

논문 2015-1-1

소프트웨어 형상관리 정의 및 관리 가이드

권기태*

The Definition and Management Guideline for the Software Configuration Management

Ki-Tae Kwon*

요 약

소프트웨어 형상관리는 소프트웨어 수명주기 동안에 생성되는 시스템 내의 소프트웨어 항목을 식별, 정의, 제작, 수정, 배포하고 소프트웨어 항목에 대한 수정 요구를 보고하고 기록하는 행위를 기술적인 절차를 적용하여 하나의 프로세스로 관리하는 것이다. 효과적인 소프트웨어 형상관리는 소프트웨어 항목의 안전성, 일관성 및 정확도를 보장한다. 본 연구는 소프트웨어 사업 이해당사자들이 프로젝트 전반에 걸쳐 일어나는 변경에 대해 소프트웨어 개발 산출물들의 배치나 구성을 일관성 있게 관리할 수 있도록 형상관리 전체 프로세스, 작업 절차, 산출물, 고려사항을 정의하고, 이를 기반으로 문서화 프로세스 및 관리를 위한 가이드라인을 제시한다. 본 연구는 소프트웨어 수명주기 프로세스 표준인 '공공부분 SW사업 발주 관리 표준 프로세스(TTAS.KO-09.0038)' 및 'ISO/IEC 12207:2008 소프트웨어 및 시스템 공학 수명주기'를 기초로 작성하였으며, '소프트웨어사업 관리감독에 관한 세부지침'에 명시된 활동 및 작업들간과도 상호 호환성을 유지하도록 구성하였다.

Abstract

Software configuration management is identifying software items in the system that can be produced during the software life cycle. And it defines, creates, modify, distributes, reports the correction request for software items, manages in a single process. Effective software configuration management is to ensure consistency, and accuracy. In this study, we present the guideline for the entire configuration management process so that it can be managed consistently the configuration of software development artifacts, work procedures, artifacts. This study is based on the standard process(TTAS.KO-09.0038) of the public part SW business and the standard of life cycle process of software(ISO / IEC 12207 2008 software and system engineering life cycle). This guideline is configured to maintain the activities of 'detailed instructions on software project management supervision' and maintain mutual compatibility of them.

한글키워드 : 소프트웨어 형상관리, 수명주기 프로세스, 소프트웨어사업 관리 감독

keywords : software configuration, life period process, management

* 강릉원주대학교 컴퓨터공학과 교수

(email: ktkwon@gwnu.ac.kr)

접수일자: 2015.5.15. 심사완료: 2015.5.28.

게재확정: 2015.6.20.

1. 서 론

소프트웨어 형상관리는 소프트웨어 수명주기

동안에 생성되는 시스템 내의 소프트웨어 항목을 식별, 정의, 제작, 수정, 배포하고 소프트웨어 항목에 대한 수정 요구를 보고하고 기록하는 행위를 기술적인 절차를 적용하여 하나의 프로세스로 관리하는 것이다. 효과적인 소프트웨어 형상관리는 소프트웨어 항목의 안전성, 일관성 및 정확도를 보장한다.

소프트웨어 형상관리의 목적은 다음과 같다.

- 산출물 품질의 향상
- 개발 및 유지보수 생산성의 향상
- 사용자 요구사항을 체계적으로 관리

소프트웨어 수명주기 모델은 소프트웨어 사업의 프로세스 기준을 제시한다. 소프트웨어 수명주기 모델은 소프트웨어 사업을 진행할 때 수행되어야 할 활동들과 그 활동들의 순서를 정하고 관리, 통제하기 위한 기준을 제공하며, 사업에 참여한 여러 사람들 간의 원활한 의사소통을 위한 기준을 제공한다. 소프트웨어 수명주기 모델은 중요한 산출물이 작성되는 이정표를 정의한다. 소프트웨어 형상관리의 활동은 이러한 이정표에 맞추어 형상 항목들을 관리하고 기준선을 설정해야 한다.

소프트웨어 형상관리는 1960년대에 하드웨어를 관리하는데서 시작되어 1970년대에 미 국방부의 표준의 일부로 작성되었다. 이후 지속적으로 발전하다가 1990년대에 들어 컴퓨팅 환경이 복잡해지고 품질 기준이 엄격해지면서 본격적으로 적용되기 시작했다. 현재의 소프트웨어 형상관리는 복잡해진 개발 환경에서 소프트웨어를 체계적으로 작성하고 관리하여 좋은 품질의 소프트웨어를 생산하고, 새로운 버전의 소프트웨어를 보다 효율적으로 만들기 위해 반드시 필요한 프로세스 중의 하나이다.

소프트웨어는 그림 1과 같이 소프트웨어 개발 사업의 어느 단계에서나 변경이 일어난다. 변경

이 일어나면 소프트웨어 개발 산출물들의 배열에 변화가 일어나고, 새로운 형상이 구성된다. “형상관리”란 이러한 소프트웨어 개발 산출물들의 배치나 구성을 관리하는 것이다.

