

오픈소스 형상도구와 결합추적도구를 이용한 개발 프로젝트 관리에 관한 연구

이지현*, 박영식*

*한국전자통신연구원 BigData 소프트웨어연구소 품질보증연구팀
e-mail : {jihyun, ysp}@etri.re.kr

A Study on Development Project Management using Open Source Configuration Management and Defect Tracking Tools

Jihyun Lee*, Youngsik Park*

*Quality Assurance Research Team, BigData Software Research Laboratory, ETRI

요 약

팀으로 구성된 개발 조직은 주어진 시간 안에 가용할 수 있는 인력으로 고객에게 전달될 최종 산출물인 소프트웨어를 개발한다. 개발과 통합을 거친 소프트웨어는 테스트를 거치며 발생하는 결함에 대해 수정 조치를 한다. 하지만 결함 수정과 병행하여 초기 요구사항은 변경되고 추가되어 개발되는 소프트웨어에 반영되어야 함으로 팀에서는 메일, 전화, 또는 게시판 방식으로 결함 사항을 알리고 최종 소프트웨어를 개발해 나간다. 본 논문은 이러한 환경에 결합추적도구를 사용하여 결함 발견 시 등록하고, 해결된 소프트웨어의 버전을 관리하기 위한 방법에 대해 오픈소스 기반의 형상관리도구와 결합추적도구를 활용한 환경을 연구해 나가고자 한다.

1. 서론

소프트웨어를 개발하는 프로젝트에서 모듈 개발 및 통합, 테스트의 과정을 거치면서 발생하는 문제점들은 개발자 한 명이 수정해야 하는 경우와 여러 팀원이 함께 확인하고 수정해야 하는 경우가 있다.

예를 들어 A 제품, B 제품, C 제품을 개발할 때 결함이 수정된 코드는 각각의 제품에 반영되어야 하고, 반영 후 확인 테스트를 거쳐야 한다. 이 때 각 제품의 결함이 수정되기 전과 수정된 후의 버전은 서로 다른 버전으로 관리되어야 하고 필요하다면 원하는 이력의 버전으로 돌아가 코드를 확인할 수 있어야 한다.

이와 같은 버전 관리는 체계적으로 수행될 필요가 있다. 프로세스를 체계적으로 관리하는 것은 버전에 대한 형상을 지속적으로 등록 및 변경, 제어함을 뜻한다. 따라서 각각의 제품 코드와 수정된 코드의 버전은 관리될 수 있어야 한다.

본 논문은 프로세스에 따라 개발 코드를 관리하고 소프트웨어 코드에서 발생하는 문제를 등록하고 처리 상태를 표현하며 추적하기 위한 환경 구축에 대해 소개하고 효과성에 대해 설명하고자 한다.

본 논문의 구성은 2 장은 형상관리 및 도구 활용에 대해 살펴보고, 3 장은 결함관리 및 도구 활용에 대해 살펴본다. 4 장은 형상과 결함 관리 환경 구축에 사용 가능한 오픈 소스 도구들을 비교 설명하고 5 장에서 향후 연구 및 결론을 맺는다.

2. 형상관리 및 도구 활용

형상은 형상항목(configuration item)의 줄인 말로 정형적으로 검토되고 이해당사자들이 동의한 작업산출물들이라고 정의된다[1]. 즉, 형상은 중요 산출물이다.

기준선(baseline)은 중요한 단어인데 정형적으로 검토되고 이해당사자들에 의해 동의된 명세(specification)와 작업산출물(work products)로서, 시스템 개발 생명주기(life cycle)에서 발생하는 중요한 형상집합을 의미한다.

CMMI 에서 형상관리는 프로젝트를 수행하면서 생성되는 연구산출물의 무결성(integrity)을 확립하고 유지하는 것으로 정의된다. 형상을 관리하는 것은 조직의 개발 능력수준(capability level)이 얼마나 성숙한가 측면에서 숫자로 표기하면 2 에 해당한다. 참고로 프로세스와 품질의 품질 보증, 프로젝트 모니터링과 제어는 능력수준 1 로 형상관리보다 낮은 단계에 해당한다[1].

형상관리 활동은(1) 형상 기준선(baseline) 수립, (2) 변경 추적 및 제어, (3) 무결성 확립으로 크게 나뉜다.

형상 기준선을 수립하기 위해 수행되어야 하는 활동으로는 (1-1) 형상항목 식별, (1-2) 형상관리시스템 구축, (1-3) 기준선 생성/릴리즈가 있다 [1][2][3].

형상항목은 고객에게 전달될 제품, 의도했던 중간 작업산출물, 획득된 제품, 프로젝트 작업 환경(도구, 자산)가 예가 된다. 형상식별은 그 형상항목에 책임 있는 주인(owner)은 누구이고 다른 형상과의 관계는 무엇인 지, 그리고 그 형상의 중요한 속성은 무엇인

지 (예: 언제 테스트될 수 있는 지) 기술되어야 한다.

