

Redmine과 Git을 활용한 헬리콥터 능동진동제어시스템 소프트웨어 형상관리

백승길* 박중용
한국항공우주연구원

Software Configuration Management for Helicopter AVCS using Redmine and Git

Seung-Kil Paek*, Joongyong Park
Korea Aerospace Research Institute

Abstract : Korea Aerospace Research Institute takes part in the vibration control software development for an Active Vibration Control System (hereafter, AVCS) for helicopters with Korea Aerospace Industries Ltd. in the Light Civil Helicopter (LCH) Project. The vibration control software is being developed per RTCA/DO-178C certification regulation and use of configuration management tools for software outputs is required. Redmine, an open source software is for issue or bug tracking and management software. Git, another open source software is a distributed version control software and is developed for Linux OS development. This paper introduces the functionalities of Redmine and Git, the reason why they are selected for the configuration management tool for the software outputs, and how they are being used for AVCS software development.

Key Words : Helicopter, Active Vibration Control System, Software Configuration Management, Redmine, Git

Received: November 1, 2016 / **Revised:** January 15, 2017 / **Accepted:** January 15, 2017

* 교신저자 : Seung-Kil Paek, seungkilpaek@kari.re.kr

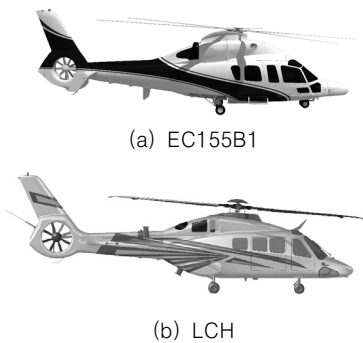
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

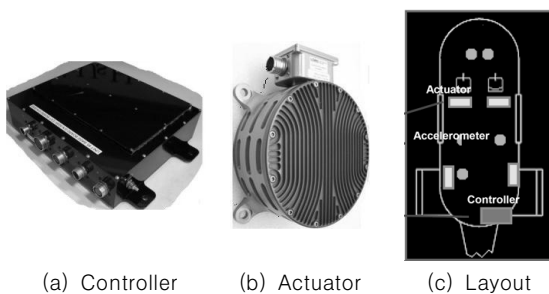
지난 2015년부터 산업통상자원부의 자금 지원으로 소형무장헬기 연계 민수헬기(LCH : Light Civil Helicopter) 핵심기술개발사업이 시작되었다. 백승길 등¹⁾이 LCH 상품성 향상 및 진동 저감에 대한 국내 기술 성숙을 위해 능동 진동 제어시스템 개발이 필요하다고 주장한 바와 같이, 헬리콥터 진동 수준을 고정익 항공기 수준으로 저감할 수 있는 능동 진동제어시스템(이하 AVCS)을 개발하기 위한 연구가 이 핵심기술개발사업의 일부로 진행되고 있다. LCH는 Figure 1과 같이 Airbus Helicopter社의 EC155B1 헬리콥터를 개조하여 개발한다.

AVCS의 원리는 로터 블레이드에서 발생하여 동체로 전달된 진동을 동체 구조에 설치된 구동기를 이용하여 상쇄함으로써 진동을 저감하는 방식이다.

AVCS는 Figure 2와 같이 제어기 컴퓨터와 구동기, 가속도 및 타코미터 센서, 와이어하니스로 구성된다. Figure 2의 (b)에 보면 헬리콥터 AVCS에 사용되는 구동기를 볼 수 있는데, 그림과 같이 원형으



[Figure 1] Light Civil Helicopter



[Figure 2] AVCS Components and Layout

로 생긴 이유는 내부에 전기모터에 의해 회전되는 편심중량이 있기 때문이다.

본 과제에서는 특히 제어기 컴퓨터에 탑재될 진동제어 구동 소프트웨어(Application SW)를 민간 항공 인증규격인 RTCA DO-178C²⁾의 DAL(Design Assurance Level) C급 인증기준에 따라 개발하고 있다. 개발된 AVCS는 LCH에 장착하여 STC (Supplemental Type Certificate) 신청을 통해 국토교통부의 인증을 받을 계획으로서 매우 수준 높은 연구라 할 수 있다. 한국항공우주연구원은 AVCS의 진동제어 소프트웨어 개발에 참여하고 있다.

