

발 간 등 록 번 호

11-1410119-000038-01

높은 품질의 소프트웨어를, 더 빨리, 더 적은 비용으로 개발하기 위한

# ISO/IEC15504(SPICE) 기반의 중소기업을 위한 소프트웨어 표준 프로세스

2007

산업자원부 기술표준원

# 서 문

## 본 가이드라인의 목적

본 가이드라인은 중소 소프트웨어 개발 기업에 소프트웨어 표준 프로세스 예제를 제공하여 자체 표준 프로세스 정립과 프로세스 개선을 할 수 있도록 하는 것이다. 이를 통해 중소기업이 **높은 품질의 소프트웨어를(Better), 더 빨리(Faster), 적은비용(Cheaper)**으로 개발하여 고객 만족과 경쟁력 향상을 할 수 있도록 하는 것이다.

이미 서문에서 언급한 바와 같이 ISO/IEC 15504 표준을 산업 현장에서 사용하기 위해서는 프로세스 심사에서 사용되는 심사모델인 ISO/IEC 15504-5에 따라 기업 실정에 맞는 프로세스를 정의한 "표준 프로세스"가 필요하다. 그러나 심사모델은 말 그대로 프로세스 심사를 위한 표준이지 산업현장에서는 일상적으로 사용할 수 있는 표준이 아니다. 즉, 심사모델은 어떠한 활동(프랙티스)을 수행했는가를 심사하기 위해 만든 표준으로 "What"만을 규정하고 있어 기업 현장에서 사용하기 위해 실제로 필요한 "How"는 제공하지 않는다. 본 가이드라인은 ISO/IEC 15504-5 심사모델 표준에 맞게 일하는 "How"를 제공한다. 강조하여 표현하면, **본 가이드라인은 심사모델이 아니라 중소기업이 거의 수정 없이 소프트웨어 개발 및 유지보수 프로젝트에서 사용가능한 표준 프로세스를 제공한다.**

국제적으로 소프트웨어 프로세스 심사표준은 20여 가지가 넘는 종류가 존재 했었으나, 현재는 ISO/IEC 15504-5와 CMMI for Development (v1.2)만이 국제적으로 통용되고 있다. 본 가이드라인에서 프로세스는 ISO/IEC 15504-5의 정의를 따랐으며, 'How'를 제공하기 위한 설명에서는 CMMI-DEV, IEEE 소프트웨어 공학 표준, ISO 9001 등의 내용이 필요시 사용되었다. 본 가이드라인에서 제공하는 표준 프로세스는 ISO/IEC15504 표준에 따른 프로세스 심사와 개선을 수행하는 기업뿐만 아니라 CMM-DEVI (v1.2)에 따라 프로세스 심사와 개선을 실시하는 기업에서도 사용할 수 있다. 또한 표준 프로세스를 만들어 사용하는 기업은 본 가이드라인의 내용과 벤치마킹해보는 것도 프로세스 개선에 도움이 될 수 있다고 여겨진다.

본 가이드라인에서는 편의상 ISO/IEC 15504 및 SPICE<sup>1)</sup>와 혼용하여 사용하고 있으며, 표기의 편의성을 위하여 ISO/IEC 15504의 표기에서는 ISO/IEC라는 것은 제외하였다. 또한 15504 각 부(Part)를 15504-x로 표기한다(현재 x=1, 2, 3, 4, 5, 6, 7, 8).

---

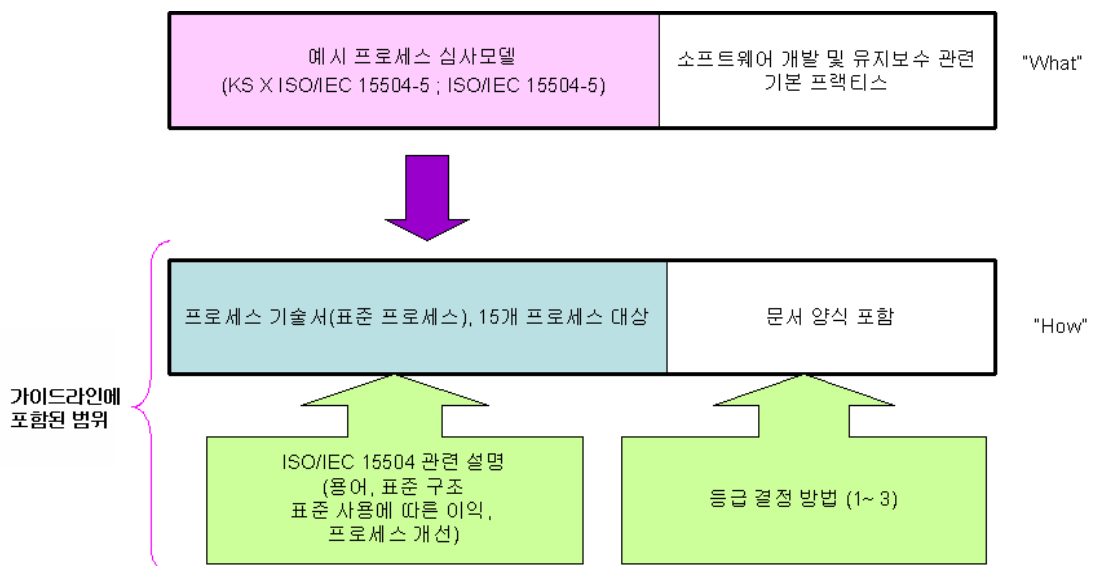
1) SPICE는 Software Process Improvement and Capability dEtermination의 약자로 통상 ISO/IEC 15504 표준을 통칭합니다.

## 왜 중소기업인가?

중소기업은 고용과 경제 안정성을 위하여 매우 중요하다. 국내 약 6000여 개 이상의 소프트웨어 개발업체 중 90% 이상이 중소기업이고, 소프트웨어를 개발하기에 충분한 개발체계를 갖추고 있지 못한 것으로 알려져 있다. 또한 중소기업은 대기업의 협력업체로 대형 프로젝트에 참여하고 있어, 중소기업의 소프트웨어 프로세스 개선 없이는 대기업도 좋은 품질의 소프트웨어를 빠르게 적은 비용으로 만들어 내기 어려운 것이 사실이다. 또한 언어가 다르면 의사소통이 불가능 하듯이, 여러 기업이 공동으로 참여하는 프로젝트에서 용어 표준이 없으면 원만한 의사소통이 어렵다. 본 가이드라인은 이러한 문제점을 해결하는데 도움을 줄 수 있다.

## 본 가이드라인의 구성

본 가이드라인을 쓰면서 주의를 기울인 것 중에 하나는 가벼운 책을 만들자는 것이었으나 원래의 의도보다는 다소 두꺼운 책이 되었다. 본 가이드라인은 크게 2개의 부와 1개의 부속서로 구성되어 있다. 제 1 부 프로세스 모델에서는 15504 표준의 사용과 심사를 이해하는데 필요한 기본 사항(그림의 맨 아래 부분)을, 제 2 부에서는 15개의 표준 프로세스(그림의 중간)를 제공하고 있다. 마지막으로 부속서 A에서는 프로세스 수행에 필요하다고 여겨지는 작업산출물과 프로세스 수행에 따라 나올 수 있다고 여겨지는 작업산출물의 내용(특성)을 제공한다.



좀 더 세부적으로 살펴보면, 제 1 부에서 제 1 장은 15504 표준화 배경, 15504 표준의 종류, 프로세스 개선에 따라 기대되는 성과를 다루고 있다. 제 2 장에서는 소프트웨어 예시심사모델인 15504-5의 구조, 능력 수준 결정방법을 제 3 장에서는 프로세스 능력수준 결정에 사용되는 지표를 다루고 있다. 제 3 장에서는 15504-5 (예시 심사모형)에 대한 2차원 구조, 능력 수준 결정 방법, 지표를 설명하고 있다. 제 4 장은 프로세스 개선의 단계를 소개하고 프로세스 개선과 심사의 관계를 설명하고 있다.

제 2 부에서는 15504-5에서 규정한 48개 프로세스 중 14개 프로세스를 선정하여 중소기업에서 수정 없이 사용할 수 있도록 만든 표준 프로세스이다. 이 14개의 프로세스는 기업이 프로세스 활동을 시작할 때 처음으로 시작해야 하는 핵심적인 프로세스라고 인정된 것들로 구성되었다. 이 중 프로젝트 관리 프로세스(MAN.3)는 그 포함 내용이 광범위하기 때문에, 프로젝트 계획 수립 프로세스 (MAN.3-1)와 프로젝트 진행 관리 프로세스 (MAN.3-2)로 나누었다. 결과적으로 아래와 같이 총 15개의 프로세스를 대상으로 표준 프로세스를 작성하였다.

ENG.1 요구사항 도출 프로세스	MAN.3-1 프로젝트 계획 수립 프로세스
ENG.4 소프트웨어 요구사항 분석 프로세스	MAN.3-2 프로젝트 진행 관리 프로세스
ENG.5 소프트웨어 설계 프로세스	MAN.5 위험 관리 프로세스
ENG.6 소프트웨어 구축 프로세스	
ENG.7 소프트웨어 통합 프로세스	SUP.1 품질 보증
ENG.8 소프트웨어 시험 프로세스	SUP.8 형상 관리 프로세스
ENG.11 소프트웨어 설치	SUP.9 문제 해결 관리 프로세스
ENG.12 소프트웨어 및 시스템 유지관리	SUP.10 변경 요청 관리 프로세스

부속서 A는 표준 프로세스에서 인용한 작업산출물들이 포함하여야 할 내용을 정의한 것으로서, 15504-5를 준해서 인용하였다. 이 작업산출물의 내용을 참조하여 템플릿으로 작성하여 활용하면 많은 도움이 될 것이다. 15504에서는 5개의 능력수준(CL: Capability Level)을 정의하고 있으나 본 가이드라인에서는 대부분의 중소기업 현황을 고려하여 능력수준 3까지만 정의한다. 본 가이드라인은 단순화를 위하여 가능하면 본문에서는 참고문헌을 인용하지 않고 있다.

## 참 고

ISO/IEC 15504-5는 소프트웨어 개발 및 유지보수 프로세스 심사를 위한 표준으로, 기업은 이 표준에서 정의한 What인 기본 프랙티스에 맞게 How인 표준 프로세스를 만들 수 있다. 기업의 모든 프로젝트는 표준 프로세스를 조정하여(tailing) 프로젝트마다 정의된 프로세스(defined process)를 만들고 이에 따라 프로젝트를 수행한다. 표준 프로세스나 정의된 프로세스가 존재하는 능력 수준은 3이다.

능력수준 2에서도 프로젝트를 수행하기 위해서는 조직 수준의 표준 프로세스는 없지만 프로젝트 수행을 위한 프로세스가 있을 수 있다. 이를 프로세스 기술서라고 부른다. 능력수준 3에서는 프로세스 기술서와 정의된 프로세스가 같은 의미로 사용된다. 참고로 ISO에서 원래 프로세스 기술서란 프로세스 참고모형들의 작성 원칙이라 할 수 있으며, 그 내용은 ISO/IEC 24774 (Life Cycle Management - Guidelines for process definition)서 정의하고 있다. SPICE에서 프로세스 기술서, 표준 프로세스, 정의된 프로세스의 정의는 CMMI-DEV와 같다.

본 가이드라인에서 제공하는 표준 프로세스의 목차는 다음과 같다.

### 프로세스 이름

1. 개요
  - 1.1 목적
  - 1.2 설명
  - 1.3. 주요 관련 프로세스
  - 1.4. 용어 정의
2. 프로세스 구성
  - 2.1. 프로세스 요약  
(선행조건, 입력물, 수행활동, 출력물, 완료조건)
  - 2.2. 책임과 역할
3. 프로세스 활동

기본 프랙티스마다 "어떻게(How)"를 설명하고(☞로 표시) 예상되는 산출물 명기산출물은 ☑로 표시하고, 참조 편의를 위해 15504 표준의 번호 체계(예: 03-09)를 따른다.

4. 수행결과
5. 측정항목
6. 프로세스 조정
7. 참고문헌

---

# 목 차

제 1 장 프로세스 심사 표준: ISO/IEC15504 .....	3
1.1 표준화 배경 .....	3
1.2 ISO/IEC 15504 표준 이란? .....	3
1.3 소프트웨어 프로세스 개선의 성과 .....	5
제 2 장 ISO/IEC 15504 - 5 (예시 심사모델) .....	9
2.1 예시 심사모델이란? .....	9
2.2 ISO/IEC 15504-5의 구조 .....	9
2.3 프로세스 능력 결정 방법 .....	14
제 3 장 프로세스 속성의 등급 결정에 사용되는 지표 .....	16
3.1 PA 1.1 등급 부여에 사용되는 ‘프로세스 수행 지표’ .....	17
3.2 PA 2.1 ~ PA 5.2 등급 부여에 사용되는 ‘프로세스 능력지표’ .....	18
3.3 프로세스 수행 및 능력 지표 .....	19
제 4 장 소프트웨어 프로세스 개선과 심사 .....	25
제 5 장 엔지니어링 프로세스 그룹(ENG) .....	31
제 6 장 관리 프로세스 그룹 (MAN) .....	77
제 7 장 지원 프로세스 그룹(SUP) .....	105
부속서 A. 작업산출물 특성 .....	139
A.1 공통 작업산출물 .....	140
A.2 공통 및 고유 작업산출물 .....	143

---

# 그림 목 차

그림 1: ISO/IEC 15504 표준들의 관계 .....	4
그림 2: 프로세스 심사모델의 2차원 구조 .....	10
그림 3: 15504 - 5 프로세스 범주 및 그룹 .....	11
그림 4: 15504 - 5에서 프로세스 정의 .....	12
그림 5: 능력수준을 결정하는 지표 .....	16
그림 6: 능력수준 결정 단계 .....	19
그림 7: 프로세스 개선의 단계 (15504-4) .....	25

# 표 목 차

표 1: ISO/IEC 15504 표준 사용에 따른 성과의 예 .....	7
표 2: 프로세스 능력수준 및 프로세스 속성 정의 .....	13
표 3: 프로세스 속성의 등급판정 .....	14
표 4: 능력수준 판정 방법 .....	14
표 5: PA 2.1의 공통지표, 공통 자원, 공통 작업산출물 .....	20
표 6: PA 2.2의 공통지표, 공통 자원, 공통 작업산출물 .....	21
표 7: PA 3.1의 공통지표, 공통 자원, 공통 작업산출물 .....	22
표 8: PA 3.2의 공통지표, 공통 자원, 공통 작업산출물 .....	24

---

# 제 1 부 : 프로세스 모델



## 제 1 장 프로세스 심사 표준: ISO/IEC15504

### 1.1 표준화배경

우리나라는 그 동안 IT 분야에서 눈부신 발전을 이루어 왔다. 이에 비례하여 개발되는 소프트웨어 시스템 역시 대형화되고 복잡해지고 있다. 해외 조사에 따르면 대형 소프트웨어 프로젝트의 1%만이 계획된 기간과 예상 비용한도 내에서 고객을 만족시키며 완료된 반면, 대부분의 프로젝트들은 1년 이상 일정이 지연되고 초기 예상 비용의 2배 정도의 예산이 초과한 것으로 나타나 있다. 또한 해외에서는 수많은 소프트웨어 결함으로 인한 품질 저하는 고객 불만을 넘어서 법적인 책임 문제로까지 거론되는 상황에 이르고 있다. 시장 경쟁에서 성공하기 위해서는 높은 품질의 소프트웨어를(better quality), 더 빨리(faster), 적은 비용(Cheaper)으로 개발하는 것이다. 이는 본 가이드라인의 주제이기도하다.

품질 낮은 소프트웨어, 비용증대, 일정 지연과 같은 문제를 해결하기 위한 방안은 여러 가지가 있을 수 있으나, 소프트웨어 프로세스를 개선(SPI: Software Process Improvement)하는 것이 가장 효과적인 것으로 알려져 있다. SPI를 통해 소프트웨어 개발과 관련된 위와 같은 문제점을 완화하고자 하는 노력은 크게 두 축으로 발전해 왔다. 첫째는 미 국방성(DoD) 지원에 따라 카네기멜론 대학의 SEI (Software Engineering Institute)가 개발한 CMMI-DEV(Capability Maturity Model Integration for Development, 버전 1.2)이다. 둘째는 국제표준화 기구인 ISO(International Organization for Standardization)에서 개발한 15504로 현재 5개 문서가 국제표준으로 확정되었으며 3개가 개발중이다. 이 모두를 통칭해 SPICE(Software/Standard Process Improvement and Capability determination) 표준이라고 부르기도 한다. 이외에도 국가 또는 단체가 제정해 사용해 오던 프로세스 표준이 20여 가지가 있었다. 그러나 현재는 국제적으로 SPICE나 CMMI-DEV 만이 사용된다고 할 수 있다.

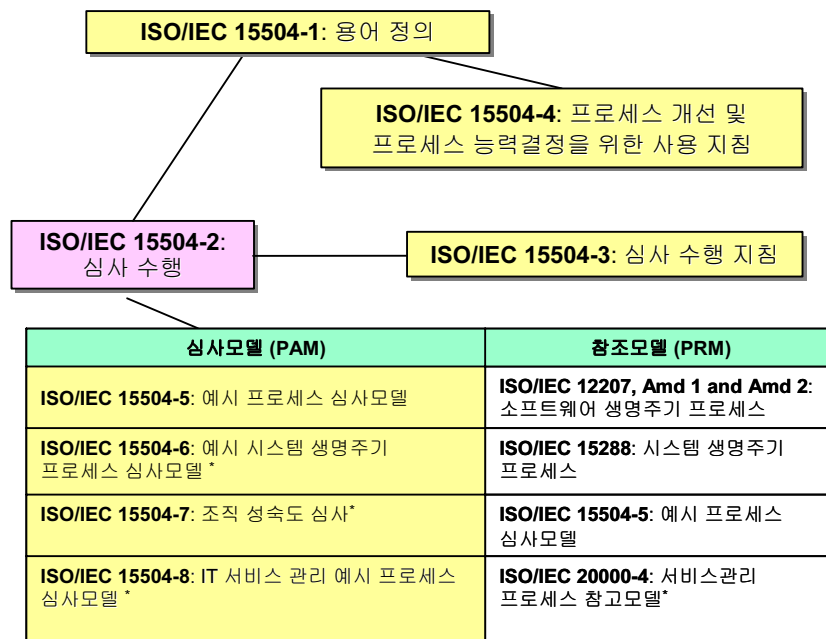
본 가이드라인은 SPICE를 근간으로 하고 있어 앞으로 SPICE를 중심으로 설명한다. 그러나 SPICE나 CMMI-DEV가 추구하는 목적은 같다는 것을 밝혀둔다. 따라서 서문에서 밝혔듯이 본 가이드라인은 SPICE와 CMMI-DEV에 무관하게 프로세스를 수립하거나 개선할 때 사용할 수 있다.

### 1.2 ISO/IEC 15504 표준이란?

SPICE라고 불리는 15504 표준은 현재 5개의 국제표준으로 구성되어 있다. 이 5개의 표준은 한글로 번역되어 우리나라 국가표준인 KS로 2006년 12월에 제정되었

다(우리나라의 KS는 "표준"이라는 용어 대신에 "규격"이라는 용어를 사용한다). 본 가이드라인에서는 이 두 용어를 혼용하여 사용하며, 그 의미는 같다. 참고로 ISO/IEC 15504 표준 뒤에 - 표시 다음에 숫자는 영어로 Part라고 하며, KS에서는 "부"라는 표현을 사용한다. 예를 들면, ISO/IEC 15504-5는 "ISO/IEC 15504: Part 5" 라고 쓰기도하고, KS에서는 "KS X ISO/IEC 15504: 5부" 또는 "KS X ISO/IEC 15504-5부" 라고 쓰기도 한다.

그림 1은 국제표준으로 확정된 문서와 개발 중인 표준을 보여주고 있다. 현재 국제 표준이고 KS 규격인 문서는 5개로 ISO/IEC 15504-1(KS X ISO/IEC 15504-1)부터 ISO/IEC 15504-5(KS X ISO/IEC 15504-5)까지이다.



\* 개발중인 표준.

그림 1: ISO/IEC 15504 표준들의 관계

그림 1에서 15504-2만이 반드시 준수하여야 하는 표준이며, 나머지는 표준 분류상 지침(Guide)이다. 여기에서 지침은 권고사항 이라기보다는 15504-2에서 규정한 프로세스 표준에 관한 원칙에 따라 만들어 졌다고 해석하여 SPICE의 모든 문서는 표준이라고 부른다. 15504-2에서는 원칙만을 규정했기 때문에 이를 해석하는데 도움을 주고자 설명한 표준이 15504-3이다.

그림 1에서 프로세스 참조모델 (PRM: Process Reference Model)은 생명주기 내의 프로세스를 정의한 모델로서, 프로세스 목적 및 수행결과인 성과(Outcomes), 그리고 프로세스 간의 연관관계를 설명하는 아키텍처를 포함하고 있다.

프로세스 참조모델에서 규정한 프로세스 목적과 성과를 달성하기 위해 무엇(What)을 해야 하는가에 대한 베스트 프랙티스(best practices)를 규정한 것을 심사 모델이라 부른다. 심사모델에서 정의한 베스트 프랙티스는 수행하면 좋은 결과(성과)가 있는 활동을 적어 놓은 것이다. 예를 들면 15504-5는 소프트웨어 개발과 유지보수에서 수행 되면 좋은 결과(예, 높은 품질의 소프트웨어 생산, 일정 준수, 비용 감소 등)를 가져 올 수 있다고 이미 산업 현장에서 사용하여 그 성과에 대한 객관적으로 증거가 있는 프랙티스(활동)들을 모아 놓은 것이다<sup>2)</sup>. 프로세스 심사란 심사 모델에 정의한 베스트 프랙티스가 수행되어 프로세스 목적을 달성했는지 판단하는 것이다.

요약하면, 국제표준이고 국내 규격인 15504-1, -2, -3 및 -4는 모든 프로세스 모델에서 사용되는 용어, 프로세스 심사 및 개선에 사용되는 문서이다. 이 4개의 표준들은 소프트웨어 개발(15504-5), 시스템 개발(15504-6), 기업의 성숙도(15504-7), IT 서비스(15504-8)에 공통으로 적용된다. 추가적인 프로세스 모델은 8번 이후의 숫자가 부여된다. 현재 (2007년 11월) IT 서비스 관리 심사모델을 위한 참고 모델이 아직 개발되지 않았기 때문에 ISO에서는 IT 서비스 관리 참조모델과 심사모델을 동시에 개발하고 있다. 참고로 그림 1에서의 심사모델이 15504-2와 연결된 것은 심사모델이 15504-2에서 규정한 요구사항에 맞게 만들어 졌다는 것을 의미한다. 본 가이드라인의 내용은 소프트웨어 프로세스에 관한 것이기 때문에 15504-1부터 15504-5까지의 표준만으로 이에 한정한다.

### 1.3 소프트웨어 프로세스 개선의 성과

아무리 잘 만들어진 프로세스 표준이라고 해도, 기업에서 사용하여 "효과"를 얻었다는 객관적인 증거가 없으면 사용을 권장하기 어렵고, 더욱이 기업에게 강요해서는 안된다. 따라서 프로세스 표준의 주요한 과제는 표준에 따라 일했을 때 어떤 성과가 있었는지에 대한 객관적인 증거 확보이다. "안 하는 것보다 좋다"고 하는 주장은 효과나 부작용에 대한 검사 없이 신약을 환자에게 투약하는 바와 다르지 않다. 또한 이러한 객관적 증거는 표준의 제정 목표와 일치하여야 한다. 표준 제정 목표에 어긋나는 효과는 성과가 아닌 부작용에 가깝다.

프로세스에 따른 활동과 개선은 기업의 목표에 기여해야 한다. 기업의 목표에 기여하지 못하는 개선이나 성과는 "프로세스 유희"라고 까지 부른다. 기업은 기업의

---

2) 일부 프로세스관련 가이드라인이나 보고서에 프로세스를 "과정"으로 번역하여 사용하는 경우가 종종 있으나, 이는 틀린 번역이다. 프로세스 모델은 따라하면 진짜 좋다는 best practices를 모아 놓은 것이다. 프로세스마다 고유한 best practices를 SPICE에서는 기본 프랙티스(base practice), CMMI에서는 고유 프랙티스(specific practice)라고 부른다. 프로세스를 "과정"으로 해석하는 경우는 법이나 규정에 의무적으로 거쳐야 하는 절차나 단계를 의미한다. 이는 프로세스 표준과 무관하다.

목표에 기여하는 정도가 큰 프로세스부터 개선을 실시해야 할 것이다.