형상관리시스템은 권한 있는 관리자만 접근할 수 있어야 하며, 저장/인출(store/retrieve)는 정해진 시스템만을 통해 이뤄져야 하고, 형상관리 시스템에서 이뤄지는 제어 수준은 형상항목을 공유하고 전달할 수 있어야 한다. 또한 원하는 형상으로 복귀(recover)하고 로컬 작업 영역의 산출물을 업데이트할 수 있어야 한다. 기준선 생성과 릴리즈는 최초 개발 및 작성된 형상항목을 등록하고 형상관리승인을 거쳐 설정된다. 이 작업은 중요하며 형상관리는 프로젝트를 진행하는데 필요한 앞으로의 활동을 위해 현재 기준선을 검토하고 이해당사자가 동의하기 위함이며, 과거 활동에 대한 역추적이 가능하므로 기준선 생성과 릴리즈는 중요한 활동이다.

시스템 개발 생명 주기에서 관리되어야 할 형상 항목으로서 예를 들면 요구사항 산출물(예: 요구사항 정의서 및 명세서), 설계 산출물(예: 시스템 개략/상세 설계서) 등의 문서 산출물과 구현 코드가 있다. 이 구현 코드는 빈번한 변경과 형상 등록/변경을 거쳐야 하는 산출물이므로 저장, 변경 이력 관리, 인출하는 과정에 위에서 살펴본 형상관리 활동이 원활히 이뤄질 필요가 있다. 이를 위해 소프트웨어 기반의 형상관리 도구를 활용하여 형상관리 활동의 등록/변경/제어 수행에 도움을 받을 수 있다. 우리는 오픈소스 기반의 형상관리 도구 중 SVN[4]을 활용하였다.

3. 결함관리 및 도구 활용

결함관리는 빈번한 등록/변경/제어가 이뤄져야 하는 산출물인 구현 코드에서 발생한 결함(예: 버그)의 증상을 목록으로 정리하고 우선순위를 정하고 담당자를 지정하여 처리되면 처리된 상황을 반영하는 일련의 활동을 의미한다. 즉, 결함관리의 목적은 결함을 발견하고 해결하는데 있다. 따라서 결함은 추적 가능해야 하는데 이를 위해 대부분의 프로젝트 팀은 팀원 간에 커뮤니케이션 방법으로 전화, e-mail, 게시판 (예: wiki)를 활용하는 경우가 많다. 하지만 가장 적합한 커뮤니케이션 방식이 이들 방식에는 존재하지 않는다. 왜냐하면 전화는 모두가 참여하기 어렵고 전달 내용을 금방 잊고, e-mail의 경우는 모두 참여하기 어렵고 변경 과정을 추적하기 힘들기 때문이다. Wiki의 경우에 공지나 변경 사항을 팀 전체에 알리기 어렵다. 그러므로 버그 처리 상태를 파악하기 효율적인 방법이 필요한데 이를 위해 결함관리를 지원하는 소프트웨어 도구를 적극 활용하는 것이 바람직하다. 이러한 도구 중 웹 기반의 인터페이스를 제공하고, 증상을 첨부 파일로 제공하며, 변경 이력에 대해 팀원들에게 메일링이 지원되는 도구를 활용하면 좋다. 일부 도구들은 단일 프로젝트만 단독 관리되는 경우가 있는데 여러 프로젝트에 대한 지원이 가능하고, 다중 프로젝트의 상황을 그래프 등의 방식으로 제공하는 기능이 있으면 현황 파악에 필요한 통계정보를 얻을 수 있다. 우리는 이와 같은 특징을 지원하는 도구로 우리는 RedMine[4]을 선택하였다.

결함은 프로젝트에서 최종 개발되는 구현 코드를

고객에게 인도하는데 차질을 가져올 수 있는 문제이므로 어떻게든 해결되어야 하며 이를 위해 결함이 발견되면 증상을 규격화하여 저장하는 것이 필요하다.

4. 형상과 결함 관리 환경 구축

우리는 오픈소스 기반 후보 도구들을 형상관리 측면에서는 SVN[4], CVS (Concurrent Version System)[6], Git[7] 중에 비교하였고, 결함관리 측면에서는 RedMine[5], Mantis[8], Bugzilla[9]를 비교하였다. 그림 1은 형상관리 도구들의 특징 및 단점, 그림 2는 결함관리도구의 특징 및 단점을 보여 준다.