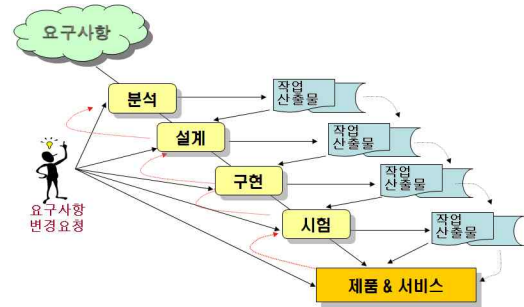


그림 1. 소프트웨어 프로세스와 변경
Fig. 1. Software process and progress

형상관리에 대한 정의는 표준들마다 다양하게 나타나고 있지만, IEEE-Std-1042의 형상관리에 대한 정의는 다음과 같다[1].

- (1) 형상 항목을 식별하여 그 기능적 물리적 특성을 문서화하고,
- (2) 그러한 특성에 대한 변경을 제어하고,
- (3) 변경 처리 상태를 기록 및 보고하고,
- (4) 명시된 요구사항에 부합하는지 검증하는 기술적이고 관리적인 감독, 감시 활동

즉, 변경할 때 관리가 필요한 항목을 선정하고, 그 항목은 아무나 변경할 수 없도록 적절한 통제를 한다. 변경한 내용을 기록하고 그 기록과 실제 변경된 내용이 일치하는지 검증하는 활동으로, 소프트웨어 사업의 수명주기 동안 제품의 무결성과 변경에 대한 추적성을 확보하기 위한 활동이라 할 수 있다.

소프트웨어 사업의 결과물은 크게 세 가지 영역으로 나눌 수 있다. (1) 컴퓨터 프로그램 (소스 코드 수준과 실행 가능한 형태 모두를 포함), (2) 컴퓨터 프로그램을 설명하는 문서 (여러 이해당사자들을 대상으로 함), (3) 데이터 또는 콘텐츠

(프로그램 내에 포함된 경우, 또는 부수적인 것). 그림 2와 같이 소프트웨어 사업의 일부로서 생산된 정보 모두를 포함하는 항목들을 소프트웨어의 형상이라 한다.

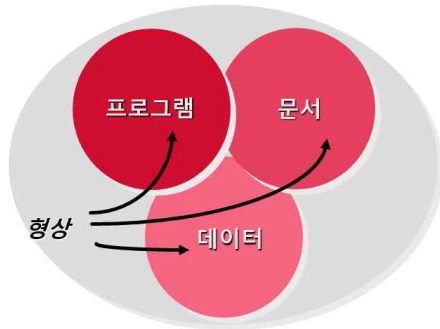


그림 2. 소프트웨어 형상
Fig. 2. Software Configuration

소프트웨어 형상 항목의 체계는 소프트웨어 프로세스로서 하나의 UML 도형으로 표현될 수 있는 소규모 항목부터 완전한 설계문서와 같은 대규모 항목까지 이름이 부여된 정보의 요소들로 생성된다. 각각의 형상 항목이 간단하게 다른 형상 항목들로부터 유도된다면 혼란은 거의 없지만, 그림 3과 같이 변경은 언제라도 일어날 수 있다.

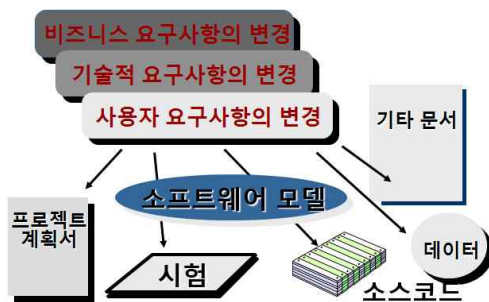


그림 3. 소프트웨어 변경
Fig. 3. Software modification

소프트웨어 형상관리는 소프트웨어 수명주기 전 과정을 통해 이루어지는 변경을 관리하기 위

해 개발된 일련의 행위이다. 소프트웨어 형상관리는 소프트웨어 사업 전반에 걸쳐 적용되는 품질보증으로 볼 수 있다.

본 논문은 서론에 이어, 2장에서 참조 표준 및 가이드, 3장에서 문서화 프로세스, 4장에서 형상관리 프로세스로 구성되고 5장에서는 연구의 결론과 향후 발전방향으로 구성되었다.

2. 참조 표준 및 가이드

2.1 ISO의 형상관리 프로세스 표준

ISO/IEC 12207 지원 수명주기 프로세스는 소프트웨어 수명주기 프로세스의 필수 구성요소로서 다른 프로세스들을 지원하고 소프트웨어 사업의 성공과 품질을 보장하기 위한 8대 프로세스들(문서화, 형상관리, 품질보증, 검증, 확인, 감리, 합동검토, 문제해결 프로세스)로 구성되어 있다. 지원 수명주기 프로세스는 핵심 수명주기 프로세스 및 조직 수명주기 프로세스와 연관관계를 가지며 이들 프로세스의 수행과정에서 필요에 의해 사용된다. 지원 수명주기 프로세스의 활동과 작업은 지원 수명주기 프로세스를 담당하는 조직의 책임이다.