경영학, ISO 9000, 15504, CMMI-DEV에서 언급하는 기업의 목표의 예로는 고객 만족 증대와 경쟁력 향상을 들 수 있다. 기업의 소프트웨어 프로세스 구현과 프로세스 개선은 이러한 기업의 목표 달성에 기여할 수 있어야 한다. 즉, 경영 목표 달성을 위해 프로세스의 성과 측도(지표, 또는 기준)가 설정되어야만 의미가 있다. 대표적인 프로세스의 성과 측도의 예는 다음과 같으며, 이 목표는 15504 표준 및 CMMI-DEV의 목표이기도 하다:

- 예산 준수 능력 향상
- 일정 준수 능력 향상
- 고객 만족 능력 향상
- 품질 (결함) 또는 요구사항 만족 능력
- 생산성 향상
- 직원의 사기 또는 업무 만족도 향상
- ROI (Return On Investment) 제고

프로세스에서의 품질은 소프트웨어 크기를 결함의 개수로 나눈 측도로 프로세스 개선의 성과를 측정하는 중요한 개념으로 사용되고 있다. 조사연구(survey study)에서는 요구사항 만족 능력이 결함의 대체 측도(surrogate measure)로 사용되기도 한다. 프로세스 개선은 결함을 줄임으로써 재작업을 줄이고, 이에 따라 일정 및 예산 준수를 달성할 수 있도록 하는 것이다. 생산성은 최고 경영층에게는 매력적인 단어이나 프로세스 관점에서는 부수적인 측도임으로 프로세스 개선의 목표로 생산성 제고를 최고의 목표로 내세우지 않는다는 것을 유념하여야 한다.

표준화 추진기관은 자신이 만든 표준이 기업 현장에서 사용하여 유용하고 가치가 있다는 객관적인 증거를 확보한 후이나 표준 사용을 권고할 수 있다. 표준을 사용할 가치가 있다는 객관적인 증거 없이 표준 사용을 강요하는 경우 법적인 문제를 야기 할 수 있다. 따라서 국제적인 표준화 기관에서는 표준 사용이 성과 달성에 기여하는가에 대한 연구에 많은 노력을 기울이고 있다. SPICE 표준 사용에 따른 성과 파악은 세 차례에 걸쳐 전 세계적으로 이루어진 SPICE Trials를 통해 표준 제정 활동의 일환으로 추진되었으며, 그 중 일부를 인용하면 표 1과 같다. 이러한 연구를 예측 타당성(predictive validity)라고 부른다.

표 1: ISO/IEC 15504 표준 사용에 따른 성과의 예

OU 크기	성과 측도	ISO/IEC 15504-5 프로세스
중소기업	예산 준수 능력	
	일정 준수 능력	소프트웨어 설계(ENG.5)
	고객 만족 달성 능력	
	요구사항 만족 능력	
	생산성	
	지원의 사기/업무 만족	
대기업	예산 준수 능력	소프트웨어 설계(ENG.5) 소프트웨어 구축(ENG.6)
	일정 준수 능력	소프트웨어 설계(ENG.5)
	고객 만족 달성 능력	소프트웨어 설계(ENG.5)
	요구사항 만족 능력	소프트웨어 설계(ENG.5)
	생산성	소프트웨어 요구사항 분석 (ENG.2)
		소프트웨어 통합(ENG.7)
		소프트웨어 시험(ENG.8)
	프로젝트 관리(MAN.3)	
지원의 사기/업무 만족	소프트웨어 설계(ENG.5)	

참고: 본 표는 ISO/IEC 15504의 PDTR version(1996)에 대해 실시한 Phase 2 Trials의 결과임

표 1의 해석에 주의해야 할 점은 중소기업의 경우 SPICE에 따른 표준 프로세스 구현과 개선이 성과측도와 관계가 없다는 것이 아니라, 최종 결론을 내릴 정도로 아직 충분한 자료가 확보 되지 않았음을 의미한다. 이유는 초창기 SPICE Trials에는 주로 대기업이 참여하였기 때문에 중소기업에 대한 데이터가 충분하지 않았다. 따라서 나머지 프로세스들에 대해 예측 타당성에 대한 연구가 추후에도 필요하다.

프로세스 개선에 있어서 한 기업의 성공 사례가 항상 다른 기업으로 전파될 수 있는 것은 아니다. 프로세스 개선을 시도한 기업 중에 실패한 기업 보다는 의미 있는 결과를 얻은 기업만이 그 결과를 발표하는 경향이 있다. 실패한 기업의 사례는 잘 알려지지 않는다. 이런 까닭에 어떤 특정 기업의 성공 사례<sup>3)</sup>는 산업 전반으로 일반화 할 수 없다. 속담에 "제비 한 마리 날아왔다고 봄이 온 것이 아니다."가 여기에 해당된다.

3) 일부 소프트웨어 공학 관련 논문을 보면, 사례연구 결과에서 자신이 제시한 측도나 프로세스 개선이 성과가 있음이 "증명"(proof)되었다고 주장하고 있다. 프로세스나 제품에 관련된 데이터를 수집하여 분석하는 것은 실증연구(empirical studies)로 증명이라는 단어는 존재하지 않는다. 증명은 수학적인 방법으로 하는 것이고, 실증연구는 이론(theory)에 대한 가설검정을 하는 것이다.

따라서 표준화를 추진하는 기관에서는 일반화(generalization)<sup>4)</sup>가 가장 주요한 연구 과제이다. 예를 들면 15504에서는 이를 위한 SPICE International Research Consortium이 이를 추진하고 있고, SEI의 경우에는 CMMI-DEV 표준팀 보다는 프로세스 측정과 분석을 수행하는 실증연구팀의 규모가 더 크다.

ISO의 정보기술 관련 다른 표준화 작업과 달리 ISO/IEC 15504 국제 표준화 작업은 표준의 유용성 및 효과에 대한 실증연구와 병행되었다. ISO/IEC 15504 표준화 작업은 1991년 6월 ISO/IEC JTC1/SC7 산하에 소프트웨어 프로세스 표준화의 필요성을 연구하기 위한 연구 그룹 설립을 허가(Resolution 144)받고, 그 연구 결과 1992년 발표 되어 1993년 6월 정식으로 WG10(Working Group 10)이 설립되었다 (ISO/IEC WG10/N017R). WG10에서 표준 제정이 이루어지면서, 표준의 실증 연구를 위한 SPICE Trials이 3차에 걸쳐 실시하여 지속적으로 표준을 보완했다<sup>5)</sup>.

---

4) 소프트웨어 공학에서 Generalization 문제는 다음 논문 참조: M. Jørgensen and D.I.K. Sjøberg, "Generalization and Theory Building in Software Engineering Research," Empirical Assessment in Software Eng. Proc., pp. 29-36, 2004.

5) SPICE Trials의 결과 보고서는 정호원 교수에게 연락하면 무료로 받을 수 있다. Phase 2 Trials 최종 요약 발표는 다음 논문 참조: Ho-Won Jung, Robin Hunter, Dennis Goldenson, Khaled El Emam, "Findings from Phase 2 of the SPICE Trials", *Software Process Improvement and Practice*, Vol. 6, Issue 5, 205-242, 2001.

## 제 2 장 ISO/IEC 15504 - 5 (예시 심사모델)

### 2.1 예시 심사모델이란?

ISO/IEC 12207(소프트웨어 생명주기 프로세스)은 소프트웨어 생명주기 프로세스에 대한 공통의 기본 틀을 규정하고 있다. ISO/IEC 12207은 기본 틀만을 제공하고 있기 때문에 프로세스를 심사하거나 개선하는 목적으로 사용하기는 힘들다. 따라서, ISO는 12207을 프로세스 심사나 개선에 사용 할 수 있는 새로운 구조인 수정조항(Amendment) 1과 2를 개발하였다. 12207 수정조항에서는 소프트웨어 프로세스의 목표(Purpose)와 프로세스 수행결과인 성과(Outcomes)를 규정하고, 이를 소프트웨어 프로세스 참고모델(PRM: Process Reference Model)이라고 부른다.

ISO/IEC 15504-2에서는 소프트웨어 프로세스 참고모델에서 규정한 프로세스 목적과 수행결과를 달성하기 위해서는 무슨(What) 활동을 해야 하는지에 대해 정의한 소프트웨어 심사모델(PAM: Process Assessment Model)을 만들어 사용할 수 있도록 하고 있다. 그러나 모든 기업이 각자의 심사모델을 만들기 어렵기 때문에 소프트웨어 프로세스에서 사용할 수 있도록 모범 답안으로 만들어 기업에서 활용할 수 있도록 한 것이 예시 심사모델인 ISO/IEC 15504-5이다. 그러나 심사모델은 무엇(What)만을 정의하고 있으며, 기업에서 프로세스를 어떻게(How) 구현하면 되는가에 대해서는 설명하지 않고 있다. 기업에서는 심사 모델에서 제시하고 있는 내용을 구체화하여 적용하기 위해서는 프로세스를 실제로 따라해 볼 수 있는 구체적인 내용이 필요하다. 따라서 본 가이드라인에서 제공하는 것이 예시 심사모델인 ISO/IEC 15504-5의 What를 구현하는 방법(How)인 표준 프로세스(프로세스 기술서)를 제공하는 것이다.

앞서 언급하였듯이 ISO/IEC 15504-5 부터 ISO/IEC 15504-8까지는 모두 특정 분야에 적합한 예시 심사모델이며, 이들은 그림 1에서와 같이 각각의 참고모델을 기반으로 하고 있다. 이 심사모델들은 모두 ISO/IEC 15504-1 ~ ISO/IEC 15504-4를 적용하여 작성된 표준이다.

### 2.2 ISO/IEC 15504-5의 구조

예시 심사모델인 ISO/IEC 15504-5는 2차원 구조로 그림 2와 같이 "프로세스 차원"과 "프로세스 능력 차원"으로 구성되어 있다. ISO/IEC 15504-5에서는 프로세스별로 능력수준을 결정한다. 따라서 어떤 프로세스는 능력수준 3이고, 다른 어떤 프로세스는 능력수준 4가 될 수 있다. 이러한 방식으로 프로세스별 능력수준을 결정

하는 것을 연속모델(continuous model)이라고 부른다. ISO/IEC 15504-7(조직 성숙도 심사)은 사전에 정의된 몇 개의 프로세스 그룹이 어떤 능력수준에 이르면 성숙 수준 (maturity level) 이라 부르며, 이러한 성숙수준 결정 모델을 단계모델(staged model)이라 부른다. 참고로 CMMI-DEV는 모델이라는 용어 대신에 연속 표현(staged representation)과 단계 표현(staged representation)이라고 부른다.

## 능력 차원

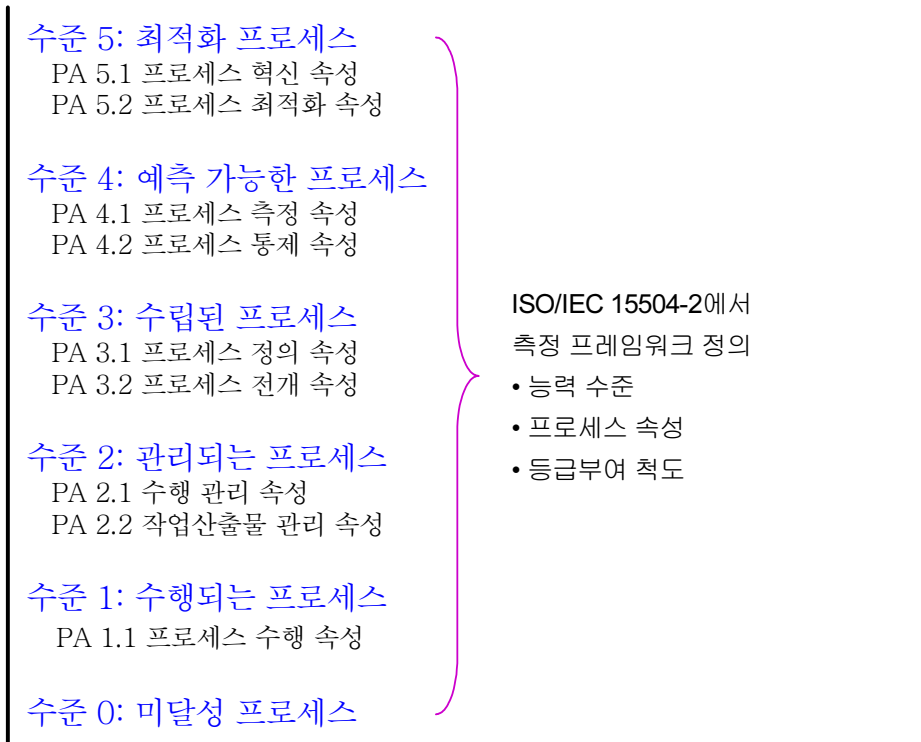


그림 2: 프로세스 심사모델의 2차원 구조

그림 2에서와 같이 15504-5에서는 프로세스 차원을 3개의 프로세스 생명주기 범주인, 기본 (생명주기) 프로세스, 조직 (생명주기) 프로세스, 지원 (생명주기) 프로세스 범주로 분류하고 있다. 이는 다시 총 9개의 그룹으로 분류하고, 9개의 그룹에는 48개의 프로세스를 정의하고 있다. 각 프로세스는 그림 2에서와 같이 0에서 5까지의 능력수준으로 정의된다.

## 프로세스 차원

그림 2에서 프로세스 차원에 속하는 48개의 프로세스는 그림 3과 같다. 이 48개 프로세스는 참조모델인 ISO/IEC 12207의 수정조항(Amendment) 1과 2에서 정의된 프로세스 중에서 가져온 것이다.



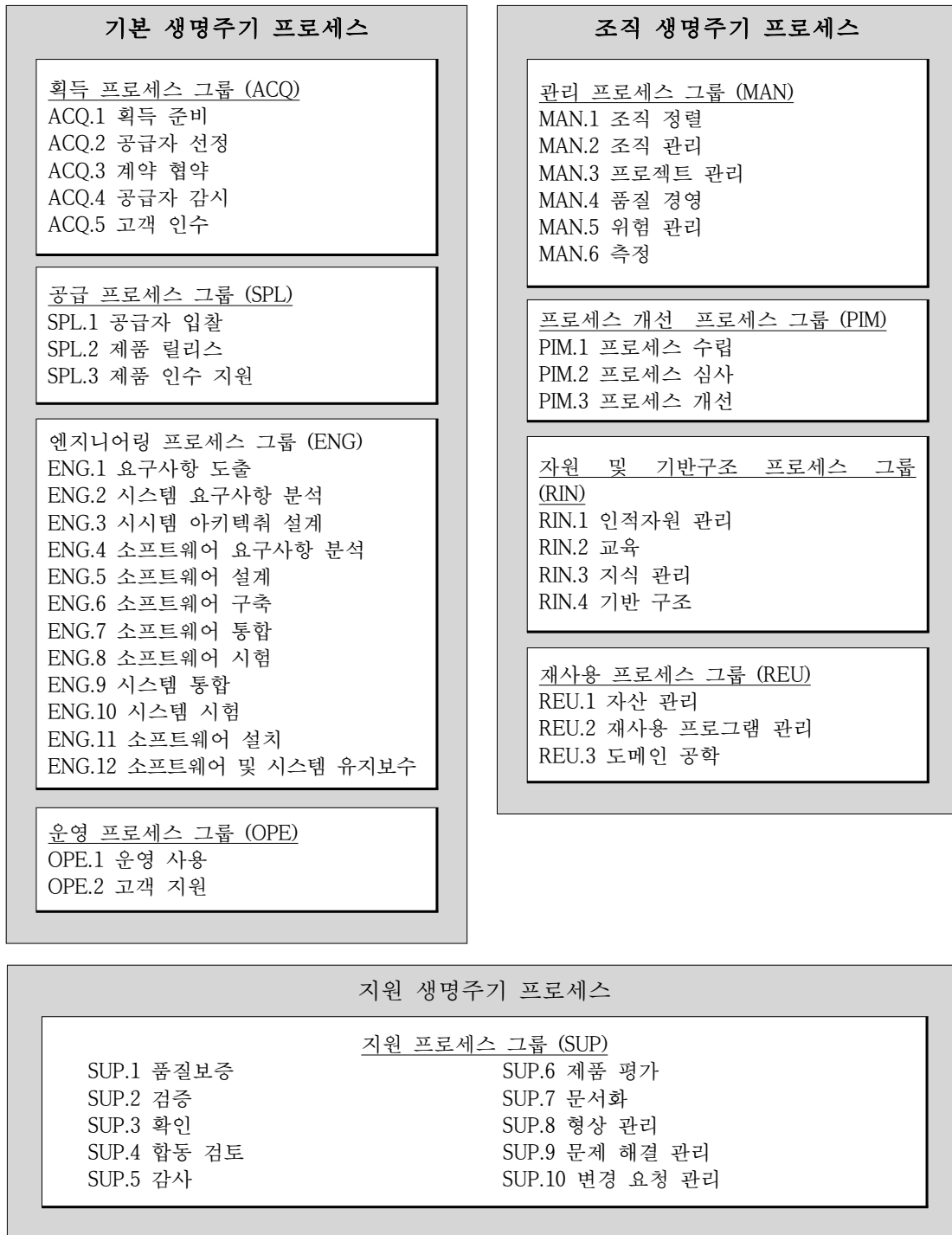


그림 3: 15504 - 5 프로세스 범주 및 그룹

ISO/IEC 15504-5에서 프로세스 구성을 MAN.5 (위험 관리)의 예를 들어 설명하면 그림 4와 같다. MAN.5에는 6개의 프로세스 수행 결과(Outcomes)와 7개의 기본 프랙티스(BP: Base Practice)가 있으나 지면 관계상 각 3개씩만 보여준다.

<b>프로세스 ID</b>	MAN.5	
<b>프로세스 이름</b>	위험 관리	
<b>프로세스 목적</b>	위험 관리 프로세스의 목적은 지속적으로 위험을 식별, 분석, 처리 및 감시하기 위한 것이다.	12207 Amd 2에서 정의
<b>프로세스 수행결과 (Outcomes)</b>	위험 관리 프로세스를 성공적으로 구현한 결과로서: 1) 수행되어야 할 위험 관리의 범위가 결정된다; 2) 적절한 위험 관리 전략이 정의되고 구현된다; 3) 프로젝트 수행 과정에서 발생하는 위험이 식별된다;	
<b>기본 프랙티스</b>	<b>MAN.5.BP1: 위험 관리 범위 수립.</b> 조직의 위험 관리 정책에 따라 수행되어야 할 위험 관리 범위를 결정한다. [수행결과: 1] <b>MAN.5.BP2: 위험 관리 전략 정의.</b> 프로젝트 및 조직 수준에서 각 위험 또는 일련의 위험을 식별하고, 분석하고, 처리하고, 감시하기 위한 적절한 전략을 정의한다. [수행결과: 2, 5] <b>MAN.5.BP3: 위험 식별.</b> 초기에는 프로젝트 전략 내에서 위험을 식별하고, 프로젝트 수행 중 발생하는 위험도 식별한다. [수행결과: 3]	프로세스 목적과 수행결과를 달성하기 위해 '무슨 일(what)'을 해야 하는지 정의
<b>작업산출물</b>		
<b>입력물</b>	<b>출력물</b>	
	07-07 위험 측도 [수행결과: 5]	
08-12 프로젝트 계획서 [수행결과: 1]		
	08-14 복구 계획서 [수행결과: 4, 6]	
08-19 위험관리 계획서 [수행결과: 4, 5]	08-19 위험관리 계획서 [수행결과: All]	기본 프랙티스를 수행하기 위해 필요하다고 여겨지는 입력물과 출력물
08-20 위험 완화 계획서 [수행결과: 6]	08-20 위험 완화 계획서 [수행결과: 3, 4]	
13-20 위험 조치 요청(서) [수행결과: 4]	13-20 위험 조치 요청(서) [수행결과: 2, 6]	
	14-02 시정 조치 등록 [수행결과: 6]	
14-09 추적 시스템 [수행결과: 3, 4, 5, 6]	14-09 추적 시스템 [수행결과: 2, 3, 4, 5, 6]	
	15-08 위험 분석 보고서 [수행결과 4]	
	15-09 위험 상태 보고서 [수행결과 4, 5]	

그림 4: 15504 - 5에서 프로세스 정의

ISO/IEC 15504-5의 모든 프로세스는 그림 4와 같이 프로세스 이름과 프로세스 수행결과인 성과는 참고모델인 ISO/IEC 12207 Amd 1 및 2에서 규정한 것을 가져 오고, 이러한 목적과 결과를 달성하기 위해 필요한 활동인 기본 프랙티스와 작업산출물을 15504-5에서 정의하고 있다. 또한 ISO/IEC 15504-5에서는 작업산출물에 포함되어야 할 내용을 부속서 A로 제공하고 있다.

기본 프랙티스를 수행하는 방법은 기업의 상황에 따라 다르기 때문에 각 기본 프랙티스를 수행하는 방법을 문서화하는 것을 프로세스 정의라고 부르며, 프로세스 정의의 결과를 표준 프로세스라고 부른다. 즉, 프로세스 심사를 위해서는 ISO/IEC 15504-5만으로 충분하다고 할 수 있으나, 산업 현장에서는 그대로 사용할 수 없다. 즉 기업에서는 기본 프랙티스에 맞게 사내 표준 프로세스인 프로세스 기술서를 만들어 사용한다.

### 프로세스 능력 차원

능력 차원에서 정의된 프로세스 능력수준(CL: Capability Level) 및 프로세스 속성들은 ISO/IEC 15504-2에서 정의된 것들과 동일하다. ISO/IEC 15504-5에서는 각 프로세스 별로 능력 수준을 결정하도록 하고 있다. 각 프로세스 능력은 가장 낮은 불완전(Incomplete)인 0 수준부터 가장 높은 최적화(Optimizing)인 5 까지 여섯 단계로 능력수준을 부여한다. 수준이 높아질수록 프로세스 능력이 높음을 의미한다.

프로세스 능력을 측정하기 위해 ISO/IEC 15504-2에서는 9개의 프로세스 속성(PA: Process Attributes)을 사용하도록 규정하고 있다. 이 프로세스 속성은 그림 3에 있는 모든 프로세스에 적용 가능하다. 각 능력수준과 이를 측정하기 위해서 분화한 프로세스 속성은 표 2와 같다.

표 2: 프로세스 능력수준 및 프로세스 속성 정의

능력수준	프로세스 속성
<b>수준 0 (CL 0: 미달성 프로세스)</b> 프로세스가 구현되지 않았거나 목적을 달성하지 못한다.	없음
<b>수준 1 (CL 1: 수행되는 프로세스)</b> 구현된 프로세스가 프로세스 목적을 달성한다	<b>PA 1.1 (프로세스 수행 속성)</b> 프로세스 목적 달성 정도를 측정하는 척도이며, 이 속성의 완전한 달성은 수행결과인 성과(Outcomes)를 달성한다.
<b>수준 2 (CL 2: 관리되는 프로세스)</b> 수준 1의 “수행되는 프로세스”가 관리(계획, 감시, 조정)되고, 작업산출물이 알맞게 구축, 통제, 유지관리된다.	<b>PA 2.1 (수행 관리 속성)</b> 프로세스 수행이 관리되는 정도를 측정하는 척도이다
	<b>PA 2.2 (작업산출물 관리 속성)</b> 프로세스에서 만들어진 작업산출물이 잘 관리되는 정도를 측정하는 척도이다.
<b>수준 3 (CL 3: 수립된 프로세스)</b> 수준 2의 “관리되는 프로세스”가 정의된 프로세스를 사용해 구현되어 프로세스 수행결과인 성과를 달성한다.	<b>PA 3.1 (프로세스 정의 속성)</b> ‘표준 프로세스’가 ‘정의된 프로세스’ 구현에서 사용되기 위해 유지관리 되는 정도를 측정하는 척도이다.
	<b>PA 3.2 (프로세스 전개 속성)</b> 표준 프로세스가 정의된 프로세스로서 효과적으로 사용된 정도를 측정하는 척도이다.
<b>수준 4 (CL 4: 예측 가능한 프로세스)</b> 수준 3의 “수립된 프로세스”가 정의된 한계(defined limits) 안에서 수행되어 프로세스 성과를 달성한다.	<b>PA 4.1 (프로세스 측정 속성)</b> 프로세스 수행이 사업 목표 달성에 기여한다는 것을 보장하기 위하여 측정 결과가 사용되는 정도를 측정하는 척도이다.
	<b>PA 4.2 (프로세스 통제 속성)</b> 정의된 한계 내에서 안정적이고, 요구사항을 만족하고, 예측 가능한 프로세스를 만들기 위해 프로세스가 정량적으로 관리되는 정도를 측정하는 척도이다.
<b>수준 5 (CL 5: 최적화 프로세스)</b> 수준 4의 “예측 가능한 프로세스”는 관련된 현재 또는 계획된 사업 목표를 위해서 지속적으로 개선된다.	<b>PA 5.1 (프로세스 혁신 속성)</b> 프로세스 변경이 식별되는 정도를 측정하는 척도이다. 프로세스 변경은 우연 원인(common causes) 분석 및 프로세스의 정의와 구현을 위한 혁신적인 접근방법의 조사 통해 이루어진다. 이를 통해 개선의 기회를 포착하고 프로세스 개선 목표를 달성할 수 있는 전략을 수립한다.
	<b>PA 5.2 (프로세스 최적화 속성)</b> 프로세스의 정의, 관리, 수행의 변경이 관련 프로세스 개선 목적 달성에 미치는 영향의 정도를 측정하는 척도이다.

### 2.3 프로세스 능력 결정 방법

프로세스의 능력수준은 표 2에 있는 프로세스 속성(PA)의 달성 정도를 표 3과 같이 4가지 순위 척도인 NPLF를 사용해 판정한다. 이를 등급부여(rating)라고 부른다.