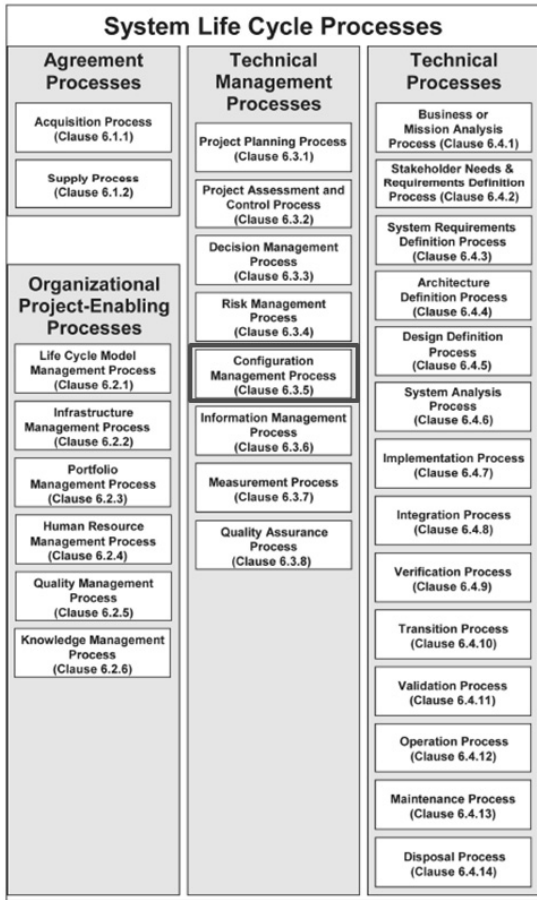
DO-178C 적용을 위하여 소프트웨어 산출물에 대한 형상관리 도구가 필요하다. 상용 형상관리 소프트웨어를 활용한 사례로서 박중선³⁾이 3SL社의 Cradle 이라는 상용 소프트웨어를 활용한 형상관리 방안을 제시한 적이 있다. 본 논문에 언급하지는 않겠지만, 연구 예산이 충분할 경우 상용 소프트웨어를 사용하는 경우가 대부분일 것으로 사료된다. 하지만, 최근에는 오픈소스 소프트웨어의 품질이 상용 소프트웨어에 못지 않고, 사용자의 목적에 맞게 소프트웨어를 수정하여 사용할 수 있기 때문에, 적절히 사용하면, 비용 대비 큰 효과를 가져 올 수 있다.

본 논문에서 오픈소스 소프트웨어를 활용한 형상관리 방안을 제시한다. Redmine⁴⁾은 이슈 및 버그 추적 및 관리 소프트웨어로서 Ruby⁵⁾라는 언어를 이용하여 작성된 오픈소스 웹 소프트웨어이다. Git⁶⁾은 또 다른 오픈소스 소프트웨어로서 분산형 버전 관리 소프트웨어인데, 세계적으로 유명한 리눅스 OS 개발을 위해 사용되고 있다. 본 논문에서는 Redmine 과 Git 의 기능에 대해 소개하고, 소프트웨어 산출물의 형상관리도구로서 선택한 근거 및 이를 활용한 AVCS S/W 개발을 위한 형상관리 방안에 대해 제시하고자 한다.

2. 국제 표준의 형상관리

2.1 시스템엔지니어링(SE)의 형상관리

SE 국제표준인 ISO/IEC/IEEE 15288⁷⁾은 시스



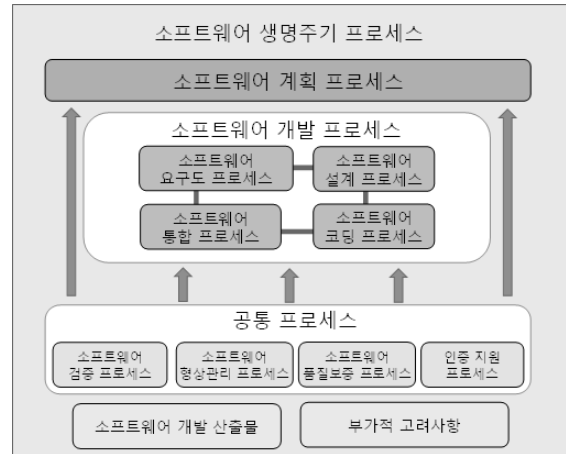
[Figure 3] System Life Cycle Process

템 생명주기를 4개의 프로세스로 구성한다. 이중 형상관리는 기술관리프로세스의 하부 프로세스로 정의한다.