지원 수명주기 프로세스의 책임을 담당하는 조직은 조직 내부에 지원 수명주기 프로세스가 존재하고 있으며, 그 프로세스가 제 기능을 발휘하고 있다는 것을 보장해야 한다.

지원 수명주기 프로세스 내의 각 프로세스가 반드시 별도의 조직에 의해 수행되어야 하는 것은 아니며, 전체 프로세스가 하나의 조직에 의해 수행될 수도 있다. 또한, 각각의 지원 수명주기 프로세스 상의 활동은 요구하는 소프트웨어 및 소프트웨어 서비스를 제공하기 위해 반복적으로 또는 순차적으로 수행이 가능하다.

ISO/IEC 12207 문서화 프로세스는 다른 수명주기 프로세스나 활동에 의하여 생산되는 정보를

기록하는 프로세스이다. 문서화 프로세스는 시스템 혹은 소프트웨어 산출물과 관련한 모든 관련자가 필요로 하는 문서에 대한 계획, 설계, 개발, 생산, 편집, 배포, 유지하는 활동을 포함한다.

ISO/IEC 12207 형상관리 프로세스는 다른 수명주기 프로세스의 필수 구성 부분이며, 형상관리를 수행하는 조직에 의해 사용되는 프로세스이다. 형상관리 프로세스는 소프트웨어 수명주기 동안 시스템 내의 소프트웨어 항목을 식별 및 정의, 소프트웨어 항목의 수정 및 배포통제, 소프트웨어 항목과 이에 대한 수정 요구의 보고 및 기록, 항목의 완전성, 일관성 및 정확도 보장, 소프트웨어 항목의 저장, 취급, 배포를 위한 관리적이고 기술적인 절차를 적용하는 프로세스이다.

2.2 PMBOK 표준

통합 변경 통제 프로세스는 프로젝트 시작부터 종료될 때까지 수행된다. 변경 통제는 프로젝트가 프로젝트 관리 계획에 따라 정확히 수행되기가 매우 힘들기 때문에 필요하다. 프로젝트 관리 계획, 프로젝트 범위 기술서 및 기타 산출물들은 반드시 주의 깊게 유지되어야 하며, 승인된 변경 사항이 수정된 기준선에 적합하도록 지속적으로 관리해야 한다. 통합 변경 통제 프로세스는 다음과 같은 활동들을 포함한다[7].

- 변경이 필요한 것과 변경이 된 것에 대한 식별
- 요청된 변경에 대한 검토 및 승인, 그리고 승인된 변경의 배포
- 요청된 변경의 흐름에 대한 규제를 통한, 승인된 변경 사항의 발생의 관리
- 기준선의 무결성의 유지 및 관련된 구성 및 계획 문서 기록의 유지
- 모든 권고된 시정 및 예방 활동에 대한 검토 및 승인

- 전체 프로젝트를 통한 변경 사항 조정으로 승인된 변경 사항에 따른 범위, 비용, 예산, 일정과 품질 요구에 대한 통제 및 갱신
- 요청된 변경의 완료 후 영향 문서화
- 결함 수정 검사
- 품질 보고서를 통한 프로젝트 품질이 표준에 적합하도록 하기 위한 통제

제안된 변경에 따라 새로운 비용 예측, 활동 순서, 일정 계획, 자원 요구사항, 위험 대응 대안의 분석 또는 이에 대한 수정이 요구될 수 있다. 이러한 변경으로 인하여 프로젝트 관리 계획, 프로젝트 범위 기술서 또는 기타 프로젝트 산출물들의 조정이 필요할 수 있다. 어느 수준의 변경 통제가 적절한가는 특정 프로젝트의 응용 영역 및 복잡도, 계약 요구사항, 프로젝트가 수행되는 환경 등에 전적으로 의존한다.

모든 문서화된 변경 요청은 프로젝트 관리 팀 내 또는 발기인, 후원자 또는 고객으로 대표되는 외부 조직 내의 권한을 가진 사람에 의해 승인 또는 반려되어야 한다. 대부분의 경우 통합 변경 통제 프로세스는 요청된 변경에 대한 승인 및 반려에 대한 책임을 가진 변경통제위원회를 포함한다. 이 위원회의 역할 및 책임은 형상통제 및 변경 통제 프로시저에서 명확히 정의되며 후원자, 고객 및 기타 이해당사자들의 동의를 받게 된다. 다수의 대형 조직은 위원회 간에 책임을 분리한 다음 다중 계층 위원회 구조를 사용한다. 만약 프로젝트가 계약 하에서 수행된다면 몇몇 변경 사항은 고객으로부터 승인이 필요할 수도 있다.

변경 통제를 포함하는 형상관리 시스템은 프로젝트 내 변경을 중앙 집중적으로 관리하기 위한 표준화되고, 효과적이고 효율적인 프로세스를 제공한다. 이러한 형상관리는 기준선의 식별, 문서화, 변경 통제를 포함한다. 형상관리의 중요 목표는 다음과 같다.

- 수립된 기준선에 대한 변경 사항의 발견과 변경 요청을 지속적으로 하며, 그러한 변경의 가치 및 효과 평가를 위한 진화적 수단을 제공한다.
- 지속적인 검사와 각 변경의 영향에 대한 연구를 통해 프로세스 개선의 기회를 제공한다.
- 프로젝트 관리 팀이 모든 변경 사항을 이해당사자들에게 지속적으로 전달할 수 있는 체계를 제공한다.