표 3: 프로세스 속성의 등급판정

판정 값	설 명
N (Not achieved) 미달성	<b>0 &lt; 프로세스 속성의 달성 정도 ≤ 15 %</b> 프로세스 속성이 달성되었다는 증거가 거의 없다.
P (Partially achieved) 부분 달성	<b>15% &lt; 프로세스 속성의 달성 정도 ≤ 50 %</b> 프로세스 속성 달성을 시도하였고, 부분적으로 달성했다는 증거가 있다. 달성된 속성의 일부조차도 향후 달성이 가능할지 예측할 수 없다.
L (Largely achieved) 대부분 달성	<b>50% &lt; 프로세스 속성의 달성 정도 ≤ 85 %</b> 프로세스 속성에 대한 체계적인 접근과 뚜렷이 달성되었다는 증거가 있다. 그러나 일부 약점이 존재할 수 있다.
F (Fully achieved) 완전 달성	<b>프로세스 속성의 달성 정도 &gt; 85 %</b> 프로세스 속성을 완전히 달성하고 완전하고도 체계적인 접근을 했다는 증거가 있다. 관련된 심각한 약점이 존재하지 않는다.

프로세스의 능력수준 CL은 표 3 에서 정의된 프로세스 속성의 판정값에 따라 표 4와 같이 결정된다. 프로세스 속성 등급판정을 프로세스의 프로파일(profile)이라 부른다. 심사 결과에는 심사된 모든 심사 대상 프로세스의 프로세스 프로파일이 포함된다. 프로세스 프로파일을 이용하여 프로세스 능력수준을 결정하는 방법은 표 4와 같다. 어떤 프로세스가 능력수준 k가 되려면 능력수준 k-1까지는 F(fully achieved)로 PA k-1.1과 PA k.2는 F(fully achieved) 또는 L(largely achieved)로 등급판정 받아야 한다.

표 4: 능력수준 판정 방법

PA 5.2 (프로세스 최적화 속성)						F 또는 L
PA 5.1 (프로세스 혁신 속성)						F 또는 L
PA 4.2 (프로세스 통제 속성)					F 또는 L	F
PA 4.1 (프로세스 측정 속성)					F 또는 L	F
PA 3.2 (프로세스 전개 속성)				F 또는 L	F	F
PA 3.1 (프로세스 정의 속성)				F 또는 L	F	F
PA 2.2 (작업산출물 관리 속성)			F 또는 L	F	F	F
PA 2.1 (수행 관리 속성)			F 또는 L	F	F	F
PA 1.1 (프로세스 수행 속성)	P or N	F 또는 L	F	F	F	F
	수준 0	수준 1	수준 2	수준 3	수준 4	수준 5

소프트웨어 프로세스 심사의 주요 목적은 현재의 프로세스 능력수준과 장단점을 파악하여 개선의 기회를 찾는 활동이다. 또한 고객이나 대외적인 요구에 의해 프로세스 능력수준인 CL이 요구되기도 하나, 이 역시 해당 기업의 요구사항을 수행할 능력이 있는가를 파악하기 위한 수단이다. 따라서 프로세스 속성의 등급결정과 함께 프로세스의 장단점을 파악하여 개선의 기회를 찾는 것이 중요하다.

### 제 3 장 프로세스 속성의 등급 결정에 사용되는 지표

프로세스 능력은 키나 몸무게와 같이 직접 측정할 수 없고, 오직 간접적인 방법으로만 측정이 가능한 추상적인 개념(abstract construct)이다. 따라서 이를 좀 더 용이하게 측정하기 위해 세분화 한 것이 PA라고 부르는 프로세스 속성이다. 그러나 프로세스 속성 역시 추상적인 개념이라 간접적인 방법으로만 측정가능하며, 간접적인 측정 도구를 프로세스 심사지표(assessment indicator)라고 부른다. 이 지표를 이용해 프로세스 속성의 달성 정도를 NPLF로 등급(rating) 수준을 부여한다. 프로세스 심사지표는 PA 1.1의 달성정도에 등급을 부여하는데 사용되는 프로세스 수행지표(process performance indicator)와 PA 2.1 ~ PA 5.2의 달성정도에 등급을 부여하는데 사용되는 프로세스 능력 지표(process capability indicator)로 나뉜다(그림 5 참조). 각 지표를 구성하는 지표는 아래와 같다.

- a) 프로세스 수행지표: 기본 프랙티스 및 작업산출물로 구성
- b) 프로세스 능력지표: 공통 프랙티스, 공통 자원, 공통 작업산출물로 구성

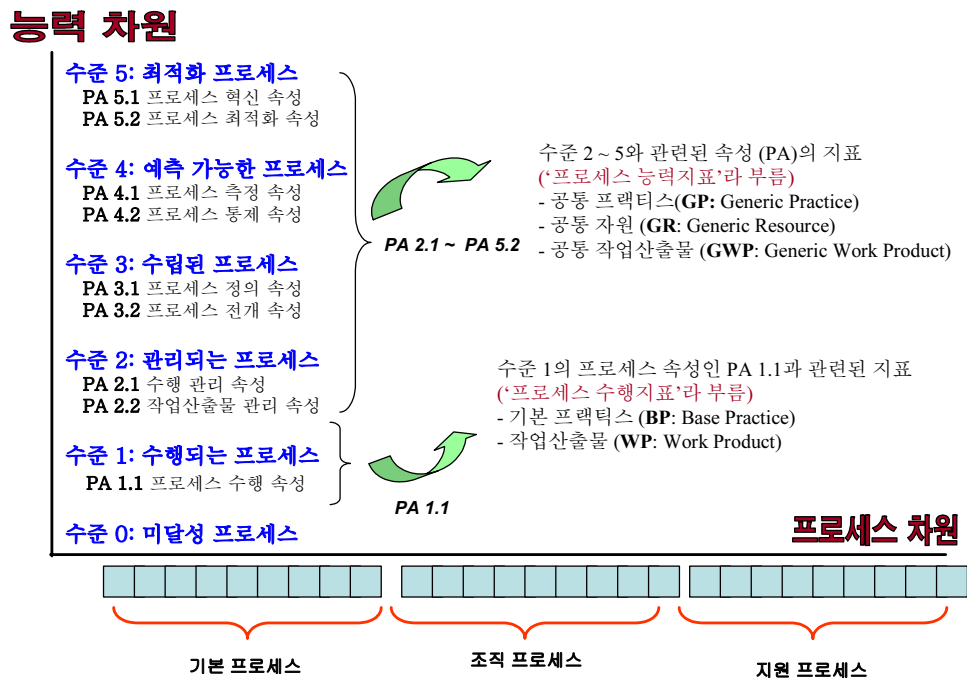


그림 5: 능력수준을 결정하는 지표

### 3.1 PA 1.1 등급부여에 사용되는 '프로세스 수행지표'

ISO/IEC 15504-5에서는 능력수준 1 달성을 결정하는데 사용되는 PA 1.1의 프로세스 수행 지표로는 (1) 기본 프랙티스(BP: Base Practice)와 (2) 작업산출물(WP: Work Product)이 있다(그림 5 참조). 이 수행지표들의 목표 달성 정도에 따라 PA 1.1에 등급(NPLF)을 부여한다. 세부적인 내용은 아래와 같다. 참고로 작업산출물은 그림 3의 각 프로세스 마다 서로 다르기 때문에 '고유 작업산출물(specific workproduct)'이라고 부르기도 한다.

#### (1) 기본 프랙티스(BP: Base Practice)

기본 프랙티스(BP)는 프로세스 수행 속성인 PA 1.1의 NPLF 등급을 부여할 때 사용되는 지표 중에 하나다. 프로세스 마다 고유하며, 프로세스 목적 달성을 위해 무엇(What)을 수행해야 하는지를 규정하고 있다. 기본 프랙티스를 일관성 있게 수행하는 것은 프로세스 목적을 일관성 있게 달성할 수 있다는 것을 의미한다.

기업마다 프로세스 상황(응용 도메인, 사업 목적, 개발 방법론, 조직 규모 등)이 다르므로 기본 프랙티스는 추상적인 수준으로 기술되어 있다. 따라서 각 기업은 기본 프랙티스를 '어떻게' 수행해야 하는지를 기술한 문서가 필요하며, 이 문서를 프로세스 기술서 (표준 프로세스)라고 부르며, 본 가이드라인의 목적은 프로세스 기술서의 예를 제공하는 것이다.

#### (2) 작업산출물(WP: Work Product)

작업산출물(WP)<sup>6)</sup>은 프로세스 수행속성인 PA 1.1의 NPLF 등급을 부여할 때 사용되는 지표 중에 하나이다. 기본 프랙티스 수행을 위해 필요하거나(입력물), 수행 결과로 나오는 결과(출력물)이다. 또한 기본 프랙티스의 수행 증거인 작업산출물과 그 특성은 프로세스 목적과 그에 따른 결과인 성과(Outcomes) 달성 정도를 판단하는 시작점이 되는 자료이다.

ISO/IEC 15504-5의 48개 각각의 프로세스에서는 프로세스를 수행하기 위해 필요한 작업산출물과 수행에 따른 작업 산출물의 예를 제시하고, 이러한 작업 산출물에 포함되어야 하는 내용을 ISO/IEC 15504-5에서 형태로 제시되고 있다. 이것은 SPICE 심사모델들 (15504-5 ~ 8)이 가지는 가장 큰 장점 중에 하나이다. 본 가이드라인의 범위 내에 있는 작업산출물의 특성은 본 가이드라인 부속서 A에서 제시한다.

6) 작업산출물은 15504-5 표준의 부속서 B. 2에서 제공되고 있다.

### 3.2 PA 2.1 ~ PA 5.2 등급부여에 사용되는 '프로세스 능력지표'

프로세스 능력수준 2~5를 결정하는 프로세스 속성인 PA 2.1 ~ PA 5.2는 능력지표인 (1) 공통 프랙티스 (GP: Generic Practice), (2) 공통 자원 (GR: Generic Resource), (3) 공통 작업 산출물 (GWP: Generic Work Product)의 목표 달성 정도에 따라 NPLF로 등급이 부여된다. 여기서 공통이란 '모든 프로세스 적용될 수 있다'라는 의미이며, 그 내용은 아래와 같다.

#### (1) 공통 프랙티스 (GP: Generic Practice)

PA 2.1 ~ PA 5.2의 NPLF 등급을 부여하기 위해 사용되는 지표 중에 하나이다. 공통 프랙티스는 모든 프로세스에 적용되는 공통 유형의 활동이다. 공통 프랙티스는 표 2에 있는 프로세스 속성(PA)의 달성을 지원하는 활동으로, 주로 경영 활동과 관련된 내용이다.

프로세스 능력수준을 결정할 때 공통 프랙티스의 수행에 초점이 맞춰진다. 일반적으로 공통 프랙티스의 성공적인 수행은 해당 프로세스 속성(PA)이 달성될 것이라고 기대된다.

#### (2) 공통 자원 (GR: Generic Resource)

공통 자원 지표는 프로세스 속성(PA)을 달성하기 위해 프로세스를 수행할 때 필요로 하는 인적 자원, 도구, 방법 및 기반구조를 말한다. 자원의 가용성은 프로세스 속성의 목적을 달성할 수 있는 가능성을 나타낸다.

#### (3) 공통 작업산출물(GWP: Generic Work)

프로세스 수행에 따른 출력물인 작업산출물에 나타나리라 기대되는 특성들이다. 공통 작업산출물은 능력수준 1의 지표 중에 하나인 작업산출물을 분류하는 기반이 된다. 즉, GWP는 모든 유형의 프로세스의 입력물 또는 출력물이 될 수 있는 작업산출물의 기본 유형을 나타낸다. 본 가이드라인의 범위 내에 있는 (PA2.1 ~ PA 3.2) 공통 작업산출물의 특성은 본 가이드라인 부속서 A에서 제시한다.



### 3.3 프로세스 수행 및 능력지표

SPICE 심사에서는 각각의 프로세스에 대해 능력수준을 부여하며, PA 1.1 (프로세스 수행 속성)의 달성 정도를 NPLF로 등급을 부여하고 이를 기반으로 능력수준 1 (CL1)의 달성여부를 판단한다. 능력수준 2~5도 프로세스 속성들의 달성 정도를 NPLF로 등급을 부여하고, 이를 기반으로 표 4에 따라 능력수준을 결정한다. 프로세스 속성인 PA의 달성 정도에 등급을 부여하기 위해 사용되는 것을 PA 1.1에서는 수행지표(기본 프랙티스, 작업산출물), PA 2.1 ~ PA 5.3에서는 능력지표(공통 프랙티스, 공통자원, 공통 작업산출물)라 한다. 이를 요약하면 그림 6과 같으며 본 절에서는 능력수준 1과 관련된 간단한 설명과, 그림 6의 왼쪽 편에서 능력수준 2~3에 해당하는 공통 프랙티스, 공통 자원, 공통작업산출물에 대해 표를 이용하여 설명한다. 제 2 부에서 기본 프랙티스를 수행하는 방법(How)인 프로세스 기술서를 제공한다. 또한 작업산출물 WP의 특성은 부속서 A에서 제공한다.

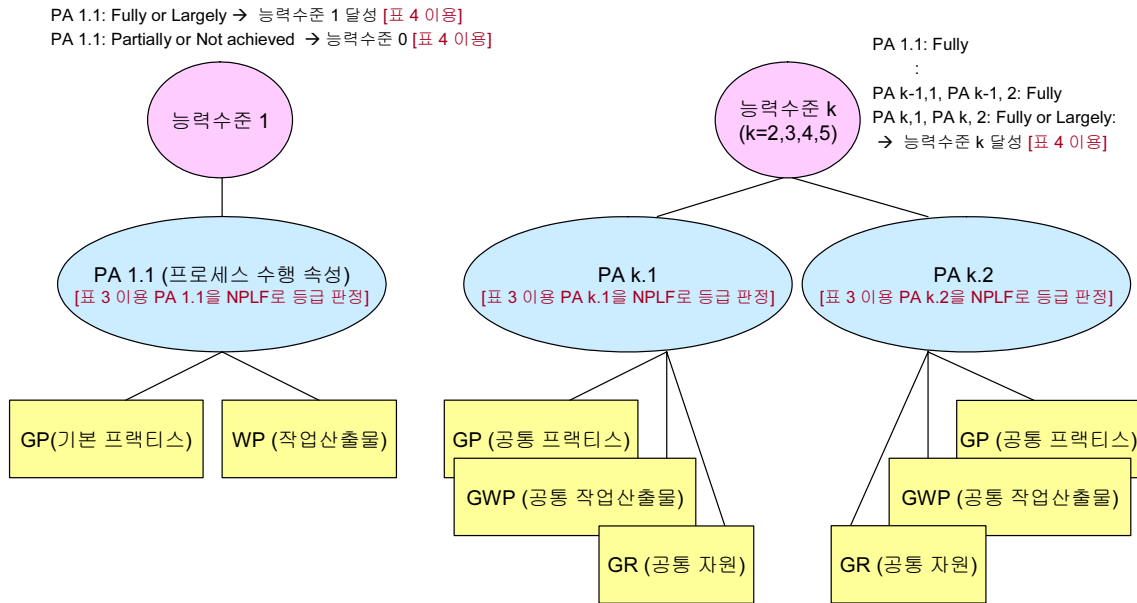


그림 6: 능력수준 결정 단계

그림 6에서 PA 1.1의 프로세스 수행지표인 기본 프랙티스 BP는 그 내용의 분량이 너무 커서 본 가이드라인에서 소개하지 않는다. 또한 본 가이드라인에서는 능력수준 3까지에 해당하는 PA까지(PA 1.1~PA3.2)만 설명한다. 기본 프랙티스와 능력수준 4와 5에 대한 설명은 KS 표준이나 15504 표준을 참조 바란다.

**PA 1.1 (프로세스 수행 속성)**

GP 1.1.1 프로세스 수행결과를 달성

프로세스 수행 속성의 달성은 프로세스 수행 증거로 작업산출물이 존재하고, 작업산출물은 부속서 A 에서 기술한 특성을 가지며, 프로세스 목적을 달성하는 것이다.

**PA 2.1 (프로세스 수행 속성)**

표 5에서 GP 2.1.1 부터 GP 2.1.3까지는 프로세스 수행 목적 (방침)을 정하고 이 목적을 달성하기 위하여 계획을 세우고, 감시를 하는 PDCA(Plan-Do-Check-Act) 싸이클이다. GP 2.1.1은 조직이 가지고 있는 프로세스의 목표(기대)를 조직 구성원에게 표명하는 것이며, 조직이 방향과 원칙을 결정하는 최고 관리자급이 결정한다. 계획서인 08-00에는 PA 2.1의 모든 GP, GR, GWP가 포함되고, 프로세스 기술서가 포함되거나 인용된다.

**표 5: PA 2.1의 공통지표, 공통 자원, 공통 작업산출물**

PA 2.1 (수행 관리 속성) 프로세스 수행이 관리되는 정도를 측정하는 측도이다								
GP	GP vs. GR							GP vs. GWP
GP 2.1.1 (a) <b>프로세스 수행 목적 식별</b> : 목적은 프로세스 요구사항에 근거; 수행범위; 목적 정의에서 가정 및 제약조건 식. [목적의 예: 산출물 품질, 프로세스 수행시간, 자원 사용, 프로세스 경제]								08-00 계획서
GP 2.1.2 (b) <b>식별된 목표를 달성하기 위한 프로세스 수행 계획을 만들고 감시</b> : 수행 계획서 개발 (프로세스 생명주기 정의, 이정표 설정, 추정); 활동과 과업이 정의; 일정; 작업산출물 감도 계획; 감시 활동.								08-00 계획서 15-00 보고서
GP 2.1.3 (c) <b>프로세스 수행을 조정</b> : 문제점 식별; 시정조치; 계획서 수정; 일정 재수립.								08-00 계획서 14-00 등록
GP 2.1.4(d) <b>프로세스를 수행을 위한 책임과 권한을 규정</b> : 책임과 권한 정의 (작업산출물 검증 포함); 프로세스 수행을 위해 필요한 경험, 지식, 스킬 규정								08-00 계획서 13-00 기록
GPI 2.1.5 (e) <b>계획에 따른 프로세스 수행을 위해 가용자원의 파악 및 사용</b> : 인적 물적 자원; 프로세스 수행에 필요한 정보								08-00 계획서 13-00 기록 14-00 등록
GP 2.1.6 (f) <b>관련자간의 인터페이스 관리</b>								08-00 계획서 13-00 기록

목적, 책임과 권한을 가지는 인적 자원  
 시설 및 기반구조 자원  
 프로젝트 계획, 관리와 통제 도구 (시간과 비용 보고 포함)  
 작업흐름 관리 시스템  
 전자 우편 및 (또는) 다른 의사소통 절차와 수단  
 정보 및 (또는) 경험 저장소  
 문제 및 이슈 관리 절차와 수단

프로세스 계획과 관련하여 작업 분해도(work breakdown structure)가 사용될 수 있다. 계획은 감시(monitoring)를 수반하며, GP 2.1.2와 GP2.1.3은 프로젝트 계획과 감시에 따른 시정조치를 말한다. 조직 차원의 표준 프로세스인 MAN.3 (프로젝트 관리)가 구현되지 않으면 프로젝트 관리라고 할 수 있다.

표 5의 GP 2.1.4에서는 책임과 권한을 규정하고(프로세스 기술서 참조), 해당 프로세스를 수행하기 위해 필요한 경험이나, 지식, 스킬이 무엇인지 기록한다. 프로젝트 차원의 교육에 관한 것으로 이것이 조직 차원에서 표준 프로세스가 확립되면 RIN.2 (교육) 프로세스가 된다. GP 2.1.5는 계획에 따라 프로세스를 수행하기 위해 필요한 인적 물적 자원, 그리고 필요한 정보가 파악된다. GP 2.1.6은 프로세스 수행에 관련된 개인 및 그룹이 프로젝트 참여에서 책임과 권한, 정보 공유, 또는 조직 내의 직급 또는 업무상 상하 관계자의 위험이나 문제점 등의 검토가 포함된다. GP 2.1.2에서 만든 위험관리 체계에 따라 GP 2.1.3과 GP 2.1.5에서는 위험에 대한 감시를 실시한다. GR은 GP를 수행하기 위하여 사용될 수 있는 자원의 예이다.

## PA 2.2 (작업산출물 관리 속성)

표 6는 GP 2.2.1에서는 작업산출물의 기능과 비기능 요구사항을 표준으로 규정하고, 작업산출물의 품질기준(만들어질 작업산출물과 관련된 속성)을 정의하고, 검토와 승인 기준이 식별된다. GP 2.2.1에서는 작업산출물 관리를 위해 선정된 정책 및 전략, 작업산출물을 개발, 배포, 유지관리하기 위한 요구사항, 품질 유지하기 위해 필요한 품질관리 활동을 계획서에 규정한다. 또한 산출물간 의존성을 식별하고 승인 요구사항을 정의한다.

표 6: PA 2.2의 공통지표, 공통 자원, 공통 작업산출물

PA 2.2 (작업산출물 관리 속성) 프로세스에 의해 생산된 작업산출물이 적절하게 관리되는 정도를 측정하는 측도이다								
GP	GP vs. GR							GP vs. GWP
GP 2.2.1 (a) <b>작업산출물의 요구사항 정의:</b> 작업산출물 요구사항 정의; 품질 기준 수립; 검토와 승인 기준 정의								18-00 표준 20-00 템플릿 21-00 작업산출물
GP 2.2.2 (b) <b>작업산출물의 문서화 및 통제를 위한 요구사항 정의:</b> (배포, 작업출물 및 그들의 구성요소 식별, 추적성 등); 산출물간 의존성 식별; 승인 요구사항 정의								08-00 계획서 20-00 템플릿 21-00 작업산출물
GP 2.2.3 (c) <b>작업산출물을 식별하고, 문서화하고, 통제 실시</b> (형상관리에 해당)								14-00 등록 16-00 저장소 21-00 작업산출물
GP 2.2.4 (d) <b>요구사항을 만족을 위해 작업산출물을 검토하고 조정</b>								13-00 기록

요구사항 관리 방법도구  
형상관리 시스템  
문서 상세화 및 지원도구  
문서 식별 및 통제 절차  
작업산출물 검토 방법 및 경험  
검토 관리 방법 및 도구  
의사소통 절차와 수단(메일, 인터넷 등)  
문제 및 이슈 관리 절차와 수단

또한 GP 2.2.2는 형상 항목 및 형상 관리 방법을 정의하고, 그 방법에 따라 GP 2.2.3에서는 형상관리를 수행한다. GP 2.2.4에서는 작업산출물에 대한 검증 활동을 수행한다.

### PA 3.1 (프로세스 정의 속성)

표 7은 PA 3.1의 모든 공통 프랙티스(GP)는 정책(policy)을 GWP로 규정하고 있다. 여기서 정책이란 표준 프로세스 유지관리에 대한 조직의 이행의지의 증거이다. 또한 프로세스 기술서가 모든 GP의 GWP이다. (GP 3.1.4는 프로세스 기술서가 해당되지 않는 것으로 나와 있으나, 실제로는 해당된다). 따라서 PA 3.1은 전체가 표준 프로세스를 수립하는데 관련된 GP이다. 표준 프로세스를 수립한다는 것은 회사 차원의 (프로젝트 수준의 프로세스 기술서는 능력수준 3 이전에도 존재) 프로세스 기술서를 만드는 것이다. 표준 프로세스에는 기본적인 프로세스 요소, 다른 프로세스와 상호작용, 적절한 조정 지침이 포함된다. 여기서 만든 표준 프로세스에 따라 프로젝트에 맞게 조정하여 사용하는 것이 '정의된 프로세스(defined process)'이고, 이는 PA 3.2에 해당된다. 또한 프로세스의 효과성과 적절성을 감시하기 위한 방법을 다룬다.

표 7: PA 3.1의 공통지표, 공통 자원, 공통 작업산출물

PA 3.1 (프로세스 정의 속성) '표준 프로세스'가 '정의된 프로세스' 구현을 지원하기 위해 유지관리 되는 정도를 측정하는 측도이다.							
GP	GP vs. GR					GP vs. GWP	
GP 3.1.1 (a) 표준 프로세스의 정의							09-00 정책 10-00 프로세스 기술서 18-00 표준
GP 3.1.2 (b) 프로세스간 순서 및 상호작용 결정							09-00 정책 10-00 프로세스 기술서
GP 3.1.3 (c) 표준 프로세스 수행을 위한 역할 및 능력을 식별							10-00 프로세스 기술서 19-00 전략
GP 3.1.4 (d) 프로세스를 수행하기 위해 필요한 인프라와 작업환경 식별							09-00 정책 16-00 저장소 19-00 전략
GP 3.1.4 (d) 프로세스를 수행하기 위해 필요한 인프라와 작업환경 식별							09-00 정책 16-00 저장소 19-00 전략
GP 3.1.5 (e) 프로세스의 효과 및 적절성을 감시할 수 있는 도구 결정							09-00 정책 10-00 프로세스 기술서

프로세스 모델링 방법/도구  
 훈련 자료 및 교육 과정  
 자원 관리 시스템  
 프로세스 기반구조  
 감리 및 동향 분석 도구  
 프로세스 감시 방법

GP 3.1.4에서 GWP인 전략이란 표준 프로세스의 정의, 유지, 지원을 위한 접근법으로 인프라, 작업환경, 훈련, 내부감사, 경영검토 등이 포함된다. 또한 GP 1.1.5가 수행되었을 때 프로세스 계획과 수행에서 수집된 산출물(artifacts)인 계획서, 데이

데, 데이터 분석 방법, 분석 결과가 저장되는 저장소가 포함된다. GP 3.1.5는 프로세스를 개선하기 위하여 프로젝트 계획이나 정의된 프로세스를 수행할 때 수집되는 데이터와 기록을 식별한다.