형상관리의 목적은 시스템 요소와 형상을 전 생명주기에 걸쳐서 관리하고 통제하는 것이다. 주요 업무로는 형상관리 계획의 수립, 형상 식별, 형상 변경 관리, 형상 현황 관리, 형상 평가, 형상 배포 통제 등이 있다.

2.2 RTCA/DO-178C의 형상관리

항공기에 장착하는 전자 시스템 소프트웨어의 인증기준인 RTCA/DO-178C에서 소프트웨어 생명주기는 계획(Planning), 개발(Development), 공통(Integral) 프로세스의 3개의 프로세스로 구성한다. 이중 형상관리는 Figure 4⁸⁾와 같이 공통 프로세스



[Figure 4] Software Lifecycle Process

(Integral Process)의 하부 프로세스로 정의한다. 소프트웨어 형상 관리 프로세스의 활동은 형상 식별(Configuration identification), 기준선 수립 및 추적성(Baselines and traceability), 문제 보고, 추적 및 시정 활동(Problem Reporting, Tracking and Corrective Action), 변경 통제(Change Control), 변경 검토(Change Review), 형상 현황 관리(Configuration Status Accounting), 보존 및 배포 관리(Archive, Retrieval, and Release)로서 앞서 시스템엔지니어링 형상관리와 대체로 일치한다. 단, 형상관리 계획 수립은 소프트웨어 계획 프로세스의 업무이다.

3. 오픈소스 형상관리 도구

3.1 Redmine

Redmine은 오픈 소스로 제공되는 유연한 프로젝트 관리 웹 어플리케이션으로 간단히 정의할 수 있다. Redmine은 다음의 기능을 제공한다.

- 다중 프로젝트 동시 지원
- 유연한 역할 기반의 접속 통제
- 유연한 이슈 추적 시스템
- 간트 차트와 달력 기능
- 뉴스, 문서, 파일 관리
- 코드버전 관리도구와의 연동

역할 » 연구자

이름 * 연구자

이 역할에게 일감을 맡길 수 있음

일감 보임: 비공개 일감 제외

Time logs visibility: All time entries

Users visibility: All active users

Member management: All roles, Only these roles: 관리자, 연구자, 보고자, 형상통제위원회(CCB), QA

권한

프로젝트: 프로젝트 생성, 프로젝트들 닫거나 다시 열기, 구성원 관리, 하위 프로젝트 만들기, 프로젝트 편집, 프로젝트 모듈 선택, 버전 관리, Create SSH keys

일감관리: 일감 범주 관리, 일감 추가, Copy issues, 하위 일감 관리, 일감 보기, 일감 편집, 일감 관계 관리, 일감을 공개나 비공개로 설정

[Figure 5] Role and Authority

- 다중 데이터베이스 지원
- 플러그인을 통한 기능확장
- 다중언어 지원
- 이메일 통보

여기서 다중 프로젝트 동시 지원이란 여러 개의 프로젝트를 한꺼번에 관리 할 수 있다는 의미이다. 한 프로젝트 아래에 하부 프로젝트를 구성할 수도 있다. 유연한 역할 기반의 접속 통제란 개발 담당자, 품질보증 담당자, 형상관리 담당자라는 역할이 있을 때, 각 역할 별로 어떤 이슈의 상태를 변경할 수 있는 권한을 다르게 조정할 수 있어서 사용자가 원하는 논리에 따라 업무 흐름을 구성할 수 있다는 것이다. 이것이 Redmine의 가장 큰 장점인데, 이를 바탕으로 갖가지 업무 흐름을 작성할 수 있어서 문제 보고, 추적 및 시정 활동, 변경 통제, 변경 검토의 업무 흐름을 원하는 대로 새로 개발하고 조정 가능하다. Figure 5는 특정 역할에 대한 권한설정의 예를 보여주고 있다.