통합 변경 통제 프로세스에 포함된 형상관리 활동의 일부는 다음과 같다.

- 형상식별: 제품의 형상 중 어떤 형상이 정의되고 검증되어야 하는지, 어떤 제품 및 문서에 라벨이 붙어야 하는지, 변경을 관리해야 하는지, 책임이 유지되어야 하는지에 대한 기초를 제공한다.
- 형상상태 보고: 제품 및 제품 정보를 효과적으로 관리하기 위해 필요한 형상 정보를 확보하고, 저장하고, 접근한다.
- 형상 검증 및 감사: 형상문서에 정의된 성능 및 기능 요구사항이 충족되도록 한다.

3. 문서화 프로세스

3.1 문서화 준비

소프트웨어 수명주기 동안에 생산되는 문서화 대상을 식별하여 정의하고 문서의 작성, 생산 및 배포에 대한 계획을 작성하는 활동이다.

1) 담당자

활동명	발주자	공급자	개발자	운영자	유지보수자
문서화준비	◎	◎	◎	◎	◎

2) 입력물

- 문서화 요구사항
- 계약서

3) 출력물

- 문서화 계획서

4) 고려사항

- 모든 식별된 문서에는 다음 사항이 포함되어야 한다.
 - 제목 또는 이름
 - 목적
 - 수령인
 - 입력, 개발, 검토, 수정, 승인, 생산, 저장, 배포, 문서의 유지보수 및 형상관리를 위한 절차와 책임
 - 중간 산출물 및 최종 산출물 작성 일정

3.2 문서화 표준 개발

식별된 모든 문서에 대하여 공통으로 따를 수 있는 문서화 표준을 개발하는 활동이다.

1) 담당자

활동명	발주자	공급자	개발자	운영자	유지보수자
문서화표준개발	◎	◎	◎	◎	◎

2) 입력물

- 문서화 계획서
- 조직의 문서화 지침

3) 출력물

- 문서화 표준

4) 고려사항

- ① 모든 식별된 문서에 대하여 양식, 내용 설

명, 페이지 번호부여, 그림 및 표 배치, 소유권/보호 표시, 문서의 외관, 기타 필요한 표현 항목 등에 대한 문서화 표준을 개발해야 한다.

- ② 문서화 표준에는 요구사항, 설계, 코드, 시험 또는 기타 정보를 이해하는 데에 필요한 약속을 포함하거나 참조해야 한다.
- ③ 수명주기 프로세스를 실행하는 동안 발생하는 제반 산출물에 관한 자료의 표현 형태와 문서화에 대한 추가적인 지침은 다른 표준을 참고할 수 있다.
- ④ 문서 작성에 포함되는 모든 입력 자료에 대해서는 그 출처와 적절성을 확인해야 한다. 이를 위해 자동화된 문서화 도구를 사용할 수 있다.
- ⑤ 작성하는 문서는 개발된 문서화 표준에 따라 검토하고 편집해야 한다.
- ⑥ 문서화 표준은 모든 문서에 적용하기에 앞서 권한을 가진 사람으로부터 사용의 적절성을 승인받아야 한다.

3.3 문서 생산 및 배포

작성된 문서를 생산하고, 이를 배포하는 활동이다. 문서 생산 및 배포 활동에서는 작성된 문서를 어떠한 매체(예, 종이, 전자문서)를 통해 생산할 것인지와 어떠한 경로(예, 직접전달, 이메일 전달)를 통해 배포할 것인지를 정의한다.

1) 담당자

활동명	발주자	공급자	개발자	운영자	유지보수자
문서생산및배포	◎	◎	◎	◎	◎

2) 입력물

- 문서화 계획서
- 문서화 표준

3) 출력물

- 문서
- 배포 결과

4) 고려사항

- ① 문서를 계획에 따라 생산하고 배포해야 한다. 문서의 생산과 배포 시에는 종이, 전자 문서 혹은 기타의 매체를 사용할 수 있다. 원본은 보존, 보호, 유지보수 및 백업 등의 기준에 따라 유지해야 한다.
- ② 생산된 문서는 변경통제를 위해 『형상관리』 프로세스에 따라 관리되어야 한다. 모든 식별된 문서에 대하여 양식, 내용 설명, 페이지 번호부여, 그림 및 표 배치, 소유권/보호 표시, 문서의 외관, 기타 필요한 표현 항목 등에 대한 문서화 표준을 개발해야 한다.

3.4 문서 유지보수

문서 유지보수 활동은 작성되어 배포된 문서를 사용자 요구사항, 소프트웨어 설계 및 개발 정보의 변화에 맞추어 내용 변경사항이나 보완사항이 적시에 반영될 수 있도록 최신화하는 활동이다. 문서 유지보수 활동에서는 언제, 누가, 어떠한 이유로, 어느 부분을 변경했는지에 대한 모든 정보를 기록하고 관리한다.