특징 항목	가격 (라이센스)	특징	단점
CVS	무료 (GPL)	- 동시버전 관리시스템 - 파일 별 히스토리 저장	- 저장소의 파일 이름 변경 불가 - 디렉토리의 이동과 이름 변경 불허
SVN	무료 (Apache)	- 저장소의 크기에 관계 없이 가지치기(branch)와 태그 넣기(tagging) 지원	- Apache를 설치해야 함 - http를 이용하는 방법 외에는 로그 남기는 방법 부재
Git	무료 (GPL)	- 분산버전 관리환경 - Git 기반의 오픈소스 프로젝트 참여 편리	- CVS나 SVN을 미리 써 보지 않으면 익히기 어려움

(그림 1) 형상관리 도구

특징 항목	라이센스	특징	단점
Mantis	무료 (GPL)	- 프로젝트 변경 이력에 대한 추적 관리 및 유지 복구 가능	- 특정 결함 검색에 검색어 and/or 연산 불가
Bugzilla	무료 (GPL)	- 결함 추적/관리 가능	- 설치/관리가 어려움 - 검색조건 입력창과 결과창이 분리
Redmine	무료 (GPLv2)	- 다중 프로젝트 가능	- 예측시간이 시간/일 단위로만 표기

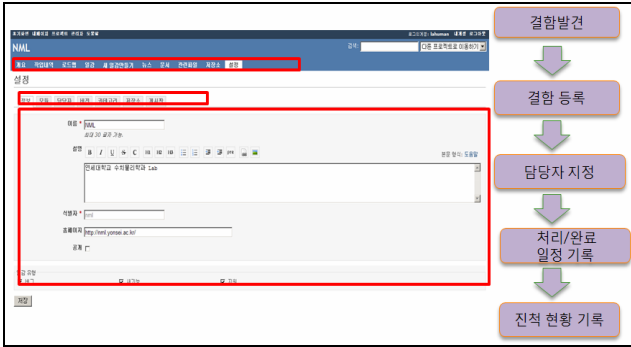
(그림 2) 결함관리 도구

이들 도구들 중 우리는 형상관리와 결함관리에 대한 환경으로 적합한 도구로 형상관리는 SVN, 결함관리는 Redmine 으로 선택하였다. 선택된 형상관리와 결함관리 도구는 15 개발 팀에 개발 프로젝트를 관리하기 위하여 시범 적용 중에 있다.

지분 표기가 부족하여 개발자 간의 공여도 표기 및 관리에 대한 표시 방법 및 저장할 수 있는 공통으로 사용될 방법이 보완될 필요가 있어 보인다.

참고문헌

[1] CMMI-DEV. 버전 1.3, CMU-SEI, 2010 년 11 월.
 [2] 한국전자통신연구원 연구개발표준프로세스 ver. 3.0, 2009 년 7 월.
 [3] ISO/IEC 15504, 2012.
 [4] Ben Collins-Sussman, “Version Control with Subversion”, O’Reilly Media, 2009.
 [5] RedMine, <http://www.redmine.org/>
 [6] CVS (Concurrent Version System)
 [7] Git, <https://git.wiki.kernel.org>
 [8] Mantis, <http://www.mantisbt.org/>
 [9] Bugzilla, <http://www.bugzilla.org/>



(그림 3) Redmine 을 이용한 결함 등록 화면

시범 운영 중인 형상관리 도구는 Eclipse IDE 환경과 연동되어 개발된 코드에 대한 commit 으로 서버에 로컬 개발된 내용을 반영할 수 있고, update 를 통해 다른 팀원이 개발한 코드에 대한 개발자 로컬 디렉토리의 반영이 가능하다. 이에 대해 대부분의 개발 팀은 유용하다는 견해인데 이는 형상관리 프로세스의 등록/변경 요청에 대해 도구를 사용한 편리성으로 해석될 수 있다.

결함관리는 Redmine 에 형상관리 디렉토리를 등록하여 형상에 대한 결함을 등록 관리할 수 있어 형상관리 저장소와 연동되어 결함관리를 체계적으로 할 수 있다는 견해이다.

또한 시스템 개발의 근간이 되고 있는 시스템 요구사항과 결함 간의 추적성이 유지될 수 있도록 프로젝트의 중요한 요구사항들을 결함관리에 반영하여 구현 결과를 형상으로 등록하고 관리하는 활동과 연결함으로써 형상관리 및 결함관리는 프로젝트 수행의 이슈관리를 포함할 수 있게 확대 되었다는 의견을 얻을 수 있었다.

5. 결론

오픈소스 기반의 형상관리 및 결함관리를 위한 도구 활용은 비용 측면에서는 무료로 진행되었고 형상은 통합개발환경에서 지속적으로 서버 저장소로 저장될 수 있었다. 이를 위해 개발 팀에 대한 형상관리 프로세스 활동에 대한 교육을 실시하여, 도구로 구축된 환경에서 개발자 개인이 작업한 결과가 변경/등록될 수 있도록 하였다.

오픈소스 기반, 웹 기반의 결함관리는 개발 팀원 간의 처리할 업무를 규격화하여 등록할 수 있는 환경을 제공하였고 수정되는 결함에 대해서는 수정 현황을 반영할 수 있었다. 결함관리는 구축된 환경에서 형상관리와 연동될 수 있어 팀 커뮤니케이션이 효율적으로 이뤄질 수 있었다. 하지만 Redmine 에서는 공동 작업에 대한 누가 얼마나 기여하였는 지에 대한