사용자 정의항목은 이슈 등록과 관리를 프로젝트의 개별특성에 맞게 설정할 수 있는 기능이다. 필요에 따라 Figure 6과 같이 이슈유형, 심각도, 품보화

사용자 정의 항목

이름	형식	필수	모든 프로젝트	사용됨	정렬
이슈유형	목록	<input checked="" type="checkbox"/>		4개 프로젝트	▲ ▼ ↕
심각도	목록	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
품보화수치	문자열	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
식별일자	날짜	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
점검담당자	사용자	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
점검영역	목록	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
점검항목	문자열	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
점검속성	목록	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
점검대상물	목록	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕
결과유형	목록	<input checked="" type="checkbox"/>		한 프로젝트	▲ ▼ ↕

[Figure 6] Custom Fields

이력

백 송길(가) 8달 전에 변경 #1

- 상태를(를) 등록에서 진행(으)로 변경되었습니다.
- 진척도를(를) 0에서 10(으)로 변경되었습니다.

체크리스트 작성

백 송길(가) 7달 전에 변경 #2

- 설명이 변경되었습니다. (diff)

백 송길(가) 7달 전에 변경 #3

- 진척도를(를) 10에서 20(으)로 변경되었습니다.

항목이 선택되었지만 아직 속성 필드를 지정하지 않았습니다. 이 항목을 선택한 후, 이 항목의 속성을 지정하십시오.

- 최초의 초기화 함수는 각 함수에서 사용할 array를 global 변수로 선언한다. à compile하면 map 파일에 설정된 메모리 변수가 계산되어 나옴(ebss). Over 시에는 compile error가 나옴.

관련된 개정판:

개정판 cc45da4t 백 송길(가) 5달 (5) PDI 값 복사

개정판 e302cd07 백 송길(가) 5달 CAS_Init.eap 추방안 필요성 추가!

개정판 5dff692e 백 송길(가) 5달 Data Flow Diagram

개정판 be063a11 백 송길(가) 5달 Data Flow Diagram

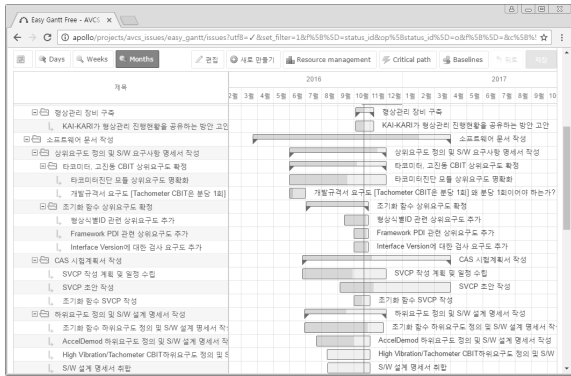
개정판 cd37458c 백 송길(가) 5달 Flow Chart 에 AI

개정판 d3d7f549

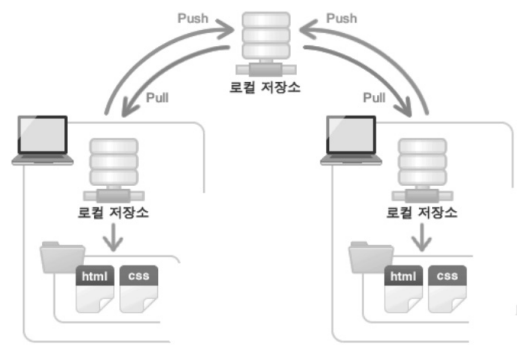
[Figure 7] Change History

동차수 등의 사용자 정의항목을 작성하여, 이슈 관리에 활용할 수 있다. 이슈는 Redmine의 중심 기능이다. 사용자가 정해 놓은 이슈 유형을 선택하여 프로젝트 구성원이 등록함으로써 이슈를 프로젝트 구성원에게 보고할 수 있다. 각 이슈의 종결을 위해 담당자는 적절한 업무를 수행하고 수행 내용을 기록하고, 이슈의 해결 과정에 새로운 담당자를 선정하여 잔여 업무의 처리를 요구한다. Redmine 내의 모든 활동은 E-Mail 통보가 가능하고 E-Mail 상의 링크를 통해 해당 업무에 쉽게 접근할 수 있다. 이러한 과정은 Figure 7과 같이 각 이슈별로 이력으로 남게 되어 각 이슈의 진행과정을 나중에라도 쉽게 추적할 수 있다.