1) 담당자

활동명	발주자	공급자	개발자	운영자	유지보수자
문서 유지보수	◎	◎	◎	◎	◎

2) 입력물

- 문서
- 배포 결과

3) 출력물

■ 문서(수정)

4) 고려사항

- ① 변경 요구가 발생하였을 경우, 문서를 수정해야 한다. 예를 들어, 기배포된 문서에 대한 수정이 요구될 때, 모든 관련 조직에게 통보를 해야 한다.
- ② 형상항목으로 관리 중인 문서에 대해서는 『형상관리』 프로세스에 따라 이력을 관리해야 한다. 문서를 계획에 따라 생산하고 배포해야 한다. 문서의 생산과 배포 시에는 종이, 전자문서 혹은 기타의 매체를 사용할 수 있다. 원본은 보존, 보호, 유지보수 및 백업 등의 기준에 따라 유지해야 한다.

4. 형상관리 프로세스

4.1 형상관리 준비

형상관리 준비는 형상관리를 수행하기 위한 계획을 정의하는 활동이다. 형상관리 준비 활동에서는 소프트웨어 사업의 모든 형상항목에 대하여 형상을 식별하는 것부터 시작하여 형상변경에 대한 통제 및 평가 등의 제반사항을 어떠한 절차에 따라 수행할 것인지를 사업 관점에서 정의한다.

형상관리 계획은 형상관리를 위해 사용되어야 하는 표준과 절차를 기술한다. 형상관리 계획을 세우기 위한 출발점은 형상관리 표준의 집합이어야 하고, 이 표준은 각 특정 프로젝트의 요구사항과 제약 조건에 적합하도록 수정되어야 한다. 형상관리 계획서는 여러 절로 구성되어야 한다 [9].

- 관리되어야 하는 형상항목과 이 개체를 식별하기 위해 사용해야 하는 방식을 정의한다.

- 형상관리 절차와 형상관리팀에 제어되는 개체를 제출하는 것에 대해서 책임을 지는 사람을 선정한다.
- 형상통제와 형상 이력관리를 위해 모든 팀 구성원이 사용해야 하는 형상관리 정책을 정의한다.
- 형상관리를 위해 사용하는 도구와 이 도구를 사용하는 프로세스를 명시한다.
- 형상정보를 기록하기 위해 사용하는 형상 데이터베이스 구조와 이 데이터베이스에 유지해야 하는 정보를 기술한다.

또한 외부에서 공급된 소프트웨어 관리와 같은 다른 정보 및 형상관리 프로세스에 관한 형상평가 절차가 형상관리 계획에 포함될 수 있다.

형상관리 계획의 중요한 부분은 책임을 정의하는 것이다. 형상관리 계획은 각 문서나 소프트웨어 컴포넌트를 품질보증과 형상관리로 전달하는 책임을 지는 사람을 정의해야 한다.

4.2 형상식별

형상식별은 소프트웨어 사업에서 통제해야 하는 소프트웨어 형상항목과 그 버전을 식별하는 활동이다. 형상식별 활동에서는 형상관리 계획에 정의된 절차 및 기준에 따라 소프트웨어 형상항목과 그 버전에 대하여 기준선이 되는 문서, 참조 버전 및 기타 세부사항 등을 식별한다. 형상식별 활동은 형상관리의 밑바탕이 되는 활동으로, 형상관리의 대상을 식별하고 기준선을 설정하는 활동이다. 형상식별 활동은 크게 형상항목 선정과 기준선 설정의 세부 활동으로 이루어져 있다.

소프트웨어를 개발하면 많은 개발 문서와 소스 코드, 이미지 등의 산출물이 발생한다. 이러한 산출물 중 계속해서 수정되는 계획서나 소스 코드는 수정에 대비한 적절한 저장 방법과 접근 방

침이 필요하다. 또한 그 중 고객과 합의한 요구 사항 명세서와 같이 임의로 변경하면 안 되는 문서들은 적절한 통제를 통하여 관리되어야 한다.

이처럼 관리 방법이나 변경에 대한 통제 여부에 따라 산출물을 구분하고, 이 중 변경에 대한 통제가 필요한 산출물을 선정하는 것을 형상항목 선정이라고 한다.

형상항목의 선정은 제품 개발 초기 단계에서 프로젝트 관리자가 형상관리자를 정하고, 그와 함께 형상관리 대상이 되는 형상항목을 정함으로써 이루어진다.

형상항목에 고유의 이름과 번호를 정하고, 버전을 부여하는 식별 체계를 형상 식별자라 한다. 형상 식별자는 산출물의 파일 이름으로 이용되며, 이는 일정한 법칙을 가지고 유지되어야 한다.

