 프로세스에 대한 체계화 및 정형화를 통해 지속적으로 동일한 품질을 보장하고, 프로젝트의 성공률을 높여 고품질의 소프트웨어 생산이 가능하다[6].

소프트웨어 개발 프로세스 개선과 향상은 소프트웨어 품질과 연관이 있다. 국내 방위사업청의 <무기체계 소프트웨어 개발 및 관리 지침>에는 소프트웨어 개발 및 관리를 위해 필요한 산출물 표준 서식 정하고 있고, 한국소프트웨어진흥원에서는 SP 인증을 위해 필요한 프로세스 가이드라인을 제공하고, SP 인증 심사에 필요한 산출물 종류를 제시하고 있다[7].

### 3. 소프트웨어

기업의 산출물 관리, 발주자의 요구사항, 기타 인증 및 품질관리를 위해 개발과정의 산출물은 문서형태로 작성되어야 한다. 하지만 문서 산출물은 개발과정에서의 부담으로 작용한다. 소프트웨어 품질 향상을 위해 소프트웨어 개발 프로세스에 준수하여 소프트웨어의 개발 효율성과 소프트웨어 개발과정의 추적성을 확보하기 위해서는 개발 전반의 활동을 시스템화하여 시스템과 연계하는 것이 매우 중요하다[8]. 본 논문에서는 소프트웨어 개발 과정에서 발생하는 문서 산출물 자동생성 시스템을 제안하고 설명한다.

#### 3.1. 소프트웨어 개발 프로세스를 위한 시스템 툴-체인 구조

개발한 시스템은 레드마인(Redmine), 젠킨스(Jenkins), 서브버전(Subversion), SQLite, PMD, MySQL, Apache, 그리고 Tomcat 과 같은 오픈소스로 구성된다. 레드마인은 요구사항 관리, 이슈 관리를 하고, 젠킨스 매일 매일 자동 빌드를 담당한다. 서브버전은 형상관리와 버전 컨트롤을 한다. SQLite와 MySQL은 데이터베이스 시스템이다. Apache와 Tomcat은 웹 어플리케이션 서버로 사용된다.