다른 기능과 함께 이런 추적 기능으로 인해 사업 관리/형상관리 도구로 많이 사용되고 있다. 특히 오픈 소스로 제공되어 비용 부담이 적으므로 주로 대



[Figure 8] Easy Gantt Chart Plug-in



[Figure 9] Remote and Local Repositories

학, 중소기업에서 프로젝트 수행 시 활용하고 있다. 초보 사용자의 경우 컨설팅 업체를 통해 수행 프로젝트의 성격에 맞게 적은 비용 부담으로 이 맞춤형 제작이 가능하다.

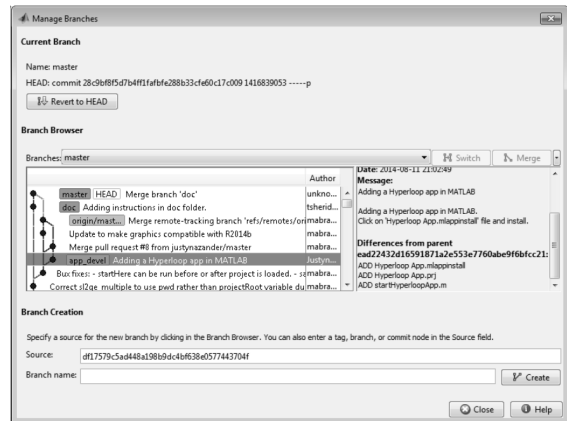
Redmine을 Software 형상관리 도구로 사용하려면 코드버전 관리도구와의 연동이 필요하다. Redmine은 Subversion⁹⁾, CVS¹⁰⁾, Git, Mercurial¹¹⁾, Bazaar¹²⁾, Darcs¹³⁾ 등의 유수의 코드버전 관리 소프트웨어와 통합이 가능하다. Git의 경우는 Redmine에서 제공하는 연동 기능 이외에도 별도의 플러그인을 통해 연동이 가능하다. 또한 기본 Redmine의 기능을 확장하는 플러그인도 활용 가능하다. Figure8은 기본 Gantt 차트 기능을 개선한 Easy Gantt라는 플러그인이다.

데이터베이스로는 무료 데이터베이스 프로그램인 MySQL¹⁴⁾, PostgreSQL¹⁵⁾, SQLite¹⁶⁾ 등을 지원한다.

3.2 Git

Git은 리눅스 OS 개발을 위해 개발된 분산형 소스 코드 버전 관리 소프트웨어로서, 원격 저장소의 모든 코드의 이력을 로컬(local) 저장소로 복사하여 독자적으로 코드 개발을 지속할 수 있다.

원격 저장소(Remote Repository)란 여러 사람이 함께 공유하기 위한 저장소로서 파일은 원격 저장소 전용서버에서 관리된다. 로컬 저장소(Local Repository)란 사용자 PC에 파일이 저장되는 개인 전용 저장소를 말한다(Figure 9 참고). 원격 저장소



[Figure 10] Matlab/Simulink Git Integration

의 정확한 복사본을 복제(clone)하여 원격 저장소에 연결하지 않고도 독자적인 코드 개발이 가능하다. 추후 여러 사람이 분산 개발한 소스 코드를 원격 저장소로 통합하기 위한 통합(Merge) 기능을 제공한다.

Git은 최근 들어, 분산 개발을 하지 않을 수 없는 오픈소스 소프트웨어 개발이 확산되면서 소스 버전 관리 소프트웨어의 대세가 되었다. 저명한 Google, facebook, Microsoft, twitter 등의 소스 코드는 Git을 이용하여 버전관리를 하고 있고, 본 연구에서 능동진동제어 논리 구현을 위해 사용하는 Matlab Simulink도 Git 연동 기능을 제공하고 있다(Figure 10 참고).¹⁷⁾

