

아리랑 위성의 탑재 소프트웨어 형상관리

강 수 연^o
한국항공우주연구원
sykang@viva.kari.re.kr

Software Configuration Management of KOMPSAT-1

Soo-Yeon Kang^o
Korea Aerospace Research Institute (KARI)

요 약

다른 어느 분야의 소프트웨어 개발 분야보다도 형상관리의 필요성의 나타나는 분야로 위성 탑재 소프트웨어 분야를 꼽을 수 있는데 이는 위성소프트웨어의 기본적인 특성에 기인한다고 볼 수 있다. 위성 탑재 소프트웨어는 어느 소프트웨어보다도 그 신뢰성을 중시하고 있으며 이러한 신뢰성은 개발 과정상의 잘 정의되어진 형상관리에 있다. 본 논문에서는 아리랑 위성 탑재 소프트웨어 개발에 과정에서의 형상관리 방법 및 절차를 기술하였으며 이러한 형상관리에 아리랑 위성 탑재 소프트웨어는 개발되고 위성에 탑재되어 그 임무를 훌륭히 수행하고 있다. 이 논문을 통해서 고도의 신뢰성을 요구하는 소프트웨어 개발에 있어서의 형상관리 방법에 도움이 되고자 한다.

1. 개요

아리랑 위성은 한반도 지형 관측, 해양관측 및 과학실험을 목적으로 하며 한국항공우주연구소를 총괄 주관으로 국내 7개 기업체가 참여하여 미국 TRW사와 공동으로 개발되어 1999년 12월에 발사되었다. 현재는 그 임무를 원활히 수행하고 있어 위성 탑재 소프트웨어의 성공적인 수행을 보여주고 있다.

아리랑 위성 탑재 소프트웨어 개발에는 10명 ~ 15명 개발 인력이 투입되어진 소프트웨어 개발에서는 중형 프로젝트라고 볼 수 있다. 그러나 어느 응용 소프트웨어 개발과는 달리 그 신뢰성이 강조되는 성격을 갖는다. 발사후의 임무 수행 중 소프트웨어 버그등의 치명적인 문제점은 전체 위성개발에 치명적인 손실을 주게된다. 다수의 개발자들이 연관되고 응용 소프트웨어의 신뢰성이 중요시되는 경우에는 개발기간동안의 전체 소프트웨어의 형상 관리는 중요한 역할을 한다. 위성 탑재 소프트웨어는 개발기간 동안에 형상관리와 버전 제어 관리하에서 개발되어진다.

형상관리 절차들은 소프트웨어 버전제어를 위해서 소프트웨어 코딩, 변경, 유니트 테스트, 전체 통합테스트들에 대한 개발 절차를 정의하며 이는 개발자들에 의해서 지켜지게 된다. 이러한 절차들의 정의는 [1]에 설명되어져 있다.

버전 제어는 베이스 라인들 사이에서 모듈 개발의 과정을 추적할 수 있는 방법을 제공하며 개발자들은 소프트웨어 개발 과정중의 버전 제어를 위해서 자동화된 도구를 사용한다. 본 위성 탑재소프트웨어 개발에서 버전 제어를 툴로서 PVCS VM(Polytron Version Control System Version Manager)을 채택하여 사용하였다. 버전 제어를 위한 많은 자동화 툴들중에서 본 소프트웨어 개발의 개발인력 규모, 비용,등의 성격을 고려하여 부합된다는 판단하에서 PVCS VM을 선택하게 되었다.

2장에서는 탑재 소프트웨어의 전체 개발에 대하여 설명하며 3장에서는 개발과정에서의 형상관리, 버전제어 및 이와 연관된 빌드 과정에 대하여 서술한다. 4장 결론을 마지막으로 본 논문을 마치고자 한다.