프로세스 기술서 [10-00]에 포함되어야 하는 내용은 다음과 같다.

- 프로세스와 절차에 대한 자세한 기술서는 다음의 사항을 포함함:
  - (적용 가능할 경우) 표준 프로세스의 조정
  - 프로세스의 목적
  - 프로세스의 수행결과
  - 수행되어야 할 과업과 활동 및 과업의 순서
  - 과업 활동들 사이의 주요한 의존관계
  - 과업을 수행하기 위해 필요한 예상시간
  - 작업산출물의 입력물/결과물
  - 작업 입력물과 결과물 사이의 연결
- 프로세스 시작과 종료 기준을 식별함
- 프로세스에 대한 내외부 인터페이스를 식별함
- 프로세스 측도를 식별함
- 기대 품질을 식별함
- 기능별 역할과 책임을 식별함
- 책임있는 사람에 의해 승인됨

### PA 3.2 (프로세스 전개 속성)

표 8에서 PA 3.2는 PA 3.1에서 만든 표준 프로세스를 조정하여 정의된 프로세스를 만드는 속성이다. 계획서에는 모든 GP에 해당하는 내용을 포함하는 전략을 표명한다. 프로세스 기술서에는 프로젝트에서 사용할 정의된 프로세스를 기술하고, 프로젝트의 정의된 프로세스가 조직의 표준 프로세스에 적합한지를 보장하기 위해 필요한 검증 활동을 기술한다. 또한 정의된 프로세스와 다른 프로세스와 상호작용을 기술한다.

GP 3.2.2의 기록은 프로젝트에 참여하는 직원이 필요한 권한, 기술, 경험, 지식을 가지고 있다는 증거를 제시한다. GP 3.2.3의 기록은 프로젝트에 참여하는 직원이 요구되는 훈련을 받았다는 증거를 제시한다. GP 3.2.5에서 기록은 정의된 프로세스를 수행하기 위해 프로젝트의 기반구조와 업무환경이 이용 가능하고 유지관리 된다는 증거를 제시한다.

표 8: PA 3.2의 공통지표, 공통 자원, 공통 작업산출물

PA 3.2 (프로세스 전개 속성) 프로세스 수행결과(outcomes)를 달성하기 위해 '표준 프로세스'가 '정의된 프로세스'로서 효과적으로 전개된 정도를 측정하는 측도이다.						
GP	GP vs. GR				GP vs. GWP	
GP 3.2.1 (a) 정의된 프로세스의 전개 표준 프로세스 조정하여 정의된 프로세스를 만들고 표준 프로세스 요구사항에 적합한지 검증					08-00 계획서 10-00 프로세스 기술서 14-00 등록	
GP 3.2.2 (b) 정의된 프로세스를 전개하기 위한 역할, 책임과 권한을 배정하고 통보					08-00 계획서 13-00 기록	
GP 3.2.3 (c) 정의된 프로세스를 수행하는데 필요한 역량 보증: 배정된 직원의 역량을 식별하고, 직원에게 적합한 훈련이 가능하다.					13-00 기록	
GP 3.2.4 (d) 정의된 프로세스를 수행을 지원하는 (인적) 자원 및 정보 제공					13-00 기록 16-00 저장소	
GP 3.2.5 (e) 정의된 프로세스 수행을 지원하는 프로세스 기반 구조를 제공 (작업 환경 포함)					13-00 기록	
GP 3.2.6 (f) 프로세스의 적절성과 효과성 증명을 위해 프로세스 수행에 대한 데이터를 수집하고 분석: 프로세스 개선에 사용					08-00 계획서 03-00 데이터 07-00 측도 14-00 등록 15-00 보고	

프로세스 저장소  
 자원 관리 시스템  
 지식 관리 시스템  
 작업환경 및 기반구조

- 피드백 절차 (고객, 직원, 기타 이해관계자)
- 문제 및 변경 관리 시스템
- 데이터 수집 분석 시스템
- 프로세스 심사 프레임워크
- 감리/검토 시스템

GP 3.2.4의 저장소에는 정의된 프로세스를 수행하기 위하여 정보가 이용 가능하다는 증거를 제시한다. GP 3.2.6에서는 측도를 사용하여 데이터를 수집하고 분석하여 프로세스 개선에 이용한다. 이와 관련된 모든 자료는 저장소에 저장한다.

## 제 4 장 소프트웨어 프로세스 개선과 심사

조직에서 프로세스와 관련된 활동을 프로세스 정립 및 구현, 그리고 프로세스 개선으로 크게 대별하면 다음과 같다.

(1) 프로젝트 수준에서 프로세스 기술서를 만들고 이를 발전시켜 조직 수준에서 표준 프로세스의 일환으로 프로세스 기술서를 만드는 것은 프로세스 정립 (establishment) 활동이고, 이에 따라 업무를 수행하는 것은 프로세스 구현 활동이다.

(2) 기업의 사업 목표를 보다 더 효과적이고 효율적으로 달성하기 위하여 복잡하고 비능률적인 프로세스를 단순하고 능률적인 프로세스로 변경하고, 시간이 많이 걸리고 결함이 많은 작업은 새로운 도구를 도입하여 이를 타개하는 것은 소프트웨어 프로세스 개선 활동이다.

프로세스 개선을 규정한 ISO/IEC 15504-4에서는 그림 7과 같이 8단계로 개선 활동을 규정하고 있다.

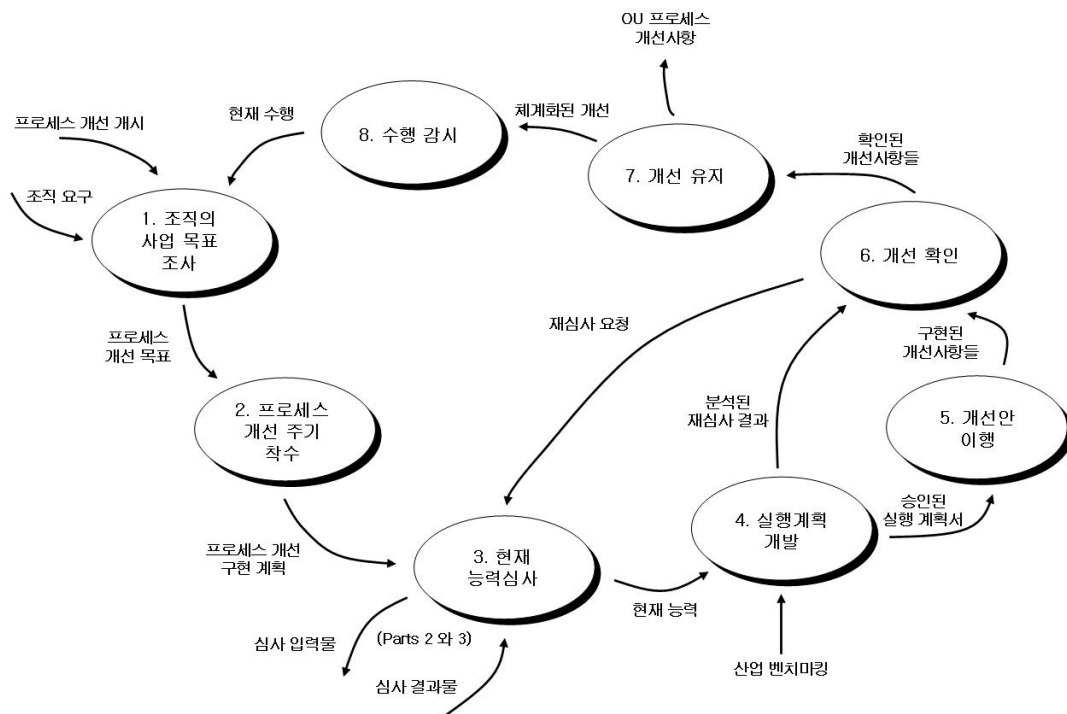


그림 7: 프로세스 개선의 단계 (15504-4)

조직의 사업 목표와 개선 동기 분석을 통해 프로세스 개선의 목적이 수립되고 (단계 1), 프로세스 개선 프로그램은 후원자, 프로젝트 관리, 예산, 이정표, 책임이 규정된 하나의 프로젝트로 진행되어야 한다(단계 2). 세 번째 단계가 현 프로세스의 능력과 강약점을 찾아내어 개선의 기초를 마련하는 프로세스 심사 활동이다. 즉, 프로세스 개선의 관점에서 프로세스 심사는 프로세스의 강약점을 찾아내어 개선안을 마련하는 수단이다.

자신의 조직을 가장 잘 이해하는 사람은 개선 의지를 가진 조직 구성원이다. 따라서, 조직의 능력이 일정 수준에 이르면, 조직내의 전문가가 자신의 조직을 심사하는 자체 심사가 활성화 되며, 자체 심사는 ISO/IEC 15504에서도 권고하는 사항이다. 필요에 따라 외부 심사원이 참여하는 심사가 필요하기도 하나 자체 심사가 활성화 되도록 해야 할 것이다.



## 참고문헌

- [1] Final Report of Phase 2 of the SPICE trials: Volume 1, Executive summary and detailed reports of some of the studies, Version 1.01, January 2003.
- [2] Final Report of Phase 2 of the SPICE trials: Volume 2, Published paper that summarizes the most significant studies by Ho-Won Jung et al., Dec. 2001.
- [3] Final Report of Phase 2 of the SPICE trials: Annotated Bibliography.
- [4] K. El Emam, A. Birk, Validating the ISO/IEC 15504 measure of software development process capability, *Journal of Systems and Software* 51(2) (2000) 119-149.
- [5] K. El Emam, A. Birk, Validating the ISO/IEC 15504 measure of software requirement analysis process capability, *IEEE Transactions on Software Engineering* 26(6) (2000) 541-566.
- [6] Ho-Won Jung, Robin Hunter, Dennis Goldenson, Khaled El Emam, "Findings from Phase 2 of the SPICE Trials", *Software Process Improvement and Practice: International Journal*, Vol. 6, No 5, 205-242, 2001.
- [7] Ho-Won Jung, Robin Hunter, "Evaluating the SPICE Rating Scale with regard to the Internal Consistency of Capability Measures," *Software Process Improvement and Practice*, Vol. 8, Issue 3, 169-178, 2003.
- [8] M. Sanders (Ed.), *The SPIRE Handbook: Better, Faster, Cheaper Software Development in Small Organizations*, The SPIRE Project Team: ESSI Project 23873, 1998.
- [9] T. Rout , K. El Emam, M. Fusani, D. Goldenson, Ho-Won Jung, "SPICE in Retrospect: Developing a Standard for Process Assessment", *Journal of Systems and Software*, Vol. 80, No. 9, 1483-1493, 2007.

---

## 제 2 부 : 표준 프로세스

## 제 5 장 엔지니어링 프로세스 그룹(ENG)

- ENG.1 요구사항 도출 프로세스
- ENG.4 소프트웨어 요구사항 분석 프로세스
- ENG.5 소프트웨어 설계 프로세스
- ENG.6 소프트웨어 구축 프로세스
- ENG.7 소프트웨어 통합 프로세스
- ENG.8 소프트웨어 시험 프로세스
- ENG.11 소프트웨어 설치
- ENG.12 소프트웨어 및 시스템 유지관리



# ENG.1 요구사항 도출 프로세스

## 1. 개요

### 1.1. 목적

요구사항 도출 프로세스의 목적은 요구되는 작업산출물을 정의하기 위한 근거가 되는 요구사항 베이스라인을 수립하기 위하여, 제품 및/또는 서비스의 전 수명에 걸쳐 변화하는 고객의 니즈 및 요구사항을 수집, 처리, 추적하는 것이다.

### 1.2. 설명

요구사항 도출 프로세스는 프로젝트 전 기간에 걸쳐 수행된다. 프로젝트가 진행되면서 요구사항은 새로운 기술, 새로운 요구, 환경 등의 요인에 의해 변한다. 이 프로세스는 이렇게 변화하는 요구사항을 체계적으로 관리함으로써 고객의 니즈를 충족할 수 있도록 하고, 요구사항 변화에 따라 기존에 만들어진 작업산출물들이 일관성 있게 변경할 수 있도록 해 준다. 또한 고객이 요청한 사항들에 대해 지속적으로 대응하면서 그 내용을 고객들에게 알려 준다.

### 1.3. 주요 관련 프로세스

ID	프로세스	설명
ENG.4	소프트웨어 요구사항 분석	요구사항 도출 결과는 소프트웨어 요구사항 분석에 입력이 되고, 분석을 통해 요구사항 베이스라인을 수립하도록 한다.
SUP.8	형상관리 프로세스	요구사항 변경에 따라 변화되어야 하는 작업산출물은 형상관리를 통해 처리된다.

### 1.4. 용어 정의

- 요구사항: 문제를 해결하거나 목적을 달성하기 위하여 사용자가 필요로 하는 조건 또는 기능
- 비즈니스 요구사항: 시스템을 구축하는 조직 또는 시스템을 구매하는 고객의 추상적인 비즈니스 목표를 말함
- 베이스라인: 공식적으로 검토되고 승인된 사항들로 향후 개발을 위한 근거로

사용되며, 공식적인 변경 처리 절차를 통해서만이 변경 가능함

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트 계획수립 단계가 시작됨
입력물	[13-16]변경요청서 [13-17]고객요청서 [17-03]고객 요구사항
수행활동	고객 요구사항 확보 고객 요구사항 점검 고객 요구사항 확인 고객 요구사항 변경관리 고객 응답체계 수립
출력물	[13-04]의사소통 기록 [13-21]변경관리기록 [17-03]고객 요구사항 [13-22]추적 기록
완료조건	프로젝트가 종료됨

### 2.2. 책임과 역할

활동	역할	요구사항 도출담당자	개발자	프로젝트 관리책임자	고객
	1. 고객 요구사항 확보		○		●
2. 고객 요구사항 점검		●		○	○
3. 고객 요구사항 합의		○		●	○
4. 고객 요구사항 변경관리		●	○		○
5. 고객 응답 체계 수립		●			○

(●: 주관, ○: 지원)

### 3. 프로세스 활동

#### 3.1. 고객 요구사항 확보

고객 및 사용자의 입력을 구체화하여 고객의 요구사항 및 요청을 확보한다.

☞ 고객 요구사항은 고객의 사업 제안서, 목표 운영 환경 및 고객의 요구사항을 포함하는 다음과 같은 다른 문서의 검토를 통하여 확보될 수 있다.

- 운영개념서
- 시스템 개념서
- 마케팅 문서
- 제안요청서
- 외부시스템 인터페이스

☐ 산출물: [17-03]고객 요구사항

#### 3.2. 고객 요구사항 점검

개발자와 고객이 함께 요구사항을 동일하게 이해함을 확인하기 위해 고객과 함께 요구사항을 검토하고 요구사항의 실현 가능성과 타당성을 점검한다.

고객과 직접 관련이 없는 환경적, 법적 및 다른 제약사항을 고려할 필요가 있다.

☞ 고객과 함께 요구사항 및 요청을 검토하는 방법에는 기존 시스템을 관찰하거나, 프로토타입, 시뮬레이션, 모델, 시연 등이 포함된다.

☐ 산출물: [17-03]고객 요구사항

#### 3.3. 고객 요구사항 합의

요구사항을 구현하기 위해 개발팀 및 고객과 합의한 고객 요구사항에 서명하여 공식화함으로써 고객 요구사항에 대한 당사자들의 합의를 확보한다.

☞ 합의한 내용은 회의록을 남기거나, 작성된 요구사항 정의서의 고객 확인란에 서명하게 한다.

☐ 산출물: [17-03]고객 요구사항

### 3.4. 고객 요구사항에 대한 베이스라인을 수립

고객 요구사항을 공식화하여 사업에서 적용하거나, 고객의 니즈를 감시하기 위하여 베이스라인을 수립한다.

☞ 베이스라인된 고객 요구사항은 향후 변경시 공식적인 절차에 따라 변경될 수 있다.

☑ 산출물: [17-03]고객 요구사항(베이스라인된)

### 3.5. 고객 요구사항 변경 관리

고객의 요구사항이 변경될 경우 변경 요청을 작성하도록 하고, 변경에 따른 영향과 위험을 평가하여 수락 여부를 결정한다.

☞ 변경요청을 접수하여 평가 후 수락된 요구사항 변경을 처리하고, 요구사항이 변경됨에 따라 변경되어야 하는 관련 산출물을 추적하여 변경한다. 관련 산출물에는 설계서, 소스 코드 등이 있다.

☑ 산출물: [13-16]변경요청서, [13-22]추적 기록

### 3.6. 고객 응답 체계 수립

고객의 요구사항 변경에 대한 상태 및 처리결과를 고객에게 알려준다.

☞ 고객과 공식적인 합동 회의 또는 이메일을 등을 통해 고객에게 그 내역을 전달한다.

☑ 산출물: [13-04]의사소통 기록

## 4. 수행 결과

- 고객과 지속적으로 의사소통이 이루어진다.
- 고객과 요구사항을 합의하여 베이스라인이 수립된다.
- 고객 요구사항 변경 요청을 평가하여 요구사항으로 반영시킨다.
- 고객의 니즈를 지속적으로 관찰하여 반영할 수 있다.

- 고객 요청의 상태 및 처리 상황을 쉽게 파악할 수 있다.
- 기술 및 고객 니즈의 변경에 따른 개선사항을 식별하고 그 영향이 관리된다.

## 5. 측정 항목

측정항목(단위)	측정 방법
요구사항 개수(개)	현 요구사항 베이스라인에 포함된 요구사항 건수를 측정한다.
요구사항 변경요청 건수(개)	요구사항에 대한 변경요청이 접수된 건수를 측정한다.
요구사항 변경승인 건수(개)	요구사항 변경요청에 대해 변경이 승인된 건수를 측정한다.

(사용 예시)

- 요구사항 변경율(%):  $(\sum \text{요구사항 변경승인건수} / \sum \text{요구사항 개수}) \times 100$
- 요구사항 변경승인율(%):  $(\sum \text{요구사항 변경승인 건수} / \sum \text{요구사항 변경요청 건수}) \times 100$

## 6. 프로세스 조정

요구사항 도출의 과정 일부는 소프트웨어 요구사항 분석 프로세스의 일환으로 수행할 수 있다. 이 경우 요구사항 도출 프로세스는 단지 요구사항의 변경을 관리하는 기능만으로 구성할 수 있다.

## 7. 참고문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부: 프로세스 및 프로세스 능력 참조 모델, 2006



# ENG.4 소프트웨어 요구사항 분석 프로세스

## 1. 개요

### 1.1. 목적

소프트웨어 요구사항 분석 프로세스의 목적은 시스템의 소프트웨어 구성요소에 대한 요구사항을 수립하는 것이다.

### 1.2. 설명

고객의 요구사항 중 소프트웨어에 할당된 요구사항을 분석하여 최종적으로 고객이 원하는 소프트웨어 제품을 설계할 수 있도록 하는 과정이다. 소프트웨어 요구사항 분석을 통해 요구사항이 완전한지, 상충되는 점은 없는지 등을 평가하고, 그 내용을 문서화하여 설계로 자연스럽게 진행할 수 있도록 하는데 그 목적이 있다.

### 1.3. 주요 관련 프로세스

ID	프로세스	설명
ENG.1	요구사항 도출 프로세스	소프트웨어 요구사항 분석 프로세스에서 베이스라인으로 결정된 요구사항은 요구사항 도출 프로세스에서 관리된다.
ENG.5	소프트웨어 설계 프로세스	소프트웨어 요구사항을 충족시킬 수 있도록 소프트웨어 설계 프로세스에서 설계를 개발하여야 한다.

### 1.4. 용어 정의

- 요구사항(requirement): 문제를 해결하거나 목적을 달성하기 위하여 사용자가 필요로 하는 조건 또는 기능
- 요구사항 분석(requirement analysis): 사용자 니즈를 분석하여 시스템, 하드웨어 또는 소프트웨어 요구사항을 정의하는 과정
- 일관성(consistency): 문서 또는 구성요소 사이에 상충되는 것이 없는 정도

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	소프트웨어로 구현하여야 할 요구사항이 결정됨
입력물	[04-06]시스템 아키텍처 설계
수행활동	소프트웨어 요구사항을 명세 운영환경에 대한 영향을 결정 소프트웨어 시험을 위한 기준을 개발 일관성을 보장 소프트웨어 요구사항을 평가 및 갱신 소프트웨어 요구사항을 의사소통
출력물	[17-11]소프트웨어 요구사항 [17-08]인터페이스 요구사항 [13-22]추적 기록 [15-01]분석보고서 [13-04]의사소통 기록
완료조건	소프트웨어 요구사항문서가 검토되고 승인됨

### 2.2. 책임과 역할

활동	역할	요구사항	개발자	프로젝트	고객
		분석담당자		관리책임자	
1. 소프트웨어 요구사항을 명세		●	○		
2. 운영 환경에 대한 영향을 결정		●	○		
3. 소프트웨어 시험을 위한 기준을 개발		○	○	●	○
4. 일관성을 보장		○		●	
5. 소프트웨어 요구사항을 평가 및 갱신		○	○	●	○
6. 소프트웨어 요구사항을 의사소통		○	○	●	○

(●: 주관, ○: 지원)

### 3. 프로세스 활동

#### 3.1. 소프트웨어 요구사항을 명세

시스템의 소프트웨어 구성요소의 기능적 및 비기능적 요구사항과 그 인터페이스를 정의하고 우선 순위화하고, 이들을 문서화한다. 정확성, 완벽성, 일관성, 타당성 및 시험 가능성에 대하여 소프트웨어 요구사항을 분석한다. 모든 도출된 요구사항을 식별한다.

☞ ISO/IEC 9126에는 다음과 같은 소프트웨어의 품질 특성을 정의하고 있다. 이러한 품질 특성을 고려하여 요구사항을 명세한다.

- 기능성
- 신뢰성
- 사용성
- 유지보수성
- 효율성
- 이식성

소프트웨어 요구사항은 다음과 같은 특성을 가져야 한다.

- 완전성(completeness): 사용자의 요구사항이 누락 없이 작성되어야 한다.
- 일관성(consistency): 사용자 요구사항 내부 혹은 사용자 요구사항 간에 불일치가 없어야 한다.
- 정확성(correctness): 사용자 요구사항에서 제시된 모든 사용자 기능이 만족되어 시스템이 원래 사용자 요구를 충족시켜야 한다.
- 테스트 용이성(testability): 사용자 요구사항에서 제시된 항목들은 테스트에 의해서 검증 가능하여야 한다.
- 명확성(unambiguity): 사용자 요구사항의 기술은 명확하여 애매모호함이 없어야 한다.
- 설계와의 독립성(design freedom): 사용자 요구사항 정의로 인해 설계 대안에 제약을 주어서는 안된다.
- 전달성(communicability): 사용자 요구사항은 명시적으로 기술되어 이해가 용이하도록 하고, 애매모호한 점이 없어 의사전달이 쉬워야 한다.
- 모듈화 및 변경에 대한 견고성(modularity/change robustness): 사용자 요구사항의 작성시에 새로운 항목을 추가하고, 기존 항목을 수정 및 변경할 때 다른 부분에 커다란 영향을 미치지 않도록 기술되는 항목들이 구조화되어 있어야 한다.

- 필요성(necessity): 기술된 요구사항들은 사용자 요구사항의 목적에 부합하여 원래의 사용자 요구 목적에 기여하여야 한다.

☐ 산출물: [17-11]소프트웨어 요구사항

### 3.2. 운영 환경에 대한 영향을 결정

소프트웨어 요구사항과 운영 환경의 다른 구성요소 간 인터페이스를 정의하고, 요구사항이 미치는 영향을 정의한다.

☞ 운영 환경에는 소프트웨어의 사용자가 수행하는 업무 또는 사용자가 사용하는 다른 시스템을 포함한다.

인터페이스의 예로는 다음과 같은 것이 있다.

- 시스템을 연결하기 위한 인터페이스
- 소프트웨어와 사용자 간의 인터페이스
- 소프트웨어와 하드웨어 구성요소간의 인터페이스
- 타 소프트웨어(예. DBMS, OS, 패키지 등)와의 인터페이스
- 프로토콜과 같은 통신 인터페이스
- 초기화 절차, 수정 내용 등과 같은 사이트 적용 요구사항

☐ 산출물: [17-08]인터페이스 요구사항

### 3.3. 소프트웨어 시험을 위한 기준을 개발

소프트웨어 요구사항을 사용하여 소프트웨어 제품 시험을 위한 인수 조건을 정의한다. 소프트웨어 제품 시험은 소프트웨어 요구사항과 부합됨을 입증하는 것이다.

☐ 산출물: 해당 사항 없음

※ 인수 조건은 소프트웨어 요구사항 또는 인터페이스 요구사항을 기술할 때 인수 조건을 포함시킨다. 이 조건은 [ENG.8]소프트웨어 시험 프로세스에서 활용한다.

### 3.4. 일관성을 보장

시스템 요구사항 분석과 소프트웨어 요구사항 분석 사이의 일관성을 보장한다. 필요시 시스템 요구사항과 소프트웨어 요구사항 사이의 추적성을 수립하고 유지함으로써 일관성을 지원한다.