Project id	Doc. id	Version id	Draft id
------------	---------	------------	----------

그림 4. 형상 식별자의 예
Fig. 4. Configuration identification

그림 4는 형상 식별자의 예를 나타낸다. 첫 번째 필드에는 프로젝트 고유 ID가 기록되고, 그 다음은 산출물의 ID, 버전 ID, 드래프트 ID 형식으로 쓰여 형상을 구분하도록 하였다. 만약 “SELAB_SRS_1.0.0_C”라는 식별자를 가진 형상항목은 SELAB이라는 프로젝트의 요구사항 명세서이며, 1.0.0 버전을 만들어가는 C 단계라고 구분할 수 있다. 이러한 형상 식별자는 파일명으로 분석의 종류와 버전을 쉽게 알 수 있다는 장점이 있다. 만약 버전관리 도구를 사용한다면, 버전 ID는 버전관리 도구에서 자동으로 관리해 주기도 한다. 이럴 경우 버전 ID는 불일 필요가 없다[3].

일반적으로 기준선은 소프트웨어 개발의 특정 시점에서 형상항목이 소프트웨어 개발에 하나의 완전한 산출물로써 이용될 수 있는 상태(1.0 버

전으로 표현하기도 한다)의 집합을 나타낸다. 이러한 기준선 상태가 되기 위해서는 형상통제위원회의 공식적인 검토와 승인을 거쳐야 한다.

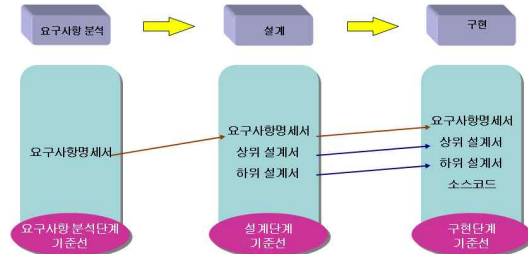


그림 5. 기준선 설정 단계
Fig. 5. Start-up process of basement

기준선은 그림 5와 같이 일반적으로 소프트웨어 수명주기를 단위로 설정되는데, 예를 들면 요구사항 분석 단계가 끝나면 요구사항 명세서를, 설계 단계 후에는 이전 단계의 기준선에 설계 문서를, 구현 단계 후에는 소스 코드를 기준선으로 포함시키는 경우이다. 이렇게 진행이 된다면 구현 단계가 끝난 후의 기준선은 그림 6과 같이 요구사항 명세서, 설계 문서, 소스 코드가 된다.

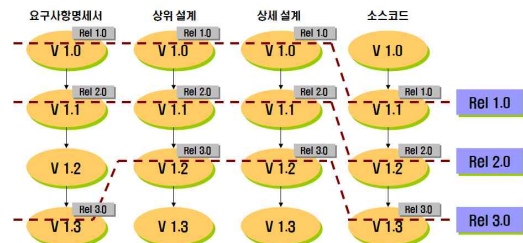


그림 6. 구현 단계 후 기준선 예제
Fig. 6. Baseline by configuration

이와 같은 기준선은 형상관리계획서 작성 시 설정되고, 기준선이 될 형상항목의 선정은 형상통제위원회에서 결정하게 된다.

기준선은 개발에 중요한 산출물에 대하여 적절한 통제를 통해 변경을 관리하기 위한 목적인

만큼, 변경을 위해서는 형상통제위원회의 평가와 승인이 필요하며, 이러한 모든 과정은 공식적인 절차에 의해 이루어진다.

4.3 형상통제

요구사항은 언제나 변하기 마련이다. 모든 요구사항의 변경은 관련 산출물의 변경을 수반한다. 만약 개발 후반부에 요구사항이 변경된다면 소스 코드뿐만 아니라 개발 전반부에서 만든 요구사항 문서부터 설계 문서까지 변경해야 한다. 이것은 비용과 납기일과 관련된 중요한 문제이다. 그러므로 변경을 위해서는 엄격한 심사 프로세스가 있어야 한다. 고객의 요구사항 변경을 수용할 것인지 평가하고, 관련자들이 모여 결정하는 절차를 형상통제 프로세스라고 한다.

다시 말하면 형상통제는 식별된 형상항목에 대하여 변경에 대한 내역 및 영향 분석, 평가, 승인 및 배포 등을 수행하기 위한 활동이다.

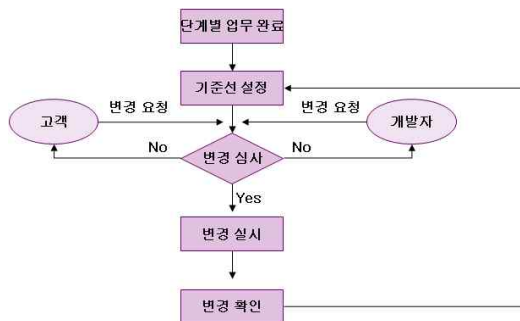


그림 7. 형상통제 절차

Fig. 7. Configuration control progress

그림 7은 일반적인 형상통제의 절차를 나타내고 있다. 기준선에 대해 고객이나 개발자의 변경 요청이 있을 경우 형상통제 절차가 시작된다. 변경 요청은 미리 지정한 양식에 맞추어 형상관리자에게 제출되어야 하며, 형상관리자는 이를 토대로 형상통제 절차를 따르게 된다.

4.4 형상평가

형상평가는 식별된 소프트웨어 형상통제 항목에 대한 요구사항 관점에서의 기능적 완전성과 물리적 완전성에 대한 평가를 수행하고, 결과를 문서화하는 활동으로, 그림 8과 같이 형상관리계획서대로 형상관리가 진행되고 있는지, 요구사항 문서대로 제품이 제작되었는지 감사하는 활동이다.