2. 탑재 소프트웨어 개발 절차 및 환경

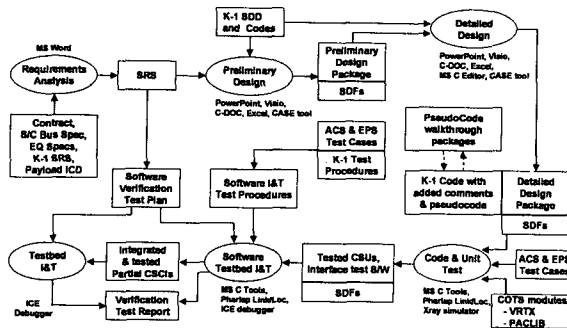


그림 1. 탑재소프트웨어 개발 절차

그림 1.은 탑재 소프트웨어 전체 개발 과정을 나타낸 것이다. 개발을 위한 시스템(여기서 위성)의 요구분석을 통하여 소프트웨어적으로 필요한 요구사항들이 분석이 되면 이를 바탕으로 소프트웨어 요구사항 문서들이 작성이 되어진다. 기본 설계 단계에서는 소프트웨어 요구사항을 바탕으로 다양한 툴들을 사용하여 전체 소프트웨어의 기본 설계를 수행하며 이 단계 수행과정 중에 발생되어진 사항들은 문서로 작성되어 SDF(Software Development Folder)에 바인딩 되어져서 관리되어진다, 상세 설계 단계에서는 기본 설계의 내용을 직접적으로 코드화 되어지며 구현될 언어로 옮겨지기 전에 PseudoCode로 구현되어 소프트웨어 탑재에서 Walkthrough를 거쳐 실제 코드화 되어진다. 모든 모듈들이 코드화 되어진 후에는 각각의 모듈 테스트를 거쳐서 중간에 생성되어진 오류들을 점검하여 이를 수정하여 재 테스트되어진다. 이러한 모든 모듈 테스트들이 완료 되어진 후에는 전체 소프트웨어를 통합하여 TestBed상에서 테스트하게 된다, 이에 대한 절차는 SW I&T(Integration & Test) 절차를 따르도록 한다. 여기서 테스트 단계들에서는 QA가 참석하여 이를 지켜보고 검증하도록 되어져있다.

3. 형상관리

소프트웨어 보증은 [2]에서 정의된바와 같이 미 TRW사의 기존의 표준과 계약 요구사항에 바탕을 두고있으며 소프트웨어 품질 보증 엔지니어에 의해서 수행이 되어진다. 소프트웨어 보증은 크게 소프트웨어 품질 보증관리와 소프트웨어 형상관리로 구성이 되어있다.

소프트웨어 품질 보증 요구사항은 [3]에 정의되어져 있으며 다음과 같은 일을 수행한다.

- 소프트웨어 문서들, 검증 테스트 계획과 절차들을 검토한다.
- 요구사항 추적, 소프트웨어 개발 폴더들과 문제 보고

서들을 조사 및 심사한다.

- 테스트 보고서를 검토하고 테스트 결과를 확인한다.
- 문제에 대한 보고서 체계를 세운다.

소프트웨어 형상관리는 문서 관리와 소프트웨어 변경 관리 체계로 나눌 수 있다. 문서 관리는 작성되어진 문서들을 승인하고 제어하며 변경 사항이 있을 때에는 이를 검토하고 재 승인하고 변경사항에 대하여 문서화한다. 소프트웨어 변경 관리 체계는 다음의 순서로 이루어진다.

- SPR(Software Problem Reports)이 발생되어진다.
- 소프트웨어 CCB(Configuration Change Board)에서는 소프트웨어 요구사항에 부합하여 필요로 되어지는 변경 사항들을 승인한다.
- ECB(Engineering Changes Board)에서는 소프트웨어 서브시스템 외부로 전달될 소프트웨어 요구사항에 대한 모든 변경을 승인한다.
- 위에서 검증되어진 변경 사항들은 베이스라인 요구사항에 반영이 되어진다.
- 소프트웨어 소스 코드 변경을 위해서 PVCS VM을 사용한다.
- 위에서 소프트웨어 소스 코드 변경등의 접근등을 제어하는 사람을 ACM(Access Control Manager)라고 하는데 ACM의 수행 업무는 다음과 같이 요약 설명할 수 있다.

- 필요할때 PVCS VM내에 Locked files을 unlocking
- 개발자와 제어되는 revision사이의 충돌을 해결
- PVCS VM archives 를 백업
- PVCS VM 접근 데이터베이스를 관리
- History files을 검토
- 소스 파일들의 형상화와 복사본 분배
- Version 과 Revision 번호를 할당

3.1 버전 제어

소프트웨어 개발은 개발자 각각의 PC에서 수행되어지며, 코딩과 유니트 테스트는 디버거 스크립터와 테스트 드라이버들을 이용하여 수행되진다. 그리고 전체 통합은 어느 특정 개발의 PC에서 행하여진다. 형상관리 제어는 전체 소프트웨어 개발 형상관리 업무를 책임지고 있는 ACM(Access control Manager)에 의해서 관리되고 수행된다. ACM은 형상화되어진 소프트웨어의 복사본을 모든 개발자들에게 배포하고 이 소프트웨어 변경들은 소프트웨어 개발 팀과 ACM사이의 대화에 의해서 수행되어진다.