3.3 Redmine 와 Git의 통합

Redmine을 소프트웨어 산출물에 대한 형상 현황을 식별할 수 있도록 저장소를 생성하고 그 현황을 Redmine 내에서 통합 운영할 수 있어야 한다. 이것

development-doc @ master

이름	크기	개정판	마지막 수정일	저자
01_DO-178C_Outputs		29e79a1f	약 한달	백 승길
02_Purchase		608aae93	약 한달	백 승길
03_Reporting		9ec453be	약 한시간	백 승길
04_Activities		a0753b5e	26일	백 승길
README.md	23 Bytes	f57ec45b	약 2달	redmine@example.net

최근 개정판

#	날짜	저자	
9ec453be	2016/05/25 07:41	백 승길	2016년 5월 24일 주간!
ed2b2ca8	2016/05/18 08:22	백 승길	2016년 5월 10일 주간!
a0753b5e	2016/04/29 18:03	백 승길	close ###
6fda4a07	2016/04/29 15:16	백 승길	붙임자료와 같이 작성하
4174fc66	2016/04/27 13:31	백 승길	2016년 4월 27일 주간!
29e79a1f	2016/04/20 16:14	백 승길	SDP : 표 3. empty cell

[Figure 11] Redmine Git Hosting Plug-in

은 역시 오픈소스인 Redmine Git Hosting 플러그인을 통해 이루어진다.

사용자는 TortoiseGit¹⁸⁾ 등의 무료 소프트웨어를 활용하여 개발 저장소로 데이터를 전달한다. 기준 형상으로 결정된 형상항목은 형상관리 담당자(이하 CCB)만 쓰기 권한을 갖는 Master 저장소로 형상관리 담당자가 직접 복사한다.

4. 형상관리

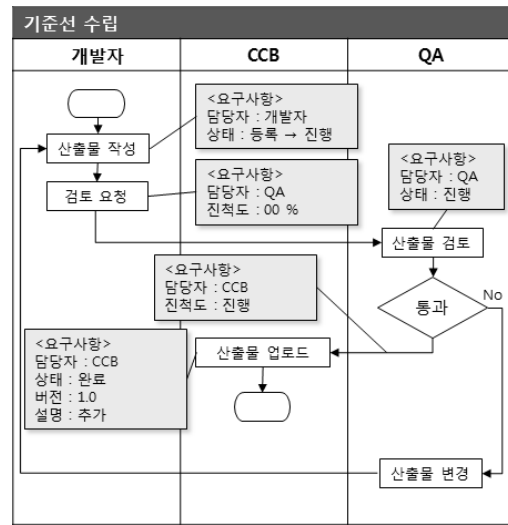
Redmine과 Git에 대한 소개에 이어 이를 통해 형상 식별, 기준선 수립 및 추적, 문제 보고 및 시정 활동, 변경 관리가 어떻게 이루어지는 살펴 보자.

4.1 형상 식별

형상식별 활동의 목표는 형상 항목들의 관리 및 참조를 위한 기준이 수립되도록 각 형상 항목(및 그들의 후속 버전)을 명확하게 선정하고 분류하는 것이다. 이것은 형상관리 담당자의 활동이라고 할 수 있다.

4.2 기준선 수립 및 추적

기준선 수립의 목표는 향후 소프트웨어 생명주기 프로세스 활동의 기반을 정의하고 형상 항목들 간



[Figure 12] Baseline Establishment

에 추적성과 이를 참조하고 관리하도록 하는 것이다.

요구사항의 기준선은 수립되었음을 전제로 특정 소프트웨어 산출물의 기준선을 수립하는 절차는 다음과 같다.

개발 담당자는 먼저 요구사항을 등록한다. 요구사항 등록은 Redmine의 이슈에 사용자 정의로 생성한 "요구사항"이라는 유형을 이용한다.

이슈 내용은 특정 소프트웨어 산출물로 작성된다. 그 후 산출물 작성이 완료되면 Git을 통해 개발 저장소로 이송하고 품질보증 담당자(이하 QA)에게 해당 Commit ID와 함께 검토를 의뢰한다. Redmine에서 이 행위는 담당자를 변경함으로써 이루어진다.

QA는 개발 저장소에서 Git을 통해 산출물을 확인한다. 검토 결과 문제점이 없으면 CCB로 형상등록을 의뢰하게 된다. 이것은 앞서와 마찬가지로 담당자를 수정함으로써 이루어진다. 만일 시정사항이 발견하면 "시정사항"이슈를 등록한다.