☐ 산출물: [13-22]추적 기록

※ [13-22]추적 기록을 활용하여 일관성을 유지한다.

### 3.5. 소프트웨어 요구사항을 평가 및 갱신

고객과 함께 요구사항을 평가하고, 비용, 일정 및 기술적 영향에 대한 제안된 변경의 영향을 평가하고, 변경을 승인 또는 기각하고, 소프트웨어 요구사항 명세서를 갱신한다.

☐ 산출물: 해당 사항 없음

※ ENG.1 요구사항 도출 프로세스를 활용하여 요구사항 변경을 처리한다.

### 3.6. 소프트웨어 요구사항을 의사소통

소프트웨어 요구사항의 배포에 대한 의사소통 절차와 수단을 수립하고 이를 사용하는 모든 당사자들에게 요구사항을 갱신하여 준다.

☐ 산출물: [13-04]의사소통 기록

## 4. 수행결과

- 시스템의 소프트웨어 구성요소 및 그 인터페이스에 할당된 요구사항이 정의된다;
- 정확성 및 시험 가능성에 대한 소프트웨어 요구사항이 분석된다;
- 운영 환경에 대한 소프트웨어 요구사항의 영향이 이해된다;
- 시스템 요구사항과 소프트웨어 요구사항 사이의 일관성과 추적성이 수립된다;
- 소프트웨어 요구사항 구현을 위한 우선순위가 정의된다;
- 소프트웨어 요구사항이 승인되고 필요에 따라 갱신된다;
- 소프트웨어 요구사항 변경의 비용, 일정, 기술적 영향이 평가된다;
- 소프트웨어 요구사항이 베이스라인 되며 영향받는 당사자들에게 의사소통된다.

## 5. 측정 항목

측정 항목(단위)	측정 방법
요구사항 개수(개)	소프트웨어 요구사항 개수를 측정한다.
인터페이스 개수(개)	소프트웨어 인터페이스 개수를 측정한다.

(사용 예시)

- 요구사항 개수
- 인터페이스 개수

## 6. 프로세스 조정

[17-08]인터페이스 요구사항은 [17-11]소프트웨어 요구사항 내에서 함께 정의할 수 있다. 요구사항을 초기에 명확히 정의하기 어려워 개발이 진행되면서 확인이 가능한 경우 소프트웨어 시험을 위한 기준준비는 미루어질 수 있다.

## 7. 참고문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

# ENG.5 소프트웨어 설계 프로세스

## 1. 개요

### 1.1. 목적

소프트웨어 설계 프로세스의 목적은 요구사항을 구현하고 검증할 수 있는 소프트웨어 설계를 제공하는 것이다.

### 1.2. 설명

소프트웨어 설계는 소프트웨어 요구사항이 어떻게 구현 되는가에 관한 것이다. 소프트웨어 요구사항을 충족할 수 있는 가능한 해결 방안(소프트웨어 아키텍처)을 찾고, 평가를 통해 최적의 방안을 선택한다. 선택된 해결 방안에 대해 구현할 수 있을 정도로 자세하게 상세 설계를 개발하여 소프트웨어를 구현할 수 있도록 한다.

### 1.3. 주요 관련 프로세스

ID	프로세스	설명
ENG.4	소프트웨어 요구사항 분석 프로세스	소프트웨어 요구사항을 기반으로 그 요구사항을 충족시킬 수 있도록 설계 프로세스가 수행된다.
ENG.6	소프트웨어 구축 프로세스	작성된 소프트웨어 설계를 기반으로 소프트웨어가 구축된다.

### 1.4. 용어 정의

- 소프트웨어 설계(design): 소프트웨어의 아키텍처, 구성요소, 인터페이스 및 기타 특성을 정의하는 과정
- 소프트웨어 아키텍처 설계(architecture design): 소프트웨어 개발을 위한 틀 (framework)을 제공하기 위해 소프트웨어 구성요소와 그들간의 인터페이스들을 정의하는 과정
- 인터페이스(interface): 하나의 구성 요소에서 다른 구성요소로 정보를 전달할 목적으로 두 개 이상의 구성요소를 연결하는 구성요소
- 상세 설계(detailed design): 구현할 수 있을 정도로 충분히 아키텍처 설계를 정제하고 확장하는 과정
- 소프트웨어 설계서: 생성할 소프트웨어 시스템의 모형 또는 표현

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	소프트웨어 요구사항이 베이스라인됨
입력물	[17-11]소프트웨어 요구사항
수행활동	소프트웨어 아키텍처를 기술 인터페이스를 정의 상세설계를 개발 설계의 시험 가능성을 분석 일관성을 보장
출력물	[04-04]상위수준 소프트웨어 설계 [04-05]하위수준 소프트웨어 설계 [11-02]소프트웨어 구성요소 [13-22]추적 기록
완료조건	설계가 승인됨

### 2.2. 책임과 역할

활동	역할	요구사항	구조설계	상세설계	프로젝트
		분석담당자	담당자	담당자	관리책임자
1. 소프트웨어 아키텍처를 기술		○	●		
2. 인터페이스를 정의			●	○	
3. 상세 설계를 개발				●	
4. 설계의 시험 가능성을 분석			●	○	
5. 일관성을 보장		○	○	○	●

(●: 주관, ○: 지원)



### 3. 프로세스 활동

#### 3.1. 소프트웨어 아키텍처를 기술

소프트웨어 요구사항을 소프트웨어 아키텍처 설계로 변환한다. 소프트웨어 아키텍처 설계는 상위 수준의 구조를 기술하고 주요 소프트웨어 구성요소를 식별한 것이다.

☞ 소프트웨어 요구사항을 기반으로 상위수준 소프트웨어를 설계하면서 구성요소를 정의한다. 주요 소프트웨어 구성요소의 예로는 데이터베이스, 의사소통 절차와 수단, 비즈니스 로직 및 사용자 인터페이스가 포함된다.

☞ 소프트웨어 요구사항을 소프트웨어 아키텍처 설계로 변환할 때 사전에 정의된 기준에 따라 소프트웨어 아키텍처 대안들을 평가한다. 현재의 소프트웨어 아키텍처가 선택된 논리를 기록하여야 한다. 평가 기준에는 품질 특성(모듈화, 유지 관리성, 확장성, 가변성, 신뢰성, 보안성 및 사용의 편리성) 및 개발-구매-재사용 분석의 결과가 포함될 수 있다.

☞ 아키텍처 설계에서는 설계의 중요한 측면을 표현하기 위해 다음과 같은 여러 관점을 포함시킨다. 각 관점은 소프트웨어 시스템에 대한 별도의 관심사항을 표현하는 것이다.

- 분해 관점: 소프트웨어 시스템을 여러 구성요소로 나누고, 각 구성요소가 특정한 기능을 수행하도록 할당한다. 계층적 분해도를 활용할 수 있다.
- 의존성 관점: 소프트웨어 시스템의 실행관점에서 구성요소간의 상호작용을 이해하기 위한 것이다. 자료흐름도, 순서도 및 트랜잭션 다이어그램을 활용할 수 있다.

☐ 산출물: [04-04]상위수준 소프트웨어 설계, [11-02]소프트웨어 구성요소

#### 3.2. 인터페이스를 정의

소프트웨어 구성요소 사이의 외부 및 내부 인터페이스를 명세화하고 문서화한다.

☞ 인터페이스는 화면 형식, 유효한 입력, 결과 출력, 통신 가능 언어를 제공하여야 한다. ENG.4 요구사항 분석 프로세스에서 운영 환경에 대한 영향을 결정할 때 인터페이스는 소프트웨어 구성요소가 아닌 외부와의 인터페이스를 정의하는 것이다.

인터페이스 정의는 상위수준 소프트웨어 설계에 포함된다.

☐ 산출물: [04-04]상위수준 소프트웨어 설계

### 3.3. 상세 설계를 개발

소프트웨어 아키텍처 설계로부터, 생성되어야 할 모든 소프트웨어 단위를 기술하는 각 소프트웨어 구성요소를 위한 상세한 설계로 분해한다. 소프트웨어 단위 및 인터페이스를 소프트웨어 설계서로 문서화한다.

☞ 상세설계는 소프트웨어 단위의 내부 구조를 정의하고, 데이터 스키마를 생성하고, 알고리즘을 개발하는 것 등을 통해 소프트웨어 단위에 할당된 요구사항을 충족할 수 있도록 하는 과정이다. 소프트웨어 단위에 대해 구현자가 충분히 이해할 수 있을 정도로 내용을 기술한다.

☐ 산출물: [04-05]하위수준 소프트웨어 설계

### 3.4. 설계의 시험 가능성을 분석

소프트웨어 단위가 구축되고 시험될 수 있음을 보장하기 위하여 설계의 정확성과 시험 가능성을 분석한다.

☐ 산출물: 해당 사항 없음

※ ENG.4 소프트웨어 요구사항 분석 프로세스에서의 소프트웨어 시험 기준 개발은 소프트웨어가 목적에 맞게 구현되었는가를 확인하기 위한 것과 구분하여야 한다.

### 3.5. 일관성을 보장

소프트웨어 요구사항 분석과 소프트웨어 설계 사이의 일관성을 보장한다. 필요시 소프트웨어 요구사항과 소프트웨어 설계 사이의 추적성을 수립하고 유지하여 일관성을 지원한다.

☐ 산출물: 해당 사항 없음

※ [13-22]추적 기록을 활용하여 일관성을 유지한다.

#### 4. 수행결과

- 소프트웨어 요구사항을 구현하는 소프트웨어 구성요소들을 기술하는 소프트웨어 아키텍처 설계가 개발되고 베이스라인이 이루어진다;
- 각 소프트웨어 구성요소들의 내부 및 외부 인터페이스가 정의된다;
- 구축되고 시험될 수 있는 소프트웨어 단위를 기술하는 상세한 설계가 개발된다;
- 소프트웨어 요구사항과 소프트웨어 설계 사이의 일관성과 추적성이 수립된다.

#### 5. 측정항목

측정항목(단위)	측정 방법
요구사항 개수(개)	소프트웨어요구사항 개수를 측정한다.
소프트웨어 구성요소의 수(개)	소프트웨어구성요소 개수를 측정한다.
소프트웨어 설계에 반영된 요구사항 수(개)	소프트웨어 설계에 반영된 소프트웨어 요구사항 개수를 측정한다.

(사용 예시)

- 요구사항 개수
- 소프트웨어 설계 반영율(%) = (소프트웨어 설계에 반영된 요구사항의 수)/(요구사항의 개수)

#### 6. 프로세스 조정

소프트웨어 아키텍처와 상세설계를 통합하여 하나의 소프트웨어 설계서로 만들 수도 있다.

#### 7. 참고문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions

## ENG.6 소프트웨어 구축 프로세스

### 1. 개요

#### 1.1. 목적

소프트웨어 구축 프로세스의 목적은 소프트웨어 설계를 적절히 반영하는 실행 가능한 소프트웨어 단위를 생성하는 것이다.

#### 1.2. 설명

본 프로세스는 최종 소프트웨어 제품을 구축할 때 구성 요소가 되는 소프트웨어 단위를 만들어내는 것이다. 소프트웨어 설계 프로세스에서 개발한 상세 설계에 기반하여 소프트웨어 단위를 실행 가능한 소프트웨어 단위로 만든다. 후속 단계에서 결함을 발견하고 시정하게 되면 많은 비용과 시간이 소요되므로 단위 시험을 철저히 수행하는 것이 필수적이다.

#### 1.3. 주요 관련 프로세스

ID	프로세스	설명
ENG.5	소프트웨어 설계 프로세스	소프트웨어 설계에서 정의된 소프트웨어 단위를 소프트웨어 구축 프로세스에서 구현한다.
ENG.7	소프트웨어 통합 프로세스	구축이 완료된 소프트웨어 단위를 소프트웨어 통합 프로세스에서 전략에 따라 결합하면서 시험한다.

#### 1.4. 용어 정의

- 코딩(coding): 설계 명세서의 논리(logic) 및 데이터를 프로그래밍 언어로 변환하는 것
- 단위(unit): 소프트웨어 설계에서 명시된 독립적으로 테스트 가능한 구성요소. 논리적으로 분리 가능한 컴퓨터 프로그램의 일부분
- 단위 시험(unit testing): 개별 소프트웨어 단위를 시험하는 것

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	소프트웨어 설계서가 준비됨
입력물	[04-05]하위수준 소프트웨어 설계
수행활동	단위 검증 절차를 개발 소프트웨어 단위를 개발 일관성을 보장 소프트웨어 단위를 검증
출력물	[08-25]단위시험 계획서 [11-05]소프트웨어 단위 [03-07]시험데이터 [14-04]시험 로그 [13-22]추적 기록
완료조건	소프트웨어 단위가 구현됨

### 2.2. 책임과 역할

활동 \ 역할	구현담당자	프로젝트 관리책임자
1. 단위 검증 절차를 개발	●	○
2. 소프트웨어 단위를 개발	●	
3. 일관성을 보장	●	
4. 소프트웨어 단위를 검증	●	○

(●: 주관, ○: 지원)

### 3. 프로세스 활동

#### 3.1. 단위 검증 절차를 개발

설계 요구사항을 만족하는 각 소프트웨어 단위를 검증하기 위한 절차 및 기준이 개발되고 문서화된다. 검증 절차에는 단위 시험 사례 및 코드 검토가 포함된다.

☞ 검증 절차에는 가용한 기법을 사용하여 원하는 품질을 달성할 수 있는가를 정의한다. 가능한 기법에는 정적/동적 분석, 코드 검토, 코드 커버리지 분석 등이 있다.

☐ 산출물: [08-25]단위시험계획서, [03-07]시험데이터

#### 3.2. 소프트웨어 단위를 개발

각 소프트웨어 단위를 프로그래밍하여 실행 가능한 형태를 개발하고 문서화한다. 시험 요구사항 및 사용자 문서를 갱신한다.

사용자 문서는 설치, 운영 및 유지관리 문서의 초기 버전을 포함한다.

☞ 적용 가능한 소프트웨어 코딩 표준과 기준을 따른다. 소프트웨어 코딩 표준에는 프로그래밍 언어, 변수 명명 규칙, 언어 사용 규칙, 주석(comments) 형식 등이 포함된다. 코딩할 때 적용할 수 있는 기준으로는 모듈화, 단순화, 구조화, 유지보수용이성 등의 기준을 적용할 수 있다.

☞ 소프트웨어 단위 개발 방법으로는 사용 언어에 따라 구조적 프로그래밍, 객체 지향 프로그래밍 등으로 나뉠 수 있고, 코드 자동 생성 방법, 재사용, 설계 패턴 사용 등의 방법을 활용할 수 있다.

☐ 산출물: [11-05]소프트웨어 단위

#### 3.3. 일관성을 보장

소프트웨어 설계와 소프트웨어 구축 사이의 일관성을 보장한다.

☐ 산출물: [13-22]추적 기록

※ [13-22]추적 기록을 활용하여 설계와 구축, 나아가서는 시험데이터와의 추적 관계를 유지한다.

### 3.4. 소프트웨어 단위를 검증

명기된 단위 검증 절차를 수행함으로써 각 소프트웨어 단위가 그 설계 요구사항을 만족하고 있음을 검증하고, 그 결과를 기록한다.

☞ 3.1에서 개발된 단위 검증 절차를 사용하여 소프트웨어 단위를 검증한다. 검증 과정에서 발견된 결함 또는 에러는 수정하고 그 결과를 기록한다.

☐ 산출물: [14-04]시험 로그

## 4. 수행결과

- 요구사항에 대한 모든 소프트웨어 단위의 검증 기준이 정의된다;
- 설계에서 정의된 소프트웨어 단위가 작성된다;
- 소프트웨어 요구사항, 설계 및 소프트웨어 단위 사이의 일관성과 추적성이 수립된다;
- 요구사항 및 설계에 대한 소프트웨어 단위의 검증이 이루어진다.

## 5. 측정항목

측정항목(단위)	측정 방법
소프트웨어 단위 수	구현하여야 할 소프트웨어 단위의 수를 측정한다.
소스코드 라인 수(줄)	주석(comments)을 제외한 소스코드의 라인 수를 측정한다.

(사용 예시)

- 소스코드 라인 수
- 단위당 평균 소스코드 라인 수(줄)=(소스코드 라인 수)/(소스코드 단위 수)

## 6. 프로세스 조정

단위 시험이 개발자 수준에서 이루어지는 경우 [08-25]단위시험계획서는 작성하지 않을 수도 있다.

## 7. 참고문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006



## ENG.7 소프트웨어 통합 프로세스

### 1. 개요

#### 1.1. 목적

소프트웨어 통합 프로세스의 목적은 소프트웨어 단위를 결합하여 소프트웨어 설계와 일치하는 통합된 소프트웨어 항목(item)을 생성하고, 동일한 또는 완벽한 운영 플랫폼에서 기능적 및 비기능적 소프트웨어 요구사항을 만족시킴을 보여주는 것이다.

#### 1.2. 설명

본 프로세스는 소프트웨어 단위가 개발되어 통합될 때 적용된다.

소프트웨어 통합은 소프트웨어 테스트에 앞서 통합된 소프트웨어 제품을 구성하는 소프트웨어 단위들이 정확하고, 안정적으로 인터페이스되고, 상호작용함을 보여주는 것이다. 주로 기능 요구사항을 중심으로 시험을 수행한다. 테스트 케이스를 설계할 경우 소프트웨어 단위들이 정확히 인터페이스 되고 상호 작용하는지에 초점을 맞추고, 단위 테스트 때 수행한 내용들을 중복하는 것은 피하도록 한다. 하지만 단위 테스트에서 사용했던 데이터를 재사용할 수는 있다.

소프트웨어 설계 프로세스에서 소프트웨어를 구성하는 단위를 정의하였다면, 소프트웨어 통합 프로세스에서는 그 구성 단위들이 잘 결합되는가를 보는 것이다.

#### 1.3. 주요 관련 프로세스

ID	프로세스	설명
ENG.6	소프트웨어 구축 프로세스	소프트웨어 구축 후 또는 구축 과정에서 소프트웨어 통합 프로세스가 수행된다.
ENG.8	소프트웨어 시험 프로세스	소프트웨어 통합 프로세스가 완료된 후 소프트웨어 시험 프로세스가 수행된다.

## 1.4. 용어 정의

- 통합(integration): 구성요소(소프트웨어, 하드웨어, 또는 둘 다)를 결합하여 전체 시스템으로 만드는 과정
- 통합 시험(integration testing): 소프트웨어 구성요소, 하드웨어 구성요소, 양자 모두를 결합하고, 그들 간의 상호작용을 평가하는 과정
- 시험 사례(test case): 시험을 위한 입력, 실행 조건, 기대되는 결과 등으로 이루어진 데이터 집합
- 회귀 시험(regression testing): 소프트웨어에 가한 변경 사항이 의도되지 않은 결과를 발생시키지 않고 명시된 요구사항에 부합함을 검증하기 위해 소프트웨어를 선택적으로 재시험하는 것

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	통합을 위한 소프트웨어 단위들 준비됨
입력물	[01-03]소프트웨어 항목 [04-04]상위수준 소프트웨어 설계 [11-05]소프트웨어 단위
수행활동	소프트웨어 통합전략을 개발 통합된 소프트웨어 항목을 위한 시험을 개발 소프트웨어 항목을 통합 통합된 소프트웨어 항목을 시험 일관성을 보장 통합된 소프트웨어 항목을 회귀시험
출력물	[08-10]소프트웨어 통합 시험계획서 [10-02]시험절차서 [17-14]시험사례명세서 [14-04]시험 로그 [13-22]추적 기록 [11-01]소프트웨어 제품
완료조건	소프트웨어 통합시험 결과가 승인됨

## 2.2. 책임과 역할

활동	역할		
	통합담당자	구현담당자	프로젝트 관리책임자
1. 소프트웨어 통합 전략을 개발	●		○
2. 통합된 소프트웨어 항목을 위한 시험을 개발	●	○	
3. 소프트웨어 항목을 통합	●	○	
4. 통합된 소프트웨어 항목을 시험	●	○	
5. 일관성을 보장	●	○	○
6. 통합된 소프트웨어 항목을 회귀 시험	●	○	

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 소프트웨어 통합 전략을 개발

소프트웨어 요구사항을 고려하는 소프트웨어 단위 통합 전략을 개발한다. 소프트웨어 아키텍처에 근거하여 소프트웨어 항목을 식별하고, 이들을 통합 및 시험하기 위한 순서 또는 규칙을 정의한다.

☞ 소프트웨어 항목 통합 순서의 예로는 상위수준 항목, 하위수준 항목, 핵심 항목, 기능적 항목, 완벽한 구조 우선 및 가능한 항목 우선 등이 포함된다.

통합은 대개 개발팀에서 수행한다. 통합시험 계획서를 수립할 경우 결함을 시정하여 재시험하는데 요구되는 시간 및 노력을 고려하는 것이 필수적이다.

계속적으로 소프트웨어 항목이 추가될 경우 점진적 통합을 적용한다. 대부분 통합 과정은 실제(live) 환경에서 이루어지지 않고 개발 환경 또는 별도의 시험 환경에서 이루어지는 경우가 많다.

☐ 산출물: [08-10]소프트웨어 통합 시험계획서

### 3.2. 통합된 소프트웨어 항목을 위한 시험을 개발

각 통합된 소프트웨어 항목에 대해 수행되는 시험을 기술한다. 이때 요구사항이 점검됨을 나타내기 위하여 인터페이스의 검증, 입력 데이터 및 검증 기준을 포함한다.

☞ 시험사례는 동등분할(equivalent partitioning), 경계값 분석(boundary value analysis)과 같은 기법을 적용하여 시험 데이터가 소프트웨어 항목의 경계 조건을 시험할 수 있도록 함과 동시에 실제 대표적인 값을 제공할 수 있도록 한다.

시험사례로 실제 데이터(live data)를 활용하기 힘들 경우가 많다. 이때 대표적인 사례를 중심으로 시험이 가능한 시험 사례를 만드는 것이 중요하다.

통합을 위해 아직 구현되지 않은 소프트웨어 항목에 대해서는 스텝(stub), 드라이버(driver) 등을 미리 만들어 놓을 필요가 있다.

☐ 산출물: [10-02]시험절차서, [17-14]시험사례명세서

### 3.3. 소프트웨어 항목을 통합

소프트웨어 항목을 구성하기 위한 통합 전략에 따라 소프트웨어 단위를 통합한다.

☞ 통합과정은 데이터베이스, 스텝(stubs), 드라이버(drivers), 다른 필요한 실행 코드 등을 결합한다. 통합시 소프트웨어 항목의 적절한 버전이 결합될 수 있도록 하여야 한다. 이를 위해 체계적인 형상관리가 필요하다.

☐ 산출물: [11-01]소프트웨어 제품

### 3.4. 통합된 소프트웨어 항목을 시험

검증 기준에 따라 운영 플랫폼 또는 적절한 동일 플랫폼에서 통합된 소프트웨어 항목을 시험하고 결과를 기록한다. 필요 시 사용자 문서를 갱신한다.

☞ 통합은 그 방식에 따라 차례적으로 반복적하여 수행하고, 기대결과와 실제 결과를 비교하여 통합 시험의 성공 여부를 결정하며 그 결과를 시험로그에 기록한다. 시험 과정에서 나타나는 이상 현상은 보고되어야 한다.

☐ 산출물: [14-04]시험 로그

### 3.5. 일관성을 보장

소프트웨어 설계와 소프트웨어 통합 사이의 일관성을 보장한다. 필요시 소프트웨어 설계와 소프트웨어 항목 사이의 추적성을 수립하고 유지하여 일관성을 지원한다.

☐ 산출물: [13-22]추적 기록

### 3.6. 통합된 소프트웨어 항목을 회귀 시험

통합된 소프트웨어 항목을 재시험하기 위한 소프트웨어 회귀시험 전략을 개발한다. 소프트웨어 단위, 설계 또는 요구사항에 변경이 있을 경우 이 전략에 따라 회귀 시험을 수행한다.

☞ 회귀 시험은 소프트웨어 항목이 변경 또는 확장되었을 경우에도 여전히 정확하게 작동하는가를 확인하기 위한 과정이다. 효율적, 효과적으로 회귀시험을 하기 위해 이전에 만들어진 시험 사례를 활용하지만, 대부분 모든 시험 사례를 다시 수행하지는 않는다. 변경에 의해 영향을 받는 범위를 식별하여 회귀 시험에서 적용한 시험사례를 선정한다. 자동화된 시험도구를 사용할 경우 이전의 시험 사례를 재수행할 수 있다.

☐ 산출물: [14-04]시험 로그

## 4. 수행결과

- 우선순위가 결정된 소프트웨어 요구사항 및 소프트웨어 설계와 일관성 있게 소프트웨어 단위를 개발하는 통합 전략이 개발된다;
- 소프트웨어 항목에 할당된 소프트웨어 요구사항이 준수됨을 보장하기 위하여 소프트웨어 항목에 대한 검증 기준이 개발된다;
- 소프트웨어 항목이 정의된 기준에 따라 검증된다;
- 통합전략에 따라 정의된 소프트웨어 항목들이 생성된다;
- 통합시험의 결과가 기록된다;
- 소프트웨어 설계와 소프트웨어 항목 사이의 일관성과 추적성이 수립된다;
- 소프트웨어 단위에 변경이 발생했을 때 소프트웨어 항목을 재검증하기 위한 회귀전략이 개발되고 적용된다.