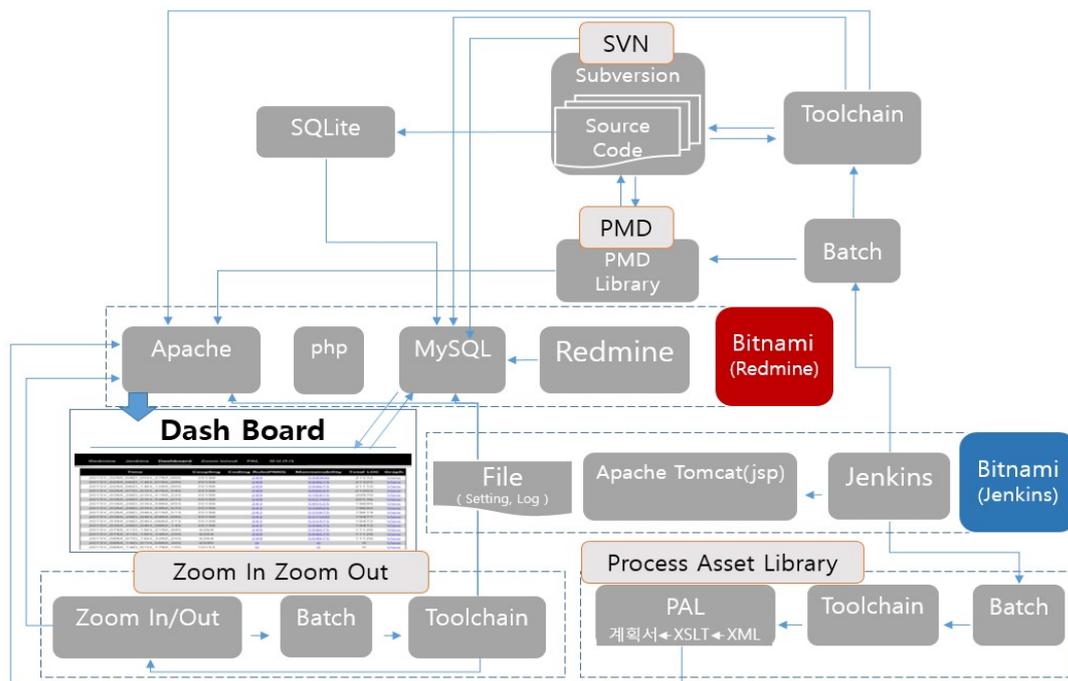


Figure 1. System Structure for Software Process Tool-Chain

“그림 1”을 설명하면, 레드마인을 통해 요구사항 정의서, 유스케이스, 기능적 요구사항, 클래스와 메서드 정의서, 테스트 시나리오, 그리고 테스트 케이스를 생성 가능하다. PAL(Project Asset Library) 시스템은 프로젝트 자산 라이브러리 시스템으로 산출물의

형식을 관리한다. Tool-Chain 은 기존 연구에서 제안했다[9, 10]. 사용자는 요구사항, 이슈들을 레드마인에 입력한다. MySQL 은 레드마인의 데이터를 관리한다. 사용자는 개발된 소스코드를 우리 시스템에 업로드하면 툴체인(tool-chain)을 통해 소스코드 분석을 하고, 그 결과는 SQLite 에 저장한다.

### 3.2. 소프트웨어 문서 산출물 자동 생성

#### 3.2.1. 요구사항 정의서

레드마인에 요구사항 스펙과 소스 코드가 입력되고 각 정보가 데이터베이스에 저장된다. 모든 개발 문서는 상위 단계 문서에 대한 식별코드를 갖는다. “그림 2”는 REQ-F-001 요구사항이다. ①REQ-F-001 은 요구사항 아이디이다. ②실행 옵션을 위한 설정은 요구사항의 이름이다. ③ 요구사항 우선순위는 2 이다. ④ 설명은 REQ-F-001 에 대한 상세 요구사항이다. “그림 2”의 하단 표는 생성된 요구사항 정의서의 일부분 이다.

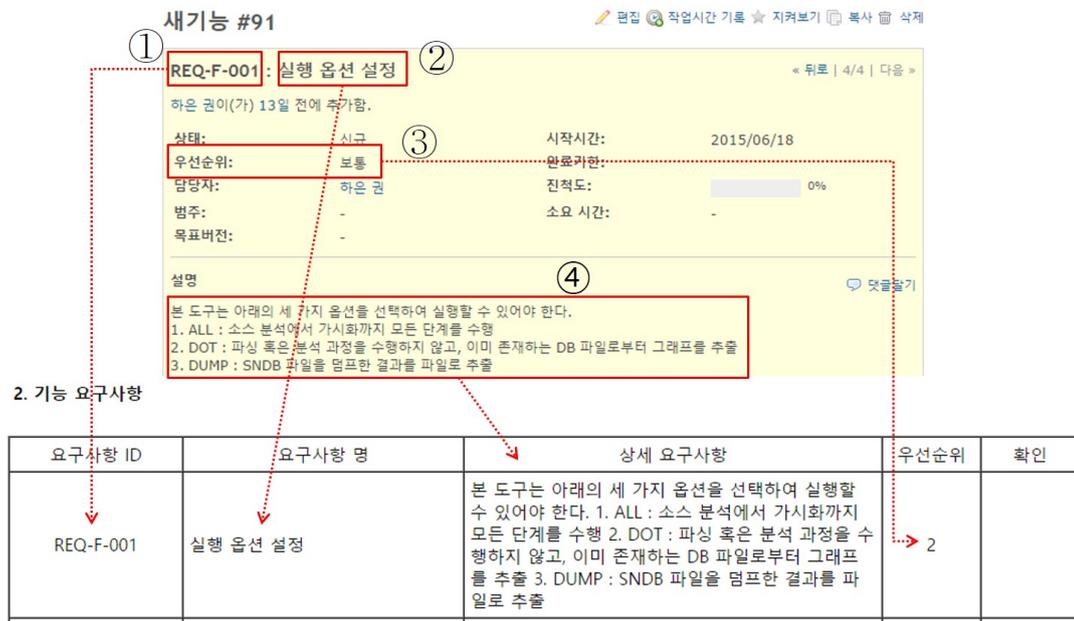


Figure 2. Example of Requirement in Redmine

#### 3.2.2. 소프트웨어 구조 분석서

소프트웨어 구조 분석서는 소프트웨어의 구조를 설계한 후 소프트웨어의 전반적인 구조와 해당 개발과정에서의 소프트웨어 품질 지표를 보여준다. 그리고 잘못된 소프트웨어의 구조를 파악하고 소프트웨어 품질을 평가할 수 있는 문서이다. 소프트웨어 구조 분석서의 목차는 다음과 같다.