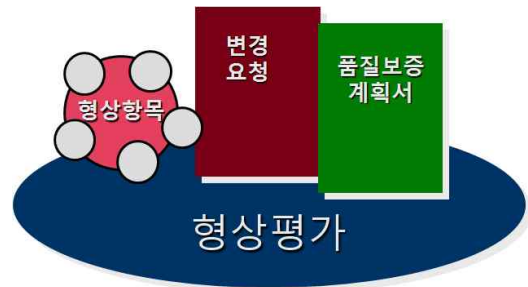


그림 8. 형상평가

Fig. 8. Configuration evaluation

형상평가의 목적은 형상관리 활동이 형상관리 프로젝트 또는 형상관리계획서에 따라 수행되는지 확인하여 기준선의 무결성을 검증하는데 있다. 형상관리자가 실시하며, 형상관리계획서에 형상평가에 대한 계획이 수립되어 있어야 한다. 그리고 정기적인 형상평가 외에 기준선이 수립되기 전에도 실시한다.

형상관리자는 사전에 준비한 체크 리스트를 통하여 감사를 실시하며, 검증하는 내용은 다음과 같다.

- 모든 승인된 변경의 반영 여부
- 관련된 항목의 갱신 여부
- 승인되지 않은 변경의 반영 여부
- 인도될 항목과 요구사항, 설계와의 일치여부
- 모든 알려진 소프트웨어의 문제가 변경 승인에 의해 검토되었는지 여부
- 버전 기술 문서의 준비 여부

4.5 배포관리 및 인도

형상평가는 소프트웨어 산출물과 관련 문서의 배포 및 인도 과정을 공식적으로 통제하고, 결과를 문서화하는 활동으로, 그림 9와 같이 기준선이 지금 어떤 상태인지, 변경한 항목들은 잘 반영되고 있는지 보고하는 절차이다.



그림 9. 배포관리 및 인도
Fig. 9. Distribution management

배포관리 및 인도의 목적은 그림 9와 같이 기준선으로 설정된 형상항목의 구조와 변경상태를 기록하여 보고함으로써, 형상항목의 개발 상태에 대한 가시성을 제공하는데 있다. 프로젝트 진행 과정상 제품의 현재 개발 상황, 형상항목의 변경 내역을 모니터링하고 감독하는 활동으로서, 형상상태 보고서를 기반으로 하여 현재 프로젝트 저장소에 저장되어 있는 형상항목들이 계획대로 유지되고 있는지 혹은 개발자들의 비공식적인 형상항목 변경이 있는지를 파악하는 활동이다.

배포관리 및 인도는 형상관리계획서에서 정한 주기대로 형상상태보고서를 작성하여 실시하되, 분기를 넘지 않도록 해야 한다. 또한 형상상태보고서는 상위 관리자에게 보고하여 형상상태와 관련된 이슈 발생 시 적절한 조치를 취해야 한다. 형상상태보고서의 주요 내용으로는 기준선이 상태, 형상통제 상태, 형상통제위원회 활동 내역,

변경요청의 상태 등이 있으며 형상관리자가 작성한다. 식별된 소프트웨어 형상통제 항목에 대한 요구사항 관점에서의 기능적 완전성과 물리적 완전성에 대한 평가를 수행하고, 결과를 문서화하는 활동이다.

5. 결 론

전형적인 형상관리 활동에는 소프트웨어 사업을 책임지는 프로젝트 관리자, 형상관리의 절차 및 정책을 책임지는 형상관리자, 소프트웨어 제품의 개발 및 유지보수를 책임지는 소프트웨어 엔지니어, 그리고 제품을 사용하는 고객 모두가 관련된다.

프로젝트 관리자의 목표는 제품이 정해진 기일 내에 개발이 완료될 것을 보장하는 것이다. 그러므로 관리자는 개발 프로세스를 주시하여 문제를 인식하고 그에 대응한다. 이러한 일은 소프트웨어 시스템의 상태 보고서의 분석과 시스템에 대한 검토를 통해 이루어진다.

형상관리자의 목표는 프로젝트에 관한 정보에 접근할 수 있게 하는 것뿐만 아니라 코드의 생성, 변경 및 시험에 관한 절차와 정책을 확인하는 것이다. 소스 코드의 변경에 대한 제어를 유지하기 위해 형상관리자는 변경을 공식적으로 요구하고, 소프트웨어 시스템의 변경을 허가하는 책임을 가진 형상통제위원회를 통하여 변경사항을 평가하며, 변경 권한을 부여하는 방법을 채택한다. 형상관리자는 엔지니어들에게 업무 목록을 작성하여 배포하고 기본적인 프로젝트의 상황을 작성한다. 또한 형상관리자는 소프트웨어 시스템의 어느 컴포넌트가 문제를 야기하는지를 결정하는 정보를 비롯한 소프트웨어 시스템의 컴포넌트에 관한 통계자료를 수집한다.