## 5. 측정 항목

측정 항목(단위)	측정 방법
전체 소프트웨어 항목 수(개)	통합대상 소프트웨어 항목의 개수를 측정한다.
통합된 소프트웨어 항목 수(개)	성공적으로 통합된 소프트웨어 항목의 개수를 측정한다.

(사용 예시)

- 통합 비율(%) = (통합된 소프트웨어 항목 수)/(전체 소프트웨어 항목 수)

## 6. 프로세스 조정

소프트웨어 통합시험 계획서는 별도로 작성하지 않고, 총괄적인 시험계획서에 포함할 수 있다. XP(eXtreme Programming)와 같이 소프트웨어 단위들에 대한 통합이 지속적으로 일어나는 경우 통합을 간편히 할 수 있는 환경을 갖추어 놓는 것이 유리하다.

## 7. 참고문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- IEEE 829-1998, IEEE Standard for Software Test Documentation

## ENG.8 소프트웨어 시험 프로세스

### 1. 개요

#### 1.1. 목적

소프트웨어 시험 프로세스의 목적은 통합된 소프트웨어 제품이 정의된 요구사항을 만족시킴을 확인하는 것이다.

#### 1.2. 설명

소프트웨어 통합 프로세스가 소프트웨어 단위들의 내부 상호작용을 확인하는 과정이라면, 소프트웨어 시험 프로세스에서는 전체 소프트웨어 제품으로 통합되어 릴리스 되기 전에 기능 뿐 아니라 비기능(예. 성능)을 고려하여 소프트웨어 제품에서의 오류 및 결함을 발견하고 시정하기 위한 과정이다.

#### 1.3. 주요 관련 프로세스

ID	프로세스	설명
ENG.7	소프트웨어 통합 프로세스	소프트웨어 통합 프로세스를 통해 통합된 소프트웨어 제품에 대해 소프트웨어 시험 프로세스가 수행된다.
ENG.12	소프트웨어 및 시스템 유지관리	소프트웨어 시험 프로세스를 통해 개발이 완료된 소프트웨어 제품은 사용자에게 이전되고, 소프트웨어 및 시스템 유지관리 프로세스를 통해 운영/유지관리가 이루어진다.

#### 1.4. 용어 정의

- 소프트웨어 시험(test): 소프트웨어를 명시된 환경하에서 실행하여, 그 결과를 관찰/기록하고 평가하는 활동
- 시험 사례(test case): 시험을 위한 입력, 실행 조건, 기대되는 결과 등으로 이루어진 데이터 집합
- 회귀 시험(regression testing): 소프트웨어에 가한 변경 사항이 의도되지 않은 결과를 발생시키지 않고 명시된 요구사항에 부합함을 검증하기 위해 소프트웨어를 선택적으로 재시험하는 것

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	소프트웨어 통합이 성공적으로 수행됨
입력물	[17-11]소프트웨어 요구사항 [11-01]소프트웨어 제품
수행활동	통합된 소프트웨어 제품에 대한 시험을 개발 통합된 소프트웨어 제품을 시험 통합된 소프트웨어를 회귀 시험
출력물	[08-21]소프트웨어 시험계획서 [10-02]시험절차서 [17-14]시험사례명세서 [11-01]소프트웨어 제품 [14-04]시험 로그
완료조건	소프트웨어 시험결과가 승인됨

### 2.2. 책임과 역할

활동	역할		
	시험담당자	구현담당자	프로젝트 관리책임자
1. 통합된 소프트웨어 제품에 대한 시험을 개발	●	○	○
2. 통합된 소프트웨어 제품을 시험	●	○	
3. 통합된 소프트웨어를 회귀 시험	●	○	

(●: 주관, ○: 지원)



### 3. 프로세스 활동

#### 3.1. 통합된 소프트웨어 제품에 대한 시험을 개발

소프트웨어 요구사항이 점검됨을 보여주기 위한 통합된 소프트웨어 제품, 입력 데이터 및 검증 기준에 따라 수행되는 시험을 기술한다. 시험의 집합이 소프트웨어 요구사항에 부합함을 보여 주어야 한다.

☞ 시험 및 시험 데이터는 소프트웨어 요구사항 분석 프로세스 (ENG.4), 소프트웨어 설계 프로세스 (ENG.5) 및 소프트웨어 구축 프로세스에서 (ENG.6) 개발될 수 있다.

릴리스 전략이 있을 경우 그 전략에 따라 소프트웨어 시험전략을 수립한다. 소프트웨어 시험을 위해 필요한 소프트웨어, 하드웨어, 시험 도구 등이 있다면 사전에 갖추어 놓는다.

☐ 산출물: [08-21]소프트웨어 시험계획서, [10-02]시험절차서, [17-14]시험사례명세서

#### 3.2. 통합된 소프트웨어 제품을 시험

검증 기준에 따라 통합된 소프트웨어 제품을 시험하고 그 결과를 기록한다. 필요 시 사용자 문서를 갱신한다.

☞ 통합된 소프트웨어 제품에 대해 시험절차서 및 시험사례 명세서를 적용하여 시험을 수행하고 그 결과를 기록한다. 시험 결과를 분석하여 소프트웨어 요구사항이 충족되었는지를 확인한다. 요구사항이 충족되지 못한 경우 그 원인을 분석하여 결함을 찾아내 시정조치를 취한다.

☐ 산출물: [14-04]시험 로그

#### 3.3. 통합된 소프트웨어를 회귀 시험

통합된 소프트웨어 제품을 재시험하기 위한 소프트웨어 회귀 시험 전략을 개발한다. 소프트웨어 항목에 변경이 발생한 경우 전략에 따라 회귀 시험을 수행한다.

☞ 회귀 시험은 소프트웨어 항목이 변경 또는 확장되었을 경우에도 여전히 정확하

게 작동하는가를 확인하기 위한 과정이다. 효율적, 효과적으로 회귀시험을 하기 위해 이전에 만들어진 시험 사례를 활용하지만, 대부분 모든 시험 사례를 다시 수행하지는 않는다. 변경에 의해 영향을 받는 범위를 식별하여 회귀 시험에서 적용한 시험사례를 선정한다. 자동화된 시험도구를 사용할 경우 이전의 시험 사례를 재수행할 수 있다.

☞ 산출물: [14-04]시험 로그

#### 4. 수행결과

- 통합된 소프트웨어가 소프트웨어 요구사항에 적합함을 보여주는 기준이 개발된다;
- 정의된 기준을 사용하여 통합된 소프트웨어가 검증된다;
- 시험결과가 기록된다;
- 소프트웨어 항목에 변경이 발생한 경우 통합된 소프트웨어를 재시험하기 위한 회귀 전략이 개발되고 적용된다.

#### 5. 측정항목

측정항목(단위)	측정 방법
결함 수(개)	소프트웨어 시험에서 발견된 결함의 수를 측정한다.
해결된 결함 수(개)	소프트웨어 시험에서 발견된 결함 중 해결된 결함의 수를 측정한다.

(사용 예시)

- 결함 해결율(%) = (해결된 결함수)/(결함 수)

#### 6. 프로세스 조정

구축하고자 하는 대상이 소프트웨어 제품만으로 이루어진 경우 이 프로세스는 사용자 승인 과정을 포함할 수도 있다. 이 경우 본 프로세스는 사용자가 원하는 제품이 만들어졌는가를 점검하는 확인 과정의 일부가 된다.

## 7. 참고문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- IEEE 829-1998, IEEE Standard for Software Test Documentation

# ENG.11 소프트웨어 설치

## 1. 개요

### 1.1. 목적

소프트웨어 설치 프로세스의 목적은 목표 환경에 합의된 요구사항을 만족시키는 소프트웨어 제품을 설치하는 것이다.

### 1.2. 설명

본 프로세스는 소프트웨어 설치 과정에 적용된다. 소프트웨어 개발이 완료되면 개발 환경에서 목표 환경으로 소프트웨어를 설치하여야한다. 이때 필요하다면 소프트웨어를 변경해 주어야 하며, 설치 후 점검을 거쳐 고객 승인을 받아야 하는 경우도 있다. 문제가 발생한 경우 그 문제점은 식별되고 보고되어야 한다. 필요하다면 임시방편이 적용되기도 한다.

### 1.3. 주요 관련 프로세스

ID	프로세스명	설명
ENG.12	소프트웨어 및 시스템 유지 관리 프로세스	소프트웨어 설치 후 유지관리 프로세스가 수행된다.

### 1.4. 용어 정의

- 설치(installation): 요구하는 대로 수행됨을 보장하기 위해 소프트웨어 제품이 운영 환경으로 통합되고 시험되는 과정

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	소프트웨어 시험이 완료됨
입력물	[06-03]설치가이드 [11-01]소프트웨어 제품
수행활동	설치 전략을 개발 설치 기준을 수립 적응에 대한 요구사항을 명세화 시스템을 적응시킴 소프트웨어 제품을 설치 제품 준비도를 확인
출력물	[06-03]설치가이드 [08-09]설치 및 유지보수 계획 [10-02]시험절차서 [13-01]인수 기록 [13-08]설치 기록 [13-07]문제 기록
완료조건	요구사항 관리 계획 및 검토결과가 승인된다.

### 2.2. 책임과 역할

활동	역할			
	설치담당자	구현담당자	프로젝트 관리책임자	고객
1. 설치 전략을 개발	○		●	○
2. 설치 기준을 수립	●			○
3. 적응에 대한 요구사항을 명세화	●	○		○
4. 시스템을 적응시킴	●	○		
5. 소프트웨어 제품을 설치	●	○	○	○
6. 제품 준비도를 확인	○		●	○

(●: 주관, ○: 지원)

### 3. 프로세스 활동

#### 3.1. 설치 전략을 개발

고객 및 운영 조직과 합의한 대로 목표 환경에 소프트웨어 제품을 설치하는 소프트웨어 설치 전략을 수립한다.

☞ 설치 전략을 개발하는 중요한 부분은 마지막으로 작동하는 시스템 버전으로 되돌아갈 수 있는 전략을 개발하는 것이다. 마지막으로 작동하는 버전으로 재설치를 할 수 있기 위해서는 설치를 시작하기 전에 시스템의 완벽한 백업이 이루어져야 한다.

☐ 산출물: [08-09]설치 및 유지보수 계획

#### 3.2. 설치 기준을 수립

설치 요구사항을 기반으로 소프트웨어가 설치될 환경에 대한 기준이 개발된다.

☞ 설치 요구사항에 따라 운영 장소를 준비한다.

☐ 산출물: [13-01]인수 기록

#### 3.3. 적용에 대한 요구사항을 명세화

의도된 환경에 대한 시스템의 적용을 위한 요구사항을 명세화한다.

☞ 설치가이드를 참조하여 설치를 위한 요구사항을 시험절차서에 명세화한다.

☐ 산출물: [10-02]시험절차서

#### 3.4. 시스템을 적용시킴

운영 요구사항을 만족하도록 시스템을 적용시킨다.

☞ 설치가이드에 따라 소프트웨어가 설치될 수 있도록 목표 시스템 환경을 조정한다.

☐ 산출물: 해당사항 없음

### 3.5. 소프트웨어 제품을 설치

소프트웨어 설치 전략에 따라 소프트웨어 제품을 설치한다. 사건과 결과를 문서화한다.

☐ 산출물: [13-08]설치 기록

### 3.6. 제품 준비도를 확인

소프트웨어 제품이 의도된 환경에서 사용될 수 있도록 준비됨을 보장한다.

☞ 설치 요구사항에 적합한지 설치 시험을 수행하고, 문제가 발생한 경우 그것을 기록으로 남긴다. 문제점은 해결한 후 재설치하고 확인하는 과정을 반복한다.

소프트웨어 설치가 성공적으로 이루어지면 고객에 의한 수락이 이루어질 수 있다.

☐ 산출물: [13-07]문제 기록

## 4. 수행결과

- 소프트웨어 설치 전략이 개발된다.
- 소프트웨어 설치 요구사항을 준수함을 보여주는 소프트웨어 설치 기준이 개발된다.
- 소프트웨어 제품이 목표 환경에 설치된다.
- 소프트웨어 제품이 의도된 환경에서 사용할 준비가 되었음이 보장된다.

## 5. 측정항목

측정항목(단위)	측정 방법
설치 시도 횟수	목표 환경에서 소프트웨어 설치를 시도한 횟수를 측정한다.

(사용 예시)

- 설치 시도 횟수

## 6. 프로세스 조정

소프트웨어 설치에 개발자가 직접 하지 않을 수도 있다.

## 7. 참고문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006



# ENG.12 소프트웨어 및 시스템 유지관리

## 1. 개요

### 1.1. 목적

소프트웨어 및 시스템 유지관리 프로세스의 목적은 결함의 수정, 성능 또는 다른 속성의 개선 및 변화된 환경에 적응하기 위해, 인도 후 시스템/소프트웨어 제품을 수정하는 것이다.

### 1.2. 설명

본 프로세스는 소프트웨어/시스템 유지관리 과정에 걸쳐 적용된다.

소프트웨어 및 시스템 유지관리는 소프트웨어/시스템 개발이 종료되고, 현업에 전달되어 실제 활용되는 기간에 수행된다. 개발 기간 동안에 미처 발견하지 못한 오류에 대한 처리, 외부 환경 변화에 따른 대응, 보다 소프트웨어/시스템을 완벽하게 하려는 예방 차원의 조치가 이루어진다. 이러한 과정은 소프트웨어/시스템이 폐기될 때까지 계속된다.

### 1.3. 주요 관련 프로세스

ID	프로세스명	설명
ENG.11	소프트웨어 설치 프로세스	소프트웨어 설치 후 유지관리 프로세스가 수행된다.
SUP.2	형상관리 프로세스	새로운 시스템을 설치하거나 업그레이드 할 경우 형상관리 프로세스를 적용한다.
SUP.8	문제점관리 프로세스	문제점을 처리할 경우 이 프로세스를 적용한다.

### 1.4. 용어 정의

- 유지관리(maintenance): 소프트웨어/시스템 설치 후 결함 시정, 성능 개선, 새로운 환경 적응을 위해 소프트웨어/시스템을 수정하는 것

- 시정 유지관리(corrective maintenance): 발견된 결함을 시정하기 위해 소프트웨어/시스템에 대해 수행되는 수동적 수정사항
- 적응 유지관리(adaptive maintenance): 변경되는 환경에서 사용할 수 있도록 소프트웨어/시스템을 변경하는 것
- 완전 유지관리(perfective maintenance): 성능을 개선하거나 유지관리성을 개선하기 위해 소프트웨어/시스템을 변경하는 것

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	소프트웨어 설치가 완료됨
입력물	[12-02]폐기 요청서 [13-07]문제 기록 [13-16]변경요청서 [13-17]고객요청서
수행활동	유지관리 전략을 개발 사용자의 문제점 및 변경요청을 분석 변경사항을 구현하고 시험 사용자 시스템을 갱신 소프트웨어 제품을 폐기 변경사항을 전달
출력물	[08-09]설치 및 유지관리 계획 [13-27]폐기 공지 [14-01]변경 이력 [13-04]의사소통 기록 [19-06]유지관리전략
완료조건	소프트웨어가 폐기됨

## 2.2. 책임과 역할

역할 \ 활동	유지관리 담당자	프로젝트 관리책임자	고객
1. 유지관리 전략을 개발	○	●	○
2. 사용자의 문제점 및 변경요청을 분석	●		○
3. 변경사항을 구현하고 시험	●		○
4. 사용자 시스템을 갱신	●		○
5. 소프트웨어 제품을 폐기	●		○
6. 변경사항을 전달	●		○

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 유지관리 전략을 개발

유지관리 요구사항, 릴리스 전략 및 가능한 보증 정책과 일치하게 제품의 변경, 이전 및 폐기를 관리하기 위한 전략을 개발한다.

☞ 유지관리 전략은 시정 및 예방 유지보수를 수행하는데 필요한 일정 및 인적/물적 자원을 정의한다. 유지관리의 범위(대상, 지원 유형), 유지관리 절차, 유지관리 담당자, 유지보수 비용 등을 포함된다.

📄 산출물: [19-06]유지관리전략, [08-09]설치 및 유지보수 계획

### 3.2. 사용자의 문제점 및 변경요청을 분석

기존의 시스템 및 소프트웨어, 시스템 인터페이스 및 요구사항을 변경하기 위한 대안들의 가능한 영향을 평가하여, 고객의 문제, 요청 및 요구되는 변경을 분석한다. 선택된 해결방안을 문서화한다.

☞ 고객의 문제, 변경요청을 접수하고, 그 내용을 분류한다. 분류는 시정내용인지, 적용내용인지, 보완내용인지, 긴급 상황인지 등으로 분류한다. 분석을 통해 변경에 따른 영향을 평가하고 해결책을 찾아 기록하여, 담당자에게 그 내용을 전달하여 해결하도록 한다.

이 기본 프랙티스는 문제 해결 관리 프로세스와 (SUP.9) 연계된다.

☐ 산출물: [13-16]변경요청서

### 3.3. 변경사항을 구현하고 시험

어떤 제품이 변경되어야 하는지를 결정한다. 변경을 통하여 시스템 및 소프트웨어 요구사항과 무결성이 손상되지 않았음을 보여줄 수 있도록 선택된 변경사항이 구현, 시험 및 문서화된다.

☞ 변경사항은 구현 후 단위시험을 거치고, 전체 시스템과 통합하면서 통합시험 및 회귀시험을 실시한다. 변경에 따른 모든 영향을 평가하고, 보완할 사항이 있으면 이를 해결하기 위해 앞의 과정을 반복한다.

☐ 산출물: [14-01]변경 이력

### 3.4. 사용자 시스템을 업그레이드

적용된 변경을 포함한 업그레이드된 시스템 및 소프트웨어를 사용자 환경에 이전한다. 필요시 이전 계획 및 활동에 대한 통지, 구 및 신 시스템의 병행 운영 및 사용자 훈련을 제공한다. 변경의 영향을 평가하기 위하여 운영 후 검토를 수행한다.

☞ 시스템에 대한 올바른 형상이 전달되도록 사전 점검을 실시하고, 사용자에게 그 사항을 알린다. 전달시 기존 시스템을 새로운 버전으로 교체하고, 백업을 위해 시스템의 이전 버전을 관리한다. 설치를 수행하고, 필요시 교육훈련을 실시한다.

☐ 산출물: 해당사항 없음

### 3.5. 소프트웨어 제품을 폐기

승인에 따라 사용자 환경에서 낙후된 시스템을 폐기한다. 적절하게 폐기 계획 및 활동을 공지하고, 전환 시스템과의 병행 운영하고, 새로운 또는 변경된 시스템으

로의 데이터 변환하고, 시스템 및 데이터 파일의 저장 및 사용자 훈련을 제공한다.

☞ 폐기 계획을 수립하고 관련 사항은 사용자에게 통보한다. 통보내용에는 다음 사항을 포함한다.

- a) 이용가능한 날짜와 함께 교체 또는 개선에 대한 기술
- b) 왜 소프트웨어 제품이 더이상 지원될 수 없는지에 대한 설명
- c) 지원이 없어진 후, 이용 가능한 다른 지원 대안에 대한 기술

필요시 새로운 시스템으로의 원활한 전환을 위하여 폐기될 소프트웨어 제품과 신규 소프트웨어 제품이 병행 운영되도록 한다. 이 기간 중 사용자 교육훈련을 마련한다.

예정된 폐기 일정이 도래하면, 공지를 통해 모든 관련자에게 통보한다. 모든 관련된 개발문서, 로그 및 코드는 기록보관소에 보관되도록 한다. 필요시 폐기된 소프트웨어 제품에 의해 사용되었거나 관련된 데이터는 데이터 보호 및 감사를 위해 접근이 가능해야 한다.

☐ 산출물: [12-02]폐기 요청서, [13-27]폐기 공지

### 3.6. 변경사항을 전달

영향을 받는 모든 당사자들에게 시스템 및 소프트웨어 변경의 배포를 위한 의사소통 절차와 수단을 수립한다.

☞ 분석보고서, 문제점보고서, 시정조치 및 추이를 유지하여 관련된 사람들에게 전달한다.

☐ 산출물: [13-04]의사소통 기록

## 4. 수행결과

- 릴리스 전략에 따라 제품의 수정, 이전, 폐기를 관리하기 위한 유지관리 전략이 개발된다.
- 기존시스템 변경이 조직, 운영 또는 인터페이스에 미치는 영향이 식별된다.
- 영향을 받는 시스템/소프트웨어 문서가 필요 시 갱신된다.
- 요구사항이 손상되지 않았음을 보여주는 관련 시험들과 함께 수정된 제품이

개발된다.

- 고객의 환경에 갱신된 제품이 이전된다.
- 요청에 따라 고객에게 불편을 최소화하는 통제된 방법으로 제품이 폐기된다.
- 시스템/소프트웨어 변경이 영향을 받는 모든 당사자들에게 의사소통된다.

## 5. 측정 항목

측정 항목(단위)	측정 방법
변경요청 건수(건)	총 변경요청 건수를 측정한다.
변경요청처리 시간(시간)	변경요청을 처리하는데 소요된 시간을 측정한다.

(사용 예시)

- 평균 변경 처리 시간(시간/건) = 변경요청처리 시간/변경요청 건수

## 6. 프로세스 조정

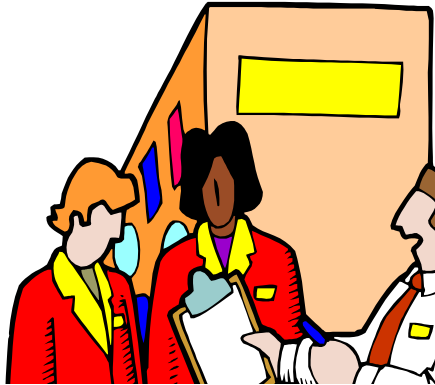
소프트웨어에 따라 유지관리 방법이 달리 적용될 수 있다. 개발 후 일정기간 동안 개발자가 유지보수를 지원해 줄 수 있으며, 그 후에는 고객 측에서 유지보수 업무를 수행할 수 있다.

## 7. 참고문헌

- IEEE Std 1219-1998, IEEE Standard for Software Maintenance

## 제 6 장 관리 프로세스 그룹(MAN)

- MAN.3-1 프로젝트 계획 수립 프로세스
- MAN.3-2 프로젝트 진행 관리 프로세스
- MAN.5 위험 관리 프로세스



# MAN.3-1 프로젝트 계획 수립 프로세스

## 1. 개요

### 1.1. 목적

프로젝트 계획 수립 프로세스의 목적은 프로젝트의 요구사항 및 제약사항 등을 기초로 제품 및/또는 서비스를 개발하는 프로젝트에 필요한 활동, 작업(task) 및 자원을 식별하여 이에 대한 계획을 수립하고 유지하는 것이다.

### 1.2. 설명

본 프로세스는 프로젝트의 착수 단계부터 완료단계까지의 프로젝트 생명주기 전 과정에 걸쳐 적용된다.

프로젝트 계획 수립 프로세스는 다음과 같은 활동들을 포함한다.

- 프로젝트 계획서 작성
- 관련 이해관계자와의 적절한 상호 활동 수행
- 프로젝트 계획서에 대한 합의 확보
- 프로젝트 계획서 유지

프로젝트 계획 수립은 소프트웨어 요구사항을 기초로 작업산출물과 활동을 견적하여 필요한 자원을 결정하고, 합의해야 할 사항을 협상하며, 일정 정의 및 프로젝트의 위험을 식별하고 분석하는 활동들을 포함한다. 프로젝트 계획서를 작성하기 위해서는 이러한 활동들이 반복적으로 수행될 수도 있다. 프로젝트 계획서는 고객과 합의한 요구사항 및 공약(commitment)을 처리하기 위한 활동을 수행하고 통제하는 근간이 된다. 또한 프로젝트가 진행되어 감에 따라 요구사항 및 공약의 변경, 부정확한 견적, 시정조치, 프로세스의 변경 등을 처리하기 위해 프로젝트 계획서를 개정해야 할 필요가 생기게 되는데, 본 프로세스에서는 이러한 계획 및 재계획과 관련된 사항들을 모두 포함하고 있다.



### 1.3. 주요 관련 프로세스

ID	프로세스	설명
MAN.3-2	프로젝트 진행 관리 프로세스	프로젝트 계획은 프로젝트 진행 상태를 결정하고 모니터링하기 위한 근간으로 활용되며, 프로젝트 진행 중에 발생하는 문제점에 대한 시정조치를 위해 재계획이 필요한 경우 프로젝트 계획 수립 프로세스에 따라 처리한다.
MAN.5	위험 관리 프로세스	프로젝트 계획은 프로젝트 위험을 식별하기 위한 기초자료로 활용되며, 식별된 위험을 제거하거나 완화하기 위하여 프로젝트 계획을 조정할 수도 있다.