CCB는 Master 저장소에 "버전 1.0"이란 Comment와 함께 Commit 함으로써 형상 등록을 완료하게 된다. 그 다음 처음 설정했던 "요구사항" 이슈의 상태를 "완료"로 설정하고 해당 Commit ID를 기록한다.

4.3 문제 보고 및 시정 활동

본 활동의 목표는 소프트웨어 계획 및 표준에 대

한 프로세스의 부적합부분을 기록하고, 소프트웨어 생명주기 프로세스 결과물의 결함과 소프트웨어 제품의 비정상적인 거동을 기록하며, 이러한 문제들의 해결책을 확보하는 것이다.

문제 보고는 QA 담당자가 시정사항을 등록함으로써 이루어진다. 시정사항 등록은 Redmine의 이슈에 사용자 정의로 생성한 “시정사항”이라는 유형을 이용한다.

이때, 다음 사항이 기술되어야 한다.

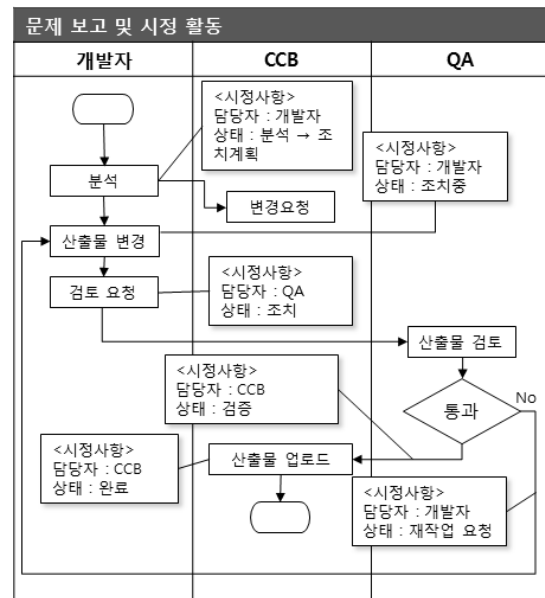
- 문제가 발견된 형상항목 및/또는 소프트웨어 생명주기 프로세스 활동의 식별
- 수정되어야 하는 형상항목의 식별 또는 변경되어야 하는 프로세스를 식별
- 문제를 이해하고 해결할 수 있도록 문제를 기술

QA는 시정사항을 등록하고 담당자를 개발자로 변경한다. 개발자는 업무 진행에 따라 이슈의 상태를 “분석”→“조치계획”→“조치 중”으로 차례를 변경한다. 이 과정 중에 기술된 내용이 문제를 해결하기 위해 취해진 시정조치 내용이 되므로 적절한 기술이 이루어져야 한다.

개발자는 변경요청이 필요할 경우 CCB에 변경요청을 한다. 산출물 변경이 허용된 경우 산출물 수정을 완료한 후 이슈의 상태를 “조치”로, 담당자를 QA로 변경하고 Git을 통해 개발 저장소로 이송하고 해당 Commit ID를 기록함으로써 검토 의뢰를 한다.

QA는 Git을 통해 산출물 확인하고 완료시 CCB로 담당자를 변경하고, 상태는 “검증”으로 변경함으로써 형상등록을 의뢰하고, 미완료시는 다시 담당자를 개발자로, 상태를 “재작업요청”으로 변경한다. 개발자는 상태를 다시 “등록”으로 변경하고 시정 사항을 처리한다.

CCB는 Master 저장소에 증가된 버전 번호의 Comment와 함께 Commit 함으로써 형상 변경을 완료한 후 “시정사항” 이슈의 상태를 “완료”로 설정하게 된다.