소프트웨어 엔지니어들의 목표는 효과적으로 업무를 수행하는 것이다. 이는 코드를 작성하

나 시험하는데, 또한 제품의 생산을 지원하는데 다른 팀원들을 불필요하게 방해하지 말아야 할 것을 의미한다. 동시에 그들은 효율적으로 의사소통을 하고 협조를 하여야 한다. 특히 엔지니어들은 안정된 소프트웨어 제품을 생산하는데 도움이 되는 도구들을 사용한다. 변경은 융합되면서 다른 팀원들의 업무에까지 전파된다. 동시에 변경된 컴포넌트들이 모순된 상태에 있지 않도록 조정하고 변경사항들을 검증한다. 변경 및 그 이유에 대한 기록과 실제로 변경된 내용에 대한 보고서와 함께 소프트웨어 시스템 내의 모든 컴포넌트의 변경 이력이 보관된다. 엔지니어는 코드를 작성, 변경, 시험하고 통합하기 위한 작업공간이 있다. 어느 시점에서 소스 코드는 기준선을 기준으로 개발이 계속되고, 기준선으로부터 변경이 이루어진다.

고객은 소프트웨어 제품을 사용한다. 소프트웨어 제품이 형상관리 제어를 받고 있으므로 고객은 변경을 요구하는 정형적 절차를 따르고 제품의 결함을 지적한다.

프로젝트 관리자는 형상관리를 감사의 수단으로 보고, 형상관리자는 형상관리를 제어, 추적 및 정책결정의 수단으로 보며, 소프트웨어 엔지니어는 형상관리를 변경, 작성 및 접근 제어의 수단으로 보고, 고객은 형상관리를 품질보증의 수단으로 본다.

소프트웨어 사업에 내재되어 있는 형상관리와 관련된 문제점들을 살펴보면 다음과 같다.

- 요구사항의 변화가 많다: 모든 요구사항을 다 변경할 것인가? 그로 인해 필요해지는 비용과 시간은 어떻게 할 것인가?
- 산출물에 대한 수정 결과가 관련자들에게 제대로 통보되지 않는다: 개발자 A가 변경한 내용을 개발자 B가 모르고 있고, 제작업 또는 덮어 쓰기를 할 수 있다.

- 많은 개발자들이 동일한 산출물에 대하여 개별적으로 이중 작업을 실시한다: 개발자 A가 변경한 산출물을 개발자 B도 동시에 수정하고 있다.
- 하나의 산출물이 여러 개의 사본으로 존재하여 작업에 혼란을 초래한다: 서버에 통합하여 관리하는 것이 아니기 때문에 어느 사본이 최신 버전인지, 어느 사본에서 모든 기능이 동작하는지 알 수 없다.

즉, 소프트웨어의 비가시성, 변경 추적의 어려움, 관리와 통제의 어려움, 요구사항 변경으로 인한 잦은 수정작업의 발생 등으로 초래할 수 있는 위험을 최소화하기 위한 활동이 필요하다.

참 고 문 헌

- [1] ANSI/IEEE, IEEE-Std 1042, "IEEE Guide to Software Configuration Management", The Institute of Electrical and Electronics Engineers, Inc., 1998.
- [2] Rogers S. Pressman, "Software Engineering: A Practitioners's Approach 7th Ed", McGraw Hill, 2011.
- [3] 한혁수, "소프트웨어공학의 소개 개정증보판", 홍릉과학출판사, 2008.
- [4] IEEE/EIA 12207.0, Software Life Cycle Processes, Clause 5.3: Development process.
- [5] 고석하, 홍성유, "소프트웨어 프로젝트 관리", 생능출판사, 2007.
- [6] Pankaj Jalote, "CMM in Practice", Addison-Wesley, 2000.
- [7] 고석하, 홍성유, "소프트웨어 프로젝트 관리 개정판", 생능출판사, 2008.
- [8] 정보통신부, "소프트웨어 사업 관리 감독에 관한 세부지침 v.1.0", 2006.
- [9] Ian Sommerville, "Software Engineering 8th Ed", Addison-Wesley, 2009.

- [10] ANSI/IEEE, IEEE-Std-610, “IEEE Standard Glossary of Software Engineering Terminology”, The Institute of Electrical and Electronics Engineers Inc., 1990.
- [11] CMU SEI, “Capability Maturity Model Integration(CMMI) Version 1.3”, <http://www.sei.cmu.edu>, Feb. 2011.
- [12] NASA, “NASA Software Documentation Standard”, NASA-STD-2100-91, 1991.
- [13] 지식경제부, “SW 분리발주 가이드라인”, 지식경제부, 2007.05.
- [14] DoD, “Software Development and Documentation”, DoD-STD-498, 1994.

저 자 소 개



권기태(Ki-Tae Kwon)

1986년 서울대학교 계산통계학과 졸업
1988년 서울대학교 계산통계학과 석사 졸업
1993년 서울대학교 계산통계학과 박사 졸업
1996년 미국 Univ. of Southern California,
전산학과 Post-Doc.
현 재 강릉원주대학교 컴퓨터공학과 교수

<주관심분야 : 소프트웨어공학, 데이터마이닝, 지능시스템>