### 1.4. 용어 정의

- 마일스톤 (Milestone): 프로젝트 진행 상황을 측정하는데 사용되는 일정상의 사건(event). 프로젝트에 관한 주요 마일스톤의 예시로는 획득자 혹은 관리자의 승인(sign-off), 명세서에 관한 기준선화, 시스템 통합의 완료, 그리고 제품 인도 등이 있으며, 기타 마일스톤으로는 소프트웨어 모듈의 기준선화 또는 사용자 매뉴얼의 한 단원의 완료 등이 있음
- 위험 (Risk): 프로젝트 일정, 예산, 작업산출물 등에 부정적인 영향을 미쳐 정해진 기간 내에 프로젝트가 종료되지 못하거나, 고객 요구사항을 만족시키지 못할 가능성이 있는 잠재적 문제점
- 이해관계자 (Stakeholders): 프로젝트에 직접 또는 간접적으로 관련이 있는 그룹 또는 개인
- 작업 분해 구조도 (Work Breakdown Structure; WBS): 프로젝트 수행과 관련된 모든 작업 항목을 계층적 구조로 세분화하여 일목요연하게 표현한 것을 말하며, 작업 분해 구조에 정의된 최하위 작업 항목은 프로젝트 진행 관리를 위해 추적되는 관리 단위임
- 하는 작업 명세. 작업 패키지는 일반적으로 고유한 이름 및 식별자, 작업 초기화를 위한 선행 조건, 인력 요구사항, 기타 필요 자원, 생성해야 할 작업 산출물, 예상 기간, 위험 요소, 선행 및 후속 과업, 작업에 특별히 고려할 사항, 그리고 생성해야 할 작업 산출물의 품질 기준을 포함한 작업 패키지의 완료 기준을 포함함
- 프로젝트 생명주기 (Project Life Cycle): 소프트웨어 개발 및 관리를 위한 기본적인 틀(Framework)을 제공하는 것으로서, 개발 방법론과 밀접한 관계가 있으며, 추진하고자 하는 프로젝트의 특성 및 위험 요인에 따라서 프로젝트 생명주기를 선정함. 프로젝트 생명주기는 일반적으로 요구사항 분석단계, 설계단계, 구현단계, 시험단계, 운영 및 유지보수단계, 그리고 때때로 폐기 단계를 포함하여 구성됨

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트 소요가 제기되었다. 프로젝트 관리자가 선임되었다.
입력물	[12-01]제안 요청서 [02-00]계약(서) [17-03]고객 요구사항 [10-01]생명주기 모델 [19-07]소프트웨어 개발 방법론
수행활동	프로젝트 범위 정의 프로젝트 생명주기 정의 프로젝트 수행 가능성 평가 프로젝트 속성 견적 및 유지 프로젝트 활동 및 작업 정의 경험, 지식 및 스킬 요구사항 정의 프로젝트 자원 할당 및 추진일정 정의 프로젝트 인터페이스 식별 책임 할당 프로젝트 계획 수립
출력물	[08-12]프로젝트 계획서 [08-06]프로젝트 활동도 [14-09]작업 분해 구조도 [14-06]일정
완료조건	프로젝트 팀이 구성되고 자원이 할당되었다. 프로젝트 계획이 수립되었다.

## 2.2. 책임과 역할

역할 \ 활동	프로젝트 관리책임자	개발자	이해관계자
1. 프로젝트 범위 정의	●	○	
2. 프로젝트 생명주기 정의	●	○	
3. 프로젝트 수행 가능성 평가	●	○	
4. 프로젝트 속성 견적 및 유지	●	○	○
5. 프로젝트 활동 및 작업 정의	●	○	
6. 경험, 지식 및 스킬 요구사항 정의	●	○	
7. 프로젝트 자원 할당 및 추진일정 정의	●	○	
8. 프로젝트 인터페이스 식별	●	○	○
9. 책임 할당	●	○	○
10. 프로젝트 계획 수립	●	○	○

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 프로젝트 범위 정의

프로젝트에서 수행하여야 할 업무 범위를 정의한다.

☞ 프로젝트 업무 범위는 제안 요청서, 계약(서), 고객 요구사항 등을 기반으로 개발 대상 제품이나 기능 구조를 식별하고, 이를 기초로 프로젝트 작업, 책임, 일정 등을 구체적으로 견적하기에 충분한 정도의 작업 패키지(Work Package)를 식별한다, 또한, 조직 외부로부터 획득하게 될 작업산출물(또는 작업산출물의 컴포넌트)이나 재사용될 작업산출물 등도 포함한다.

☐ 산출물: [08-12]프로젝트 계획서

### 3.2. 프로젝트 생명주기 정의

프로젝트의 업무 범위, 상황, 규모, 복잡도 등을 고려하여 적절한 프로젝트 생명주기와 개발 전략을 정의한다.

☞ 프로젝트 생명주기는 요구사항 분석, 설계, 구축, 통합 및 시험 등과 같은 각각의 개발 단계들로 구성되며, 개발 전략이 무엇인가에 따라서 아래에 기술한 예시 등과 같은 생명주기 모델을 고려하여 정의된다.

- 폭포수 모델 (Waterfall Model)
- 프로토타이핑 모델 (Prototyping Model)
- 점증적 개발 모델 (Incremental Development Model)
- 점진적 개발 모델 (Evolutionary Development Model)
- 나선형 모델 (Spiral Model)
- RAD 모델 (Rapid Application Development Model)
- 4세대 모델 (4th Generation Technique)

☐ 산출물: [08-12]프로젝트 계획서

### 3.3. 프로젝트 수행 가능성 평가

가용한 자원 및 제약사항 등을 고려하여 프로젝트에 부여된 목표에 대한 달성가능성을 평가한다.

☞ 프로젝트 수행 가능성 평가를 위해서는 프로젝트와 관련된 이해관계자들이 모두 참석하는 것이 바람직하며, 평가 결과로서 나타난 프로젝트 목표 달성을 위한 전제사항 및 제약사항 등은 프로젝트 계획 수립시 고려한다.

☐ 산출물: [08-12]프로젝트 계획서

### 3.4. 프로젝트 속성 견적 및 유지

프로젝트의 목표 및 위험 등을 고려하여 프로젝트 속성에 대한 견적을 수행하고, 유지한다.

☞ 프로젝트 속성은 일반적으로 소프트웨어 규모, 투입공수, 일정, 비용 등을 포함하며, 기타 필요한 사항이 포함될 수도 있다.

소프트웨어 규모 측정을 위한 대상은 아래에 기술한 예시 등과 같다.

- 인도 및 비인도 작업산출물
- 문서
- 운영 및 지원 소프트웨어

소프트웨어 규모 측정을 위한 방법은 아래에 기술한 예시 등과 같다.

- 기능 수 (Number of Functions)
- 기능 점수 (Function Points; FP)
- 코드 라인 수 (Source Lines of Code; LOC)
- 클래스 및 객체 수 (Number of Classes and Objects)
- 요구사항 수 (Number of Requirements)
- 인터페이스 수 (Number of Interfaces)
- 페이지 수 (Number of Pages)
- 입출력 수 (Number of Inputs and Outputs)
- 데이터 양 (Volume of Data)

☐ 산출물: [08-12]프로젝트 계획서

### 3.5. 프로젝트 활동 및 작업 정의

정의된 프로젝트 생명주기를 기초로 프로젝트 활동 및 작업을 식별하고, 작업들 간의 의존성을 정의한다.

☞ 프로젝트 활동 및 작업 정의는, 요구되는 경우 소프트웨어 개발 방법론을 고려하여, 프로젝트 활동도와 프로젝트 범위 정의 시 작성된 상위 수준의 작업 분해 구조도를 좀 더 구체적으로 세분화하여 적절한 수준의 최하위 작업항목을 정의하는 형태로 수행된다. 이때 적절한 수준이란 일반적으로 최하위 작업항목의 특성을 고려하여 4 ~ 40시간(1주) 내에 완료할 수 있을 정도의 작업량이 될 수 있도록 분해하는 것이 바람직하다. 하지만 작업항목의 특성에 따라 예외적인 사항이 생길 수도 있다.

프로젝트 활동도와 작업 분해 구조도를 작성하기 위한 일반적인 접근방법은 아래에 기술한 예시 등과 같다.

- 시스템 구성요소를 기준으로 전개하는 방법
- 개발 단계를 기준으로 전개하는 방법

- 개발 프로세스를 기준으로 전개하는 방법

☐ 산출물: [08-06]프로젝트 활동도, [14-09]작업 분해 구조도

### 3.6. 경험, 지식 및 스킬 요구사항 정의

프로젝트 수행에 필요한 경험, 지식 및 스킬(skill) 요구사항을 식별하고, 프로젝트 구성원 선정 및 업무 배정 시 활용한다.

☞ 프로젝트 수행에 필요한 경험, 지식 및 스킬 요구사항은 프로젝트 관리, 형상 관리, 품질 보증, 위험 관리, 외주업체 관리 등과 같은 관리적 요구사항과 분석, 설계, 구현, 시험 등과 같은 기술적 요구사항 모두를 포함한다.

☐ 산출물: [08-12]프로젝트 계획서

### 3.7. 프로젝트 자원 할당 및 추진일정 정의

프로젝트 투입공수 견적 결과와 식별된 경험, 지식 및 스킬 요구사항을 고려하여 프로젝트 활동 및 작업에 자원을 할당하고, 활동 및 작업들간의 의존성 등을 고려하여 수행 순서 및 추진일정을 정의한다.

☞ 프로젝트 자원 할당은 프로젝트 목적 달성을 위해 적절한 인원의 선정 및 배정 뿐만 아니라, 프로젝트에 부과된 업무를 수행하기 위해 필요한 모든 설비, 장비, 도구 (예를 들면, 개인용 컴퓨터, 분석 및 설계용 소프트웨어, 컴파일러 등과 같은 개발 환경 등) 등의 자산 모두를 포함한다.

프로젝트 추진일정은 먼저 주요 마일스톤(milestone)을 식별하고, 일정과 관련된 가정 및 제한사항, 작업들간의 의존성을 고려하여 프로젝트의 기간을 최소화 할 수 있는 최적의 순서를 결정하여 정의한다. 이러한 활동과 관련하여 일반적으로 활용하는 기법은 아래에 기술한 예시 등과 같다.

- 간트 차트 (Gantt Chart)
- 핵심 경로 기법 (Critical Path Method; CPM)
- 프로그램 평가 및 검토 기법 (Program Evaluation and Review Technique; PERT)
- 획득가치(기성고) 관리 기법 (Earned Value Management; EVM)

☐ 산출물: [14-09]작업 분해 구조도, [14-06]일정

### 3.8. 프로젝트 인터페이스 식별

프로젝트와 관련된 타 프로젝트, 타 조직 및 기타 영향 받는 이해관계자와의 인터페이스를 식별하고, 합의한다.

☞ 프로젝트 인터페이스란 프로젝트 생명주기의 모든 단계에서 발생하는, 프로젝트와 관련 이해관계자들간의 상호 연관된 관리적 그리고 기술적인 사항 모두를 의미한다.

☐ 산출물: [08-12]프로젝트 계획서

### 3.9. 책임 할당

프로젝트에 기여하거나 프로젝트에 의해 영향 받는 특정 조직을 식별하여 책임을 할당하고, 상호 검토 및 이해하고, 합의한다.

☞ 식별된 프로젝트 인터페이스를 처리하기 위해 이해관계자들을 식별하고 책임을 할당하는 데에 유용하게 활용하는 방법으로서 한 축에는 이해관계자들을 나열하고, 또 다른 한 축에는 프로젝트 인터페이스를 표현하는 2차원 매트릭스 형태가 있으며, 프로젝트와 이해관계자들과의 상호 활동(책임)은 이해관계자 축과 프로젝트 인터페이스 축이 교차하는 지점에 나타나게 된다.

☐ 산출물: [08-12]프로젝트 계획서

### 3.10. 프로젝트 계획 수립

프로젝트 범위 및 목표, 자원, 기반구조, 인터페이스 및 의사소통 체계 등을 포함하는 프로젝트 계획 및 기타 관련 계획을 수립하고, 유지한다.

☞ 프로젝트 계획은 프로젝트를 수행하기 위해 필요한 모든 활동들이 기술된 프로젝트 계획서 등과 같이 하나의 통합된 계획서 형태로 수립될 수도 있고, 필요에 따라 형상관리 계획서, 위험관리 계획서, 품질보증 계획서 등과 같이 여러 개의 문서로 나누어서 작성 될 수도 있다.

☞ 프로젝트에 영향을 주는 모든 계획서는, 또한 프로젝트에 영향을 주는 모든 이해관계자와 함께 검토하여, 프로젝트가 성공하기 위해 요구되는 프로젝트의 범위, 목적, 역할, 관계 등에 대한 상호 공통의 합의가 이루어져야 한다.

☐ 산출물: [08-12]프로젝트 계획서

#### 4. 수행 결과

- 프로젝트의 업무 범위가 정의된다.
- 가용한 자원 및 제약사항을 고려하여 프로젝트 목표 달성 가능성이 평가된다.
- 업무를 완수하는데 필요한 작업 및 자원이 견적된다.
- 프로젝트 구성 요소간, 타 프로젝트 및 조직과의 인터페이스가 식별된다.
- 프로젝트 추진을 위한 계획서가 수립된다.

#### 5. 측정 항목

측정항목(단위)	측정 방법
계획 소프트웨어 규모	계획된 소프트웨어 규모(예를 들면, 코드 라인 수(Line of Code; LOC), 기능 점수(Function Point; FP) 등)를 측정한다.
계획 투입공수	계획된 인원에 대한 투입공수(예를 들면, Man/Month, Man/Day 등)를 측정한다.
계획 예산	계획된 프로젝트 예산을 측정한다.
계획 일정	계획된 프로젝트 추진일정을 측정한다.
계획 작업량	정의된 측정 주기까지의 계획 작업량(예를 들면, 투입공수 등)을 측정한다.
전체 작업량	계획된 프로젝트 전체 작업량(예를 들면, 투입공수 등)을 측정한다.

(사용 예시)

- 계획 진척도(%): (측정 주기까지의 계획 작업량 / 프로젝트 전체 작업량) X 100

#### 6. 프로세스 조정

- "프로젝트 범위"는 일반적으로 "[08-12]프로젝트 계획서"에 정의된다. 그러나, "[08-12]프로젝트 계획서"뿐만 아니라 상위 수준의 "[14-09]작업 분해 구조도" 형태로도 정의할 수도 있으며, 프로젝트가 진행되면서 점점 더 구체화되어 하나의 완성된 "[14-09]작업 분해 구조도"가 된다.
- "프로젝트 속성 견적 및 유지" 결과는 "[08-12]프로젝트 계획서"에 요약된 정보만 정의하며, 구체적인 사항은 프로젝트 견적 결과서를 작성하여 유지하는 것이 일반적이다.
- "[08-06]프로젝트 활동도"는 "[14-09]작업 분해 구조도"에 포함하여 작성하는 것이 일반적이다. 따라서, 이 경우에는 "[08-06]프로젝트 활동도" 작성은 생략할 수 있다.



- 프로젝트 "[14-06]일정"도 최근에는 "[14-09]작업 분해 구조도"에 포함하여 작성하는 것이 일반적이다. 따라서, 이 경우에는 "[14-06]일정"의 작성은 생략할 수 있다.

## 7. 참고 문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- CMMI-DEV (Capability Maturity Model Integration for Development), Version 1.2
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

## MAN.3-2 프로젝트 진행 관리 프로세스

### 1. 개요

#### 1.1. 목적

프로젝트 진행 관리 프로세스의 목적은 프로젝트의 계획과 비교하여 진행 상태를 모니터링하고, 심각한 편차가 발생한 경우 적절한 시정조치를 취할 수 있도록 하는 것이다.

#### 1.2. 설명

본 프로세스는 프로젝트의 착수단계부터 완료단계까지의 프로젝트 생명주기 전 과정에 걸쳐 적용된다.

프로젝트 진행 관리는 프로젝트 계획서를 근간으로 정해진 마일스톤이나 일정 또는 작업 분해 구조도 내의 통제 시점에서 작업, 투입공수, 비용, 일정 등을 실제 작업산출물과 비교하여 진행 상태를 결정하고, 계획에 비해 실제 성과가 심각한 편차를 보이는 경우 적절한 시정조치를 취한다. 이러한 시정조치는 현재의 계획을 개정하거나, 새로운 합의를 확보하는 등의 재계획 활동 또는 현재의 계획 내에서 추가적인 완화조치를 추가하는 등의 활동으로 이어지게 된다.

#### 1.3. 주요 관련 프로세스

ID	프로세스	설명
MAN.3-1	프로젝트 계획 수립 프로세스	프로젝트 계획은 프로젝트 진행 상태를 결정하고 모니터링하기 위한 근간으로 활용되며, 프로젝트 진행 중에 발생하는 문제점에 대한 시정조치를 위해 재계획이 필요한 경우 프로젝트 계획 수립 프로세스에 따라 처리한다.
MAN.5	위험 관리 프로세스	프로젝트 진행 중에 식별되는 위험은 위험 관리 프로세스에 따라 처리되며, 위험 관리 프로세스에서 식별된 위험 또한 프로젝트 진행 관리 프로세스에서 모니터링 된다.
SUP.9	문제 해결 관리 프로세스	프로젝트 진행 중에 식별되는 문제를 해결하기 하기 위해 문제 해결 관리 프로세스가 수행될 수도 있다.
SUP.10	변경 요청 관리 프로세스	프로젝트 진행 중에 식별되는 문제를 해결하기 하기 위해 요구사항을 조정하거나, 기타 변경에 대한 요청이 이루어질 수도 있다.

#### 1.4. 용어 정의

- 마일스톤 (Milestone): 프로젝트 진행 상황을 측정하는데 사용되는 일정상의 사건(event). 프로젝트에 관한 주요 마일스톤의 예시로는 획득자 혹은 관리자의 승인(sign-off), 명세서에 관한 기준선화, 시스템 통합의 완료, 그리고 제품 인도 등이 있으며, 기타 마일스톤으로는 소프트웨어 모듈의 기준선화 또는 사용자 매뉴얼의 한 단원의 완료 등이 있음
- 심각한 편차: 해결되지 않은 채 방치할 경우, 프로젝트의 목적을 만족시키지 못하는 편차
- 이해관계자 (Stakeholders): 프로젝트에 직접 또는 간접적으로 관련이 있는 그룹 또는 개인
- 위험 (Risk): 프로젝트 일정, 예산, 작업산출물 등에 부정적인 영향을 미쳐 정해진 기간 내에 프로젝트가 종료되지 못하거나, 고객 요구사항을 만족시키지 못할 가능성이 있는 잠재적 문제점
- 작업 분해 구조도 (Work Breakdown Structure; WBS): 프로젝트 수행과 관련된 모든 작업 항목을 계층적 구조로 세분화하여 일목요연하게 표현한 것을 말하며, 작업 분해 구조에 정의된 최하위 작업 항목은 프로젝트 진행 관리를 위해 추적되는 관리 단위임

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트가 착수되었다. 프로젝트 관리자가 선임되었다. 프로젝트 팀이 구성되고 자원이 할당되었다.
입력물	[08-12]프로젝트 계획서 [08-06]프로젝트 활동도 [14-09]작업 분해 구조도 [14-06]일정 [08-19]위험관리 계획서
수행활동	프로젝트 수행 및 업무 보고 프로젝트 속성 모니터링 프로젝트 진척도 검토 시정조치 활동 수행 프로젝트 종료 검토 수행
출력물	[13-14]진척도 상태 기록 [15-06]프로젝트 상태 보고서 [14-02]시정조치 관리대장 [13-16]변경요청서 [13-19]검토 기록
완료조건	프로젝트가 종료되었다. 프로젝트 종료 검토 활동이 수행되었다.

### 2.2. 책임과 역할

활동 \ 역할	프로젝트 관리책임자	개발자	이해관계자
1. 프로젝트 수행 및 업무 보고	○	●	○
2. 프로젝트 속성 모니터링	●	○	
3. 프로젝트 진척도 검토	●	○	○
4. 시정조치 활동 수행	●	○	
5. 프로젝트 종료 검토 수행	●	○	○

(●: 주관, ○: 지원)

### 3. 프로세스 활동

#### 3.1. 프로젝트 수행 및 업무 보고

프로젝트의 계획된 활동 및 작업을 수행하고, 주기적으로 진행 상태를 기록하고, 관련 이해관계자에게 보고한다.

☞ 보고되는 진행 상태 정보에는 작업 분해 구조도에 정의된 작업항목의 진척도 뿐만 아니라 당면 문제점 및 이슈, 위험, 지연된 업무에 대한 만회 대책 등이 포함된다.

☐ 산출물: [13-14]진척도 상태 기록

#### 3.2. 프로젝트 속성 모니터링

프로젝트 범위, 투입공수, 예산, 자원, 인터페이스 및 기타 필요한 속성을 실적과 비교하여 모니터링하고, 프로젝트 계획과 심각한 편차를 문서화한다.

☞ 프로젝트 속성은 일반적으로 작업산출물 및 작업, 비용, 투입공수, 일정 등 프로젝트 진척도 및 성과를 나타내는 계기판이다. 작업산출물 및 작업 속성은 규모, 복잡성, 가중치, 기능 등의 항목이 포함된다.

☞ 모니터링 활동은 일반적으로 프로젝트 속성의 실제 결과값을 측정하는 활동, 실제 결과값을 계획서의 견적 값과 비교하는 활동, 심각한 편차를 식별하는 활동 등을 포함한다.

☐ 산출물: [15-16]프로젝트 상태 보고서

#### 3.3. 프로젝트 진척도 검토

프로젝트 계획과 비교하여 프로젝트 진행 상태를 주기적으로 보고하고, 검토한다.

☞ 프로젝트 진척도 검토는 관련 이해관계자들이 프로젝트의 상태를 이해하도록 하는 활동이다. 이러한 활동은 공식 또는 비공식적인 형태의 검토 활동이 될 수 있으며, 일반적으로 주간 단위, 월간 단위, 그리고 정해진 마일스톤에서 수행된다.

☞ 프로젝트 진척도 검토 활동에는 그 성격에 따라 프로젝트 구성원뿐만 아니라 지원 그룹, 관련 프로젝트 구성원, 그리고 상위관리자(senior management) 등이 참여한다.

☐ 산출물: [13-19]검토 기록

### 3.4. 시정조치 활동 수행

프로젝트 목표가 달성되지 않았을 경우, 계획 대비 차이점을 시정하기 위해, 그리고 식별된 편차에 대해 재발을 방지하기 위하여 시정조치 활동을 수행하고, 필요한 경우 프로젝트 계획을 변경한다.

시정조치 활동을 결정하기 위해서는 아래와 같은 문제점 등이 수집되어 분석될 필요가 있다.

- 검토 활동을 수행하는 과정에서 발견된 문제점
- 프로젝트 계획서의 계획과 비교하여 프로젝트 속성들에 대한 심각한 편차
- 만족되지 못한 공약 (내부 및 외부 공약)
- 위험 상태에 대한 중대한 변경
- 이해관계자와 관련된 문제점 (업무 진행 상태, 각종 회의 참여 상태 등)

시정조치 활동이 수행되는 형태는 아래에 기술한 예시 등과 같다.

- 업무기술서(Statement of Work)의 변경
- 요구사항의 변경
- 견적 내용 및 계획서의 개정
- 공약사항에 대한 재협상
- 프로젝트 자원 조정
- 프로세스 변경
- 프로젝트위험 상태 갱신

☐ 산출물: [14-02]시정조치 관리대장, [13-16]변경요청서

### 3.5. 프로젝트 종료 검토 수행

향후 수행될 프로젝트의 수행 가능성을 향상시키고 견적 이력 데이터를 갱신하기 위한 경험기록을 제공하기 위하여 프로젝트 성과 검토를 수행한다.

프로젝트 종료 검토를 수행하기 위해서는 아래와 같은 정보 등이 수집되고, 보고된다.

- 프로젝트 속성 모니터링 데이터 (계획 및 실적)
- 프로젝트 위험 관련 데이터
- 프로젝트 이슈 관련 데이터
- 기술적 및 관리적 측면에서의 성공 및 실패 사례 등과 같은 프로젝트 교훈 데이터

☐ 산출물: [13-19]검토 기록

## 4. 수행 결과

- 프로젝트 계획이 실행된다.
- 프로젝트 계획 내용이 모니터링되고, 보고된다.
- 프로젝트 진척도가 모니터링되고, 보고된다.
- 프로젝트 계획 대비 심각한 편차가 발생한 경우, 재발 방지를 위한 시정조치 활동이 수행된다.
- 프로젝트 경험기록을 제공하기 위해 프로젝트 종료 검토가 수행된다.

## 5. 측정 항목

측정 항목(단위)	측정 방법
실적 소프트웨어 규모	실제 개발된 소프트웨어 규모(예를 들면, 코드 라인 수(Line of Code; LOC), 기능 점수(Function Point; FP) 등)를 측정한다.
실적 투입공수	인원에 대한 실제 투입공수(예를 들면, Man/Month, Man/Day 등)를 측정한다.
실적 예산	실제 집행된 프로젝트 예산을 측정한다.
실적 일정	실제 경과된 프로젝트 추진일정을 측정한다.
실적 작업량	정의된 측정 주기까지의 실제 작업량(예를 들면, 투입공수 등)을 측정한다.
전체 작업량	계획된 프로젝트 전체 작업량(예를 들면, 투입공수 등)을 측정한다.

(사용 예시)

- 실적 진척도(%): (측정 주기까지의 실제 작업량 / 프로젝트 전체 작업량) X 100
- 일정 준수율(%): (실제 진척도 / 계획 진척도) X 100
- 공수 투입율(%): (측정 주기까지의 실적 투입공수 /  $\Sigma$ 계획 투입공수) X 100
- 투입공수 편차: 측정 주기까지의 계획 투입공수 - 측정 주기까지의 실적 투입 공수
- 예산 집행율(%): (측정 주기까지의 실적 집행예산 /  $\Sigma$ 계획 예산) X 100
- 예산 편차: 측정 주기까지의 계획 예산 - 측정 주기까지의 실적 예산

## 6. 프로세스 조정

- "프로젝트 종료 검토 수행" 활동은 보다 적극적인 방법으로 수집된 프로젝트 관련 실적 데이터를 기초로 프로젝트 종료 보고서를 작성하여 검토를 수행하는 것이 바람직하다.