각각의 개발자들 PC상의 PC디렉토리 구조는 형상화된 소스 모듈들을 저장하고 변경을 하기 위해서 일관성 있는 방법을 제공한다.

버전 관리자는 탑재 소프트웨어의 변경을 관리하기 위한 자동화된 툴이다. 앞에서 설명한 바와 같이 본 소프트웨어 개발을 위해서 PVCS VM을 사용하였다. PVCS VM은 거의 모든 버전 관리 소프트웨어들의 가지고 있는 기능을 가지고 있으며 이는 다음과 같이 설명될 수

있다.

- 소프트웨어 변경을 위한 형상관리 틀
- 소스 파일들에 대한 Check-in, Check-out 기능과 보안기능을 제공한다.
- History Log 보고서를 통해서 소프트웨어 변경의 흐름을 추적할 수 있다.
- 변경 사항들의 동일성을 위해 버전 차이에 대한 보고서들을 제공한다.

PVCS VM은 NT Server 상에 상주하면서 모든 개발자의 각각의 모듈들에 대한 접근 권리를 관리한다. 모든 개발자가 소프트웨어 모든 소스에 대한 접근 가능하나 변경은 각자의 개발 소프트웨어에 한해서 제한되어진다.

3.2 빌드 (Build)

탑재 소프트웨어 빌드는 SPR 또는 QFR(Quick Fix Report)에서 제기되어진 문제점을 해결하고 이를 반영하여 소스코드를 고치는 절차로 볼 수 있다. 문제 제기된 SPR 문서와 이를 해결하기 위해 빌드 수행 절차 중간과정에서 생성된 파일들, 결과 파일들은 ACM에 의해서 관리되어진다. 이들 파일들은 SPR Binding에 필요한 내용이 된다. 따라서 Build 제기된 SPR, 빌드 수행과정 중간 파일들과 결과 파일들, 그 외 결과 보고서들은 탑재 소프트웨어 빌드 과정에 포함되는 부산물이다. 그림 2.는 빌드 과정 전체를 보여 주고있다.

Make 유틸리티등은 모듈 레벨과 프로세서 레벨 소프트웨어 빌드 관리를 제공한다. Make 파일들을 사용하여 PVCS로부터 특정 모듈 버전 검색, 모듈 컴파일, 링킹을 제공한다. 여기서는 PVCS Configuration Builder를 사용하여 소프트웨어 빌드를 수행하였다.

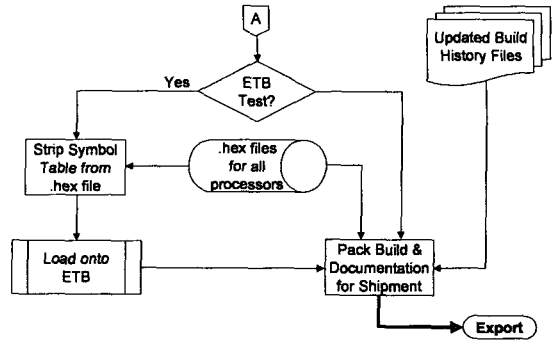


그림 2. 빌드 수행 절차

4. 결론

앞서 설명한 소프트웨어 형상관리 방법을 통하여 아리랑 위성 탑재 소프트웨어 개발을 하였으며 그 결과물인 탑재 소프트웨어는 아리랑 위성에 탑재되어 그 임무를 원활히 수행하고 있다. 이는 탑재 소프트웨어의 성공적인 개발 완수를 의미한다. 이러한 경험은 실제로 2호기 위성 탑재 소프트웨어 개발과정에도 반영이 될 것이다.

5. 참고 문헌

- [1]. "KOMPSAT Software Standards and Procedures", D23910, 1996
- [2]. "KOMPSAT Product Assurance Program Plan", No.CDRL-PA-01, 1996
- [3]. "KOMPSAT Flight Software Management Plan", D23909, 1996