- 구조 분석 날짜
- 부서 이름
- 구조 분석을 위한 다이어그램 (Class Diagram, Architecture View, etc.)
- 변경사항
- 평가 (표준 준수여부, 품질속석평가, 기타의견)

소프트웨어 구조분석서를 통해 개발과정의 추적성을 간편화 할 뿐 아니라 문서 하나만으로 소프트웨어의 전반적인 구조를 볼 수 있고 소프트웨어에 대한 평가에 대해 개발자와 이해관계자들이 쉽게 파악할 수 있는 장점이 있다.

소프트웨어 구조 분석서는 이전 연구, 소스코드 가시화[10] 방법으로 추출할 수 있던 정보를 개발한 시스템을 통해 문서화 하였다. 그리고 시스템에서 추출할 수 없는 변경사항, 기타 의견 같은 정보는 레드마인의 변경 이력 게시판에 패턴분석을 할 수 있도록 정해진 틀을 사용하여 입력한다. “그림 3”은 개발된 시스템을 통해 생성된 소프트웨어 구조 분석서이다.

#### 4. 결론

본 논문에서는 소프트웨어 개발 주기 전체에서 산출되는 문서 자동화 방법을 제안하였다. 그리고 요구사항 정의서, 소프트웨어 구조 분석서와 같은 문서 산출물을 자동 생성하는 시스템을 개발하였다. 본 연구에서 개발한 시스템은 효율적인 프로젝트 관리 및 산출물 생성에 비용과 시간을 단축시켰다. 소프트웨어 개발을 하고 그 이후에 그와 관련된 문서 작업을 직접 해야했다면 논문에서 제안한 시스템을 통해 더 편리하고 빠르게 문서를 생성할 수 있었다. 또한 일감 등록과 소스코드 관리를 통해 요구사항 정의서와 요구사항 추적 매트릭스 문서를 자동 생성하여 요구사항 추적성 확보와 문서들 간에 추적성 확보가 가능하게 되었다. 또한 개발 과정에서 요구사항에 해당하는 소스 코드와의 연관관계까지 확인할 수 있어 코드 가시화의 효과도 얻는 장점이 있다. 그리고 소프트웨어 개발은 잦은 요구사항 변경과 요구사항의 불확실성으로 예산이 초과되고 납기가 지연되어 소프트웨어의 품질 저하가 불가피했던 것이 이력 관리와 요구사항 추적이 가능해져 해소될 것이다. 이와 같은 문서 자동 생성을 통해 국내 소규모 소프트웨어 개발 업체의 프로젝트 산출물 관리에 도움을 줄 것으로 기대한다.

향후 연구로는 SP 인증과 관련한 더 많은 문서를 자동 생성하고, 소프트웨어 개발자들도와 소프트웨어 품질 확보에 기여하는 시스템으로 발전시키는 것이다.

#### 5. 감사의 글

이 논문(저서)은 2015 년 교육부와 한국연구재단의 지역혁신창의인력양성사업(NRF-2015H1C1A1035548)과 2017 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A3B03035421).