[Figure 13] Problem Report and Corrective Action

4.4 변경 관리

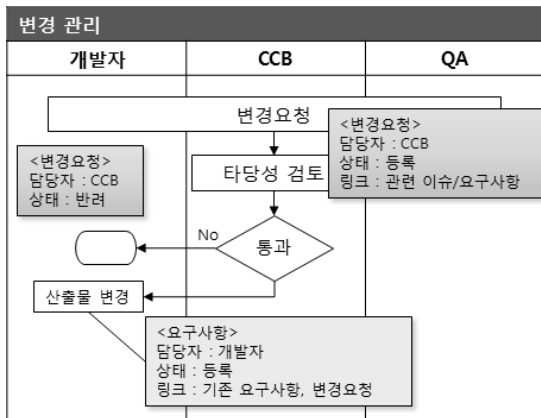
베이스 라인 수립이 완료된 형상 품목은 변경 관리를 받게 된다. 변경관리 활동의 목표는 소프트웨어 생명주기 프로세스를 통해 변경 관리 대상인 형상 항목에 대한 변경사항을 기록, 평가, 결론 및 승인을 제공하는 것이다.

변경요청 이슈가 등록되면, 담당자를 CCB로 지정한다. CCB는 타당성을 검토하여 변경요청 이슈의 상태를 “반려”로 수정함으로써 변경요청을 반려하거나, 상태를 “변경승인”으로 수정함으로써 변경승인을 결정한다. 기존 요구사항 담당자는 CCB, 상태는 “변경”이 된다. 개발자는 새로운 요구사항을 등록한다. 이때부터는 4.2절의 절차와 동일하다. 단, CCB는 Master 저장소에 증가된 버전 번호의 Comment와 함께 Commit 함으로써 형상변경을 완료한다.

5. 결론

ISO/IEC/IEEE 15288 및 RTCA /DO-178C에서 제시하는 형상관리 활동을 비교하였다.

그 결과, 시스템엔지니어링 표준인 ISO/IEC/IEEE 15288에서 제안하는 형상관리의 주요 활동인 형상



[Figure 14] Change Control

관리 계획의 수립, 형상 식별, 형상변경 관리, 형상 현황 관리, 형상 평가, 형상 배포 통제가 항공전자 시스템의 소프트웨어 개발 표준인 DO-178C에서도 유사한 체계와 내용으로 제안되었음을 확인하였다. LCH 헬리콥터 AVCS의 진동제어 소프트웨어 개발 형상관리를 RTCA/DO-178C에 따라 수행하므로 시스템엔지니어링의 형상관리 체계가 적용되었다고 할 수 있다. 형상관리의 구체적 방안으로 오픈소스 프로그램인 Redmine 과 Git을 활용한 소프트웨어 산출물 형상관리 방안을 제시하였다. 소프트웨어 개발 활동에서는 버전 관리가 매우 중요하며, 버전 관리를 위해 Git을 활용하는 점이 일반 시스템과는 다른 점이다. 또한 형상관리활동에 오픈 소스를 활용함으로써 비용을 절감하면서 형상관리 목표를 달성할 수 있을 것으로 기대된다.

후 기

본 논문은 산업통상자원부 소형무장헬기 연계 민수헬기 핵심기술개발사업 연구결과 중 일부임.

References

1. Seung-Kil Paek, Keun-Woong Song, and Seung-Ho Kim, "Current Status of Helicopter Active Vibration Control System and Development Plan," Proceeding of the 2013 KSAS Spring Conference, 2013, pp. 888-891.
2. Anonymous, *Software Considerations in Airborne Systems and Equipment Certification*, RTCA DO-178C, RTCA, Inc. Dec. 13, 2011.
3. Jong-Sun Park, "Study on Configuration Management Method using SE Computer Support Tool," Journal of the Korean Society of Systems Engineering, Vol. 7, No. 1, 2011, pp.53-56.
4. www.redmine.org
5. www.ruby-lang.org
6. git-scm.com
7. Anonymous, *Systems and software engineering - System life cycle processes*, ISO/IEC/IEEE 15288, ISO/IEC/IEEC, 2015.
8. Seung-Kil Paek, "Software Development Plan for an Active Vibration Control System," Proceeding of the 2016 KSAS Spring Conference, 2016, pp. 630-631.
9. https://subversion.apache.org/
10. http://www.nongnu.org/cvs/
11. https://www.mercurial-scm.org
12. http://bazaar.canonical.com/en/
13. http://darcs.net/
14. https://www.mysql.com
15. https://www.postgresql.org
16. https://sqlite.org
17. https://kr.mathworks.com/help/simulink/ug/set-up-git-source-control.html
18. https://tortoisegit.org