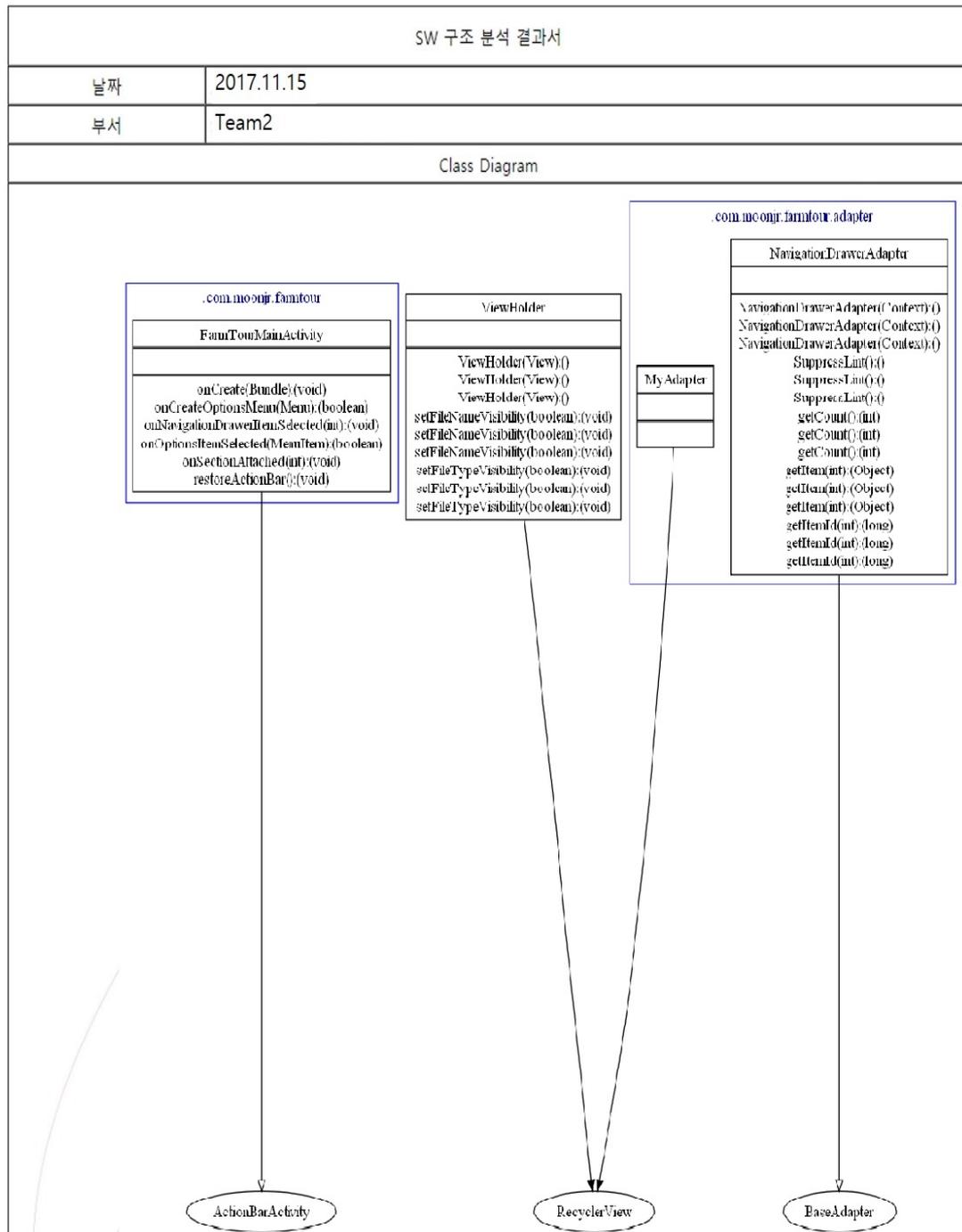


Figure 3. Software Construction Analysis

## 6. 참고문헌

- [1] <http://sw.tta.or.kr>
- [2] <http://www.sw-eng.kr>
- [3] IEEE, "Software Engineering Standards: ANSI/IEEE Std 729-1983, Glossary of Software Engineering Terminology", IEEE Computer Society, June 1983.
- [4] ISO/IEC, "ISO/IEC 9126-1 Software engineering-product quality-part 1: quality model", ISO/IEC Technical Report, 2001
- [5] Dwayne Phillips, "People, Process, and Product", American Programmer, 1995.
- [6] LeeEunJung, KoByeongSun, "Improvement of Software Quality and Process Qaulity", NIPA, 2006.
- [7] NIPA, "SP 인증을 중심으로 한 소프트웨어 프로세스", NIPA.
- [8] NIPA SW Engineering Center, "SW Development Quality Management Manual (SW Visualization)", 2013.
- [9] So Young Moon, R. Youngchul Kim, Chae Yun Seo, "A Survey for Software Visualization", Journal of Platform Technology, vol.4, No.3, p.22-29, September 2016.
- [10] So Young Moon, R. Youngchul Kim, "Code Structure Visualization with A Tool-Chain Method", Research India Publications, vol. 10, No.90, p.287-291, December 2015.