## 7. 참고 문헌

- KXSISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- Purpose (Capability Maturity Model Integration for Development), Version 1.2
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology



# MAN.5 위험 관리 프로세스

## 1. 개요

### 1.1. 목적

위험 관리 프로세스의 목적은 지속적으로 잠재적인 문제점을 발생하기 이전에 식별, 분석, 조치 및 모니터링 하여 프로젝트 목적 달성에 미치는 악영향을 줄이기 위한 것이다.

### 1.2. 설명

본 프로세스는 프로젝트의 착수단계부터 완료단계까지의 프로젝트 생명주기 전 과정에 걸쳐 적용된다.

위험 관리는 지속적이면서 관리 활동의 중요한 요소인 미래를 예측하는 프로세스이다. 위험 관리는 프로젝트 목표 달성을 저해하는 잠재되어 있는 문제점들을 처리한다. 지속적인 위험 관리 접근방법은 프로젝트에 치명적인 영향을 주는 위험을 효과적으로 예측하고 완화시킨다.

효과적인 위험 관리에는 프로젝트 관련 이해관계자들의 협력 및 참여를 통하여 프로젝트 초기에 적극적으로 위험을 식별하는 활동이 포함된다. 프로젝트 후반부 보다는 초기 단계에서 위험을 식별하여 제거하거나 완화하는 것이 더 쉽고, 비용도 절약되기 때문이다. 또한 위험 관리는 비용, 일정, 기술적인 측면과 관련된 위험에 대한 내부 및 외부 출처(source)들이 고려되어야 한다.

### 1.3. 주요 관련 프로세스

ID	프로세스	설명
MAN.3-1	프로젝트 계획 수립 프로세스	프로젝트 계획은 프로젝트 위험을 식별하기 위한 기초 자료로 활용되며, 식별된 위험을 제거하거나 완화하기 위하여 프로젝트 계획을 조정할 수도 있다.
MAN.3-2	프로젝트 진행 관리 프로세스	프로젝트 진행 중에 식별되는 위험은 위험 관리 프로세스에 따라 처리되며, 위험 관리 프로세스에서 식별된 위험 또한 프로젝트 진행 관리 프로세스에서 모니터링 된다.

## 1.4. 용어 정의

- 위험 (Risk): 프로젝트 일정, 예산, 작업산출물 등에 부정적인 영향을 미쳐 정해진 기간 내에 프로젝트가 종료되지 못하거나, 고객 요구사항을 만족시키지 못할 가능성이 있는 잠재적 문제점
- 위험 관리 (Risk Management): 위험을 식별, 분석 및 계획을 수립하고, 계획에 따라 위험을 모니터링, 완화 및 대처 하는 일련의 활동을 의미함
- 위험 완화 (Risk Mitigation): 위험이 문제점으로 현실화되기 이전에 발생 확률을 줄이기 위하여 수행하는 활동을 의미함
- 이해관계자 (Stakeholders): 프로젝트에 직접 또는 간접적으로 관련이 있는 그룹 또는 개인

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트가 착수되었다. 프로젝트 관리자가 선임되었다.
입력물	[12-01]제안 요청서 [02-00]계약(서) [17-03]고객 요구사항 [08-12]프로젝트 계획서 [08-06]프로젝트 활동도 [14-09]작업 분해 구조도 [14-06]일정 [13-14]진척도 상태 기록 [15-06]프로젝트 상태 보고서
수행활동	위험 관리 범위 설정 위험 관리 전략 정의 위험 식별 위험 분석 및 우선순위 결정 위험 완화계획 수립 및 실행 위험 모니터링 예방 또는 시정 조치
출력물	[08-19]위험관리 계획서 [15-08]위험분석 보고서 [08-20]위험완화 계획서 [14-02]시정조치 관리대장
완료조건	프로젝트가 종료되었다.

## 2.2. 책임과 역할

활동 \ 역할	프로젝트 관리책임자	개발자	위험관리 담당자	이해관계자
1. 위험 관리 범위 설정	●	○		
2. 위험 관리 전략 정의	●	○		
3. 위험 식별	●	○		○
4. 위험 분석 및 우선순위 결정	○	○	●	○
5. 위험 완화계획 수립 및 실행	○	○	●	○
6. 위험 모니터링	●	○	○	○
7. 예방 또는 시정 조치	○	○	●	○

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 위험 관리 범위 설정

프로젝트 관리자는 조직의 위험 관리 정책에 따라 수행되어야 할 위험 관리 범위를 설정한다.

☞ 위험 관리 범위를 설정하는 것은 위험을 수집하고 체계화하기 위한 것이며, 향후 위험 완화 계획의 통합 활동을 지원하기 위한 것이다. 일반적인 위험 관리 범위는 아래 예와 같다.

- 프로젝트 규모 관련 위험: 견직한 규모의 신뢰 수준, 재사용 소프트웨어의 비용, 제품의 복잡성 등을 말한다.
- 프로젝트 수행 관련 위험: 프로젝트 수행 관점에서의 위험은 기술적인 관점에서 보는 것과는 다를 수 있다. 예를 들면 인도 일정이 적절한지, 또는 프로젝트 수행 결과가 어느 정도의 비용 초과에 대한 영향을 미치는지 등을 의미한다.
- 고객 관련 위험: 부적절한 고객은 프로젝트 계획을 상당히 위협하고, 프로젝트 관리자에게 실질적인 위험이 된다.
- 개발 프로세스 관련 위험: 부득이한 상황 때문에 개발 프로세스를 부적절하게 정의하여, 그 프로세스에 따라 프로젝트를 수행한다면 위험한 상황에 처할 수 있다.
- 기술 관련 위험: 프로젝트 수행에 필요한 기술을 가지고 있지 않은 경우 프로젝트 성공에 지대한 악영향을 미친다.

- 개발 환경 관련 위험: 부적절하거나 비효율적인 도구는 훌륭한 기술을 갖고 있는 사람도 무능하게 만든다. 개발 환경은 프로젝트 팀/프로세스/제품을 지원한다. 그러나 만약 이러한 개발 환경에 문제가 있다면, 이것은 심각한 위험의 원인이 될 수 있다.
- 프로젝트 구성원 관련 위험: 프로젝트 규모에 적합한 경험을 가진 기술진을 보유하고 있지 않거나 충원이 어렵다면, 프로젝트를 성공적으로 수행할 수 없을 것이다. 또한 해당 분야에 대해 프로젝트 경험이 있느냐, 있지 않느냐에 따라 위험의 정도는 커질 것이다.

☐ 산출물: [08-19]위험관리 계획서

### 3.2. 위험 관리 전략 정의

프로젝트 관리자는 프로젝트와 조직 차원에서 위험을 식별하고, 분석하고, 처리하고, 모니터링하기 위한 적절한 위험 관리 전략 및 위험 척도를 정의한다.

일반적인 위험 관리 전략은 다음과 같은 사항들을 다룬다.

- 위험관리 투입공수의 범위
- 위험 식별, 분석, 완화, 모니터링 및 의사소통에 사용될 방법 및 도구
- 불명확한 요구사항, 불분명한 책임 및 역할, 부정확한 예측 및 견적, 부적절한 투입자원 등과 같은 위험 출처(source)
- 위험 분류 및 통합 방법
- 프로토타이핑, 시뮬레이션, 대안 설계 또는 점진적 개발 등과 같은 위험 완화 기법
- 위험의 상태를 모니터링하기 위한 지표 정의
- 위험 모니터링 또는 위험 재평가 주기

위험을 평가하고, 우선순위를 결정하기 위한 위험 척도는 아래에 기술한 예시와 같으며, 이에 대한 값의 범위는 조직이나 프로젝트의 특성에 적절하게 정의한다.

- 위험 발생 가능성
- 위험 발생 영향도
- 관리 활동이 착수되는 시점에 대한 임계점(thresholds)

☐ 산출물: [08-19]위험관리 계획서

### 3.3. 위험 식별

프로젝트 관리자는 개발자와 함께 비용, 일정, 투입공수, 자원 및 기술 등에 관련된 위험을 식별한다.

☞ 위험은 프로젝트 초기에는 프로젝트 전략 및 계획 등을 토대로 식별되고, 프로젝트 수행 중에는 프로젝트 진행 상태를 토대로 식별된다. 위험을 식별하는 방법에는 여러 가지가 있으며, 아래에 기술한 내용은 대표적으로 활용되는 방법의 예시이다.

- 브레인스토밍(brainstorming): 프로젝트 구성원과 함께 브레인스토밍 회의를 통하여 위험을 식별하는 방법으로, 위험의 출처를 같이 공유할 수 있어서 체계적으로 수행하면 위험 요인을 사전에 식별하는 데에 많은 도움이 되는 방법이다.
- 체크리스트: 사전에 위험의 범위 또는 출처(source) 별로 체크 항목을 만들어서 위험 요인을 조사하는 방법이다.
- 설문조사: 다수의 인원을 대상으로 사전에 준비된 설문지를 배포하여 위험을 식별하는 방법이다.
- 인터뷰: 프로젝트의 핵심 인력 또는 해당 분야별 전문가들과의 인터뷰를 통해서 위험 요인을 식별하는 방법이다.
- 유사 프로젝트 검토: 비슷한 상황에서의 위험 정도 및 빈도는 거의 비슷하게 나타나므로, 과거의 유사한 프로젝트에서 발생한 위험의 원인 및 정도를 파악하여 위험 요인을 식별하는 방법이다.

☐ 산출물: [15-08]위험분석 보고서

### 3.4. 위험 분석 및 우선순위 결정

위험관리 담당자는 식별된 각 위험에 대해 발생 가능성 및 영향도 등을 분석하여, 위험을 모니터링하기 위해 적용할 자원의 우선순위를 결정한다.

☞ 사전에 정의된 위험 척도에 따라 우선순위가 결정된다. 우선순위를 결정하는 기준이 아래에 기술된 표와 같다고 가정할 때, 식별된 특정 위험에 대한 발생 가능성이 70%(0.7)이고 영향도가 4점 척도에서 3에 해당한다면, 이 두 값을 곱하여 산출된 중요도 값이 2.1이 되고, 이에 대한 우선순위는 "보통"이 된다.

우선순위	판단기준	설 명
매우 낮음 (Very Low)	중요도 ≤ 0.8	프로젝트의 목적을 달성하는데 전혀 영향을 미치지 않거나 사소하게 영향을 미치는 위험 요인으로서, 위험의 발생 확률 또한 염려하지 않을 정도로 매우 낮음
낮음 (Low)	0.8 < 중요도 ≤ 1.6	프로젝트의 목적을 달성하는데 아주 적은 영향을 미치는 위험 요인으로서, 위험의 발생 확률은 크게 염려하지 않아도 될 정도로 낮음
보통 (Moderate)	1.6 < 중요도 ≤ 2.4	프로젝트의 목적을 달성하는데 영향을 미치는 위험 요인으로서, 약 중간 정도(50:50)의 위험 발생 확률을 가지고 있음
높음 (High)	2.4 < 중요도 ≤ 3.2	프로젝트의 목적·비용·일정 등에 영향을 미치는 위험 요인으로서, 약간 높은 위험 발생 확률을 가지고 있음
매우 높음 (Very High)	3.2 < 중요도	프로젝트의 목적·비용·일정 등에 매우 심각한 영향을 미치는 위험 요인으로서, 위험의 발생 확률 또한 매우 높음

☐ 산출물: [15-08]위험분석 보고서

### 3.5. 위험 완화계획 수립 및 실행

위험관리 담당자는 각 위험을 허용 가능한 수준까지 완화하기 위해 적절한 조치 활동을 계획하고, 수행한다.

☞ 위험을 완화하기 위해서는 적절한 위험 처리 기법이 결정되어야 하며, 이러한 처리 기법 결정은 위험 완화활동에 대한 비용 대 이익을 고려하여야 한다. 다음은 위험 처리에 대한 기법의 예시를 기술한 것이다.

- 회피(Avoidance): 프로젝트 계획을 변경하거나 또는 특정한 조치 활동을 통해 위험을 제거하는 것을 말한다.
- 통제(Control): 위험의 발생 가능성 및 영향도를 허용 가능한 한계점(threshold) 까지 줄이는 것을 말한다. 위험 발생 가능성 또는 프로젝트에 대한 위험의 영향을 줄이기 위해 초기 행동을 취하는 것은 위험이 발생한 후 조치하는 것보다 더욱 효율적이다.
- 전가(Transference): 위험의 영향을 보험이나 업무 재조정 등을 통해 제 3자에게 이동시키는 것을 말한다. 위험 전가는 단순히 그 관리에 대한 책임을 다른 조직에게 주는 것일 뿐, 위험을 제거하는 것은 아니다.
- 모니터링(Monitoring): 할당된 위험 척도에 대한 변경을 위해 위험을 모니터링하고 주기적으로 재평가하는 것을 말한다. 어떠한 조치활동도 수행되지 않는다.

- 수용(Acceptance): 프로젝트가 위험을 처리하기 위해 적절한 대응 전략을 정의하는 것이 불가능하거나, 위험의 발생 가능성 및 영향이 낮다는 것을 말한다. 적극적인 수용은 위험 발생시 실행하기 위한 대처 계획(contingency plan) 개발을 포함할 수 있다. 소극적인 수용은 어떠한 행동도 요구하지 않으며, 위험 발생시 프로젝트 팀은 위험을 처리해야 한다.

☐ 산출물: [08-20] 위험완화 계획서

### 3.6. 위험 모니터링

프로젝트 관리자는 각 위험의 현재 상태를 모니터링하여 위험 상태의 변경을 결정하고, 위험 완화를 위한 조치 활동의 효과성을 평가한다.

☞ 위험 완화계획 실행 중에 위험을 효과적으로 통제 및 관리하기 위해 위험과 위험 완화활동에 대한 상태 및 결과를 주기적으로 모니터링하고, 필요한 경우 위험 상태 또는 위험 완화계획 등을 개정한다.

☐ 산출물: [08-20] 위험완화 계획서

### 3.7. 예방 또는 시정 조치

프로젝트 관리자는 위험 완화를 위한 조치 활동의 성과가 미흡한 경우, 향후 각 위험의 영향을 줄이거나 회피하기 위한 적절한 예방조치 활동을 수행한다.

위험 완화를 위한 조치 활동으로 위험을 줄이거나 회피할 수 없는 경우, 위험으로부터 발생하는 문제점을 해결하기 위한 시정조치 활동을 수행한다.

☞ 예방조치 활동은 위험 완화계획의 개정을 포함하고 있으며, 시정조치 활동은 사전에 위험 대처 계획이 수립된 경우, 이 계획을 실행하는 것을 포함하고 있다.

☐ 산출물: [14-02]시정조치 관리대장

## 4. 수행 결과

- 수행되어야 할 위험 관리 범위가 결정된다.
- 적절한 위험 관리 전략이 정의되고, 구현된다.

- 프로젝트 수행 과정에서 발생하는 위험이 식별된다.
- 위험이 분석되고, 위험을 처리하기 위해 적용할 자원의 우선순위가 결정된다.
- 위험 상태의 변화와 처리 활동의 진척도를 결정하기 위해 위험 척도가 정의되고, 적용되고, 평가된다.
- 위험의 우선순위, 발생확률, 발생결과(consequence) 또는 기타 정의된 임계점(threshold) 등에 근거하여 위험의 영향을 제거하거나 회피하기 위해 적절한 조치가 취해진다.

## 5. 측정 항목

측정 항목(단위)	측정 방법
위험 건수	식별된 위험의 건수를 측정한다.
위험 제거건수	위험 건수 중에서 위험 완화활동으로 인하여 제거된 위험건수를 측정한다.
현실화된 위험 건수	위험건수 중에서 위험 완화활동의 실패로 인하여 실제로 문제점으로 발생한 위험건수를 측정한다.

(사용 예시)

- 위험 제거율(%):  $(\sum \text{위험 제거건수} / \sum \text{위험 건수}) \times 100$
- 위험 발생율(%):  $(\sum \text{현실화된 위험 건수} / \sum \text{위험 건수}) \times 100$

## 6. 프로세스 조정

- "[08-19]위험관리 계획서"는 프로젝트 규모, 특성 및 환경에 따라 "[08-12]프로젝트 계획서"에 통합하여 작성할 수 있다. 이 경우, "[08-12]프로젝트 계획서"에 "[08-19]위험관리 계획서"의 내용을 모두 포함하여 정의한다.

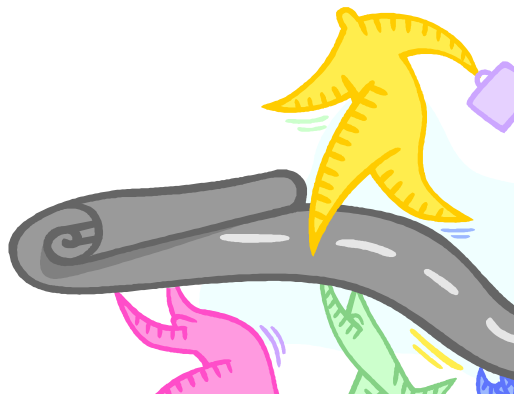
## 7. 참고 문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- CMMI-DEV(Capability Maturity Model Integration for Development), Version 1.2
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology



## 제 7 장 지원 프로세스 그룹(SUP)

- SUP.1 품질 보증
- SUP.8 형상 관리 프로세스
- SUP.9 문제 해결 관리 프로세스
- SUP.10 변경 요청 관리 프로세스



# SUP.1 품질 보증 프로세스

## 1. 개요

### 1.1. 목적

품질 보증 프로세스의 목적은 작업산출물과 프로세스들이 사전에 정의된 요구사항과 계획서에 부합함을 보증하는 것이다.

### 1.2. 설명

본 프로세스는 프로젝트의 착수단계부터 완료단계까지 프로젝트 생명주기 전 과정에 걸쳐 적용된다.

품질 보증 프로세스는 다음과 같은 활동을 포함한다.

- 적용해야 할 프로세스기술서, 표준, 절차를 기준으로 수행된 프로세스 및 작업 산출물에 대한 평가
- 부적합 이슈의 식별 및 문서화
- 부적합 이슈가 조치됨을 보장
- 프로젝트 구성원 및 관리자들에게 품질보증 활동 결과를 제공

품질 보증 프로세스는 프로젝트의 생명주기 전반에 걸쳐 프로젝트 구성원과 관리자에게 프로세스와 관련 작업산출물에 대한 적절한 가시성을 제공하고, 피드백을 통한 고품질의 제품 및 서비스를 인도하는 것을 지원한다.

### 1.3. 주요 관련 프로세스

ID	프로세스	설명
MAN.3-1	프로젝트 계획 수립 프로세스	프로젝트 계획은 품질보증 전략 수립의 근간으로 활용된다.
SUP.9	문제 해결 관리 프로세스	품질보증 활동 결과 발생하는 부적합사항을 처리하기 위해, 문제 해결 관리 프로세스가 수행될 수도 있다.
SUP.10	변경 요청 관리 프로세스	품질보증 활동 결과 발생하는 부적합사항을 처리하기 위해, 변경 요청 관리 프로세스가 수행될 수도 있다.

## 1.4. 용어 정의

- 이해관계자 (Stakeholders): 프로젝트에 직접 또는 간접적으로 관련이 있는 그룹 또는 개인
- 품질 보증 (Quality Assurance): 항목 또는 산출물들이 기술적인 요구사항들을 만족하고 있음을 증명하기 위해 요구되는 계획적이면서도 체계적인 모든 활동

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트가 착수 되었다. 품질보증 담당자가 선임되었다.
입력물	[12-01]제안 요청서 [02-00]계약(서) [17-03]고객 요구사항 [08-12]프로젝트 계획서 [08-06]프로젝트 활동도 [14-09]작업 분해 구조도 [14-06]일정
수행활동	품질보증 전략 수립 품질기록 정의 품질보증 활동 수행 부적합사항 식별 및 보고 부적합사항 조치
출력물	[08-13]품질 계획서 [13-18]품질 기록 [13-07]문제 기록 [13-19]검토 기록 [14-02]시정조치 관리대장
완료조건	프로젝트가 종료되었다. 품질보증 활동 결과 발견된 모든 문제점 및 부적합 사항들이 조치되었다.

## 2.2. 책임과 역할

역할 활동	품질보증 담당자	개발자	프로젝트 관리책임자	이해관계자
1. 품질보증 전략 수립	●	○	○	○
2. 품질기록 정의	●	○	○	
3. 품질보증 활동 수행	●	○	○	○
4. 부적합사항 식별 및 보고	●	○	○	○
5. 부적합사항 조치	○	●	○	○

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 품질보증 전략 수립

제품 및 프로세스에 대한 품질보증 활동을 수행하기 위해 조직 차원의 품질관리 전략을 고려하여 프로젝트 차원의 품질보증 전략을 수립한다.

☞ 품질보증 전략은 일반적으로 아래와 같은 내용 등을 포함하며, 품질 계획서의 일부 내용으로 통합되어 문서화된다.

- 품질보증 활동 목적
- 품질보증 활동 조직 및 역할
- 품질보증 활동 기준 (프로젝트 적용 프로세스기술서, 표준, 절차, 규약 등)
- 품질보증 활동 내역 및 절차 (검토, 검증, 감사, 평가 활동과 관련 문서 및 기록 등)
- 부적합사항 보고 및 시정조치
- 품질보증 활동 주기 및 일정

☐ 산출물: [08-13]품질 계획서

### 3.2. 품질기록 정의

프로세스 및 작업산출물이 품질 요구사항에 적합하다는 것을 보장하기 위해 관련 품질기록을 정의한다.

☞ 품질기록은 일반적으로 아래와 같은 문서 및 기록 등을 포함하며, 품질 계획서 또는 프로젝트 계획서의 일부 내용으로 통합되어 문서화된다.

- 프로세스 및 작업산출물 검토, 검증, 감사, 평가 보고서
- 부적합 보고서
- 시정조치 관련 문서 및 기록
- 품질보증 활동 보고서
- 품질경향 보고서

☞ 산출물: [08-13]품질 계획서

### 3.3. 품질보증 활동 수행

프로젝트의 프로세스가 규정된 표준에 부합하는지를 보장하기 위해, 그리고 프로젝트의 작업산출물이 품질 요구사항을 만족시키는지를 보장하기 위해 적절한 주기로 품질보증 활동을 수행한다.

☞ 품질보증 활동은 품질보증 전략에 정의된 활동 내역(유형), 주기 및 일정 등을 기준으로 프로젝트 생명주기 전반에 걸쳐 지속적으로 수행된다. 아래에 기술한 내용은 일반적으로 수행되는 품질보증 활동에 대한 예시이다.

- 사전에 정의된 프로세스기술서, 표준, 절차 등의 기준 대비 프로세스 및 작업산출물 감사
- 요구사항명세서, 설계명세서, 시험명세서 등에 대한 품질 검토(review)
- 소프트웨어 코드 검사(inspection)
- 소프트웨어 시험(test)
- 외주업체 프로세스 및 작업산출물에 대한 감사

☞ 산출물: [13-18]품질 기록, [13-19]검토 기록

### 3.4. 부적합 사항 식별 및 보고

품질보증 활동 결과 발견된 문제점 및 부적합 사항들을 식별하고, 기록하고, 이를 해결하기 위해 적절한 이해관계자에게 보고한다.

☞ 부적합 사항들은 각각의 품질보증 활동 결과에 대한 감사보고서, 검토결과서, 검사결과서, 시험보고서 등의 결과보고서 형태로 문서화된다. 그리고 이 결과보고서에

는 초기에서부터 현재까지의 품질 경향을 분석한 결과가 포함되는 것이 바람직하다. 이러한 결과보고서는 프로젝트 구성원뿐만 아니라 관리자를 포함한 관련 이해관계자와 함께 검토한다.

☐ 산출물: [13-07]문제 기록, [13-19]검토 기록

### 3.5. 부적합 사항 조치

합의된 요구사항 또는 품질 목표 관련 문제점 및 부적합 사항들을 분석하고, 해결한다.

부적합 사항을 프로젝트 자체적으로 해결하지 못하는 경우에는 사전에 설정되어 있는 보고체계를 통해 적절한 수준의 조직 내부자에게 보고하고, 그 이슈를 해결할 수 있도록 보장하여야 한다. 또한 부적합 사항은 해결될 때까지 추적되어야 한다.

☐ 산출물: [14-02]시정조치 관리대장

## 4. 수행 결과

- 품질보증 활동 수행을 위한 전략이 수립된다.
- 품질보증 활동과 관련된 문서 및 기록이 작성되고 유지된다.
- 합의된 요구사항에 대한 문제점 및 부적합사항이 식별되고 기록된다.
- 제품, 프로세스 및 활동들이 적용 가능한 표준, 절차 및 요구사항을 따르고 있는지 검증된다.

## 5. 측정 항목

측정항목(단위)	측정 방법
부적합 건수	품질보증 활동 결과 식별된 문제점 건수를 측정한다.
부적합 조치건수	식별된 문제점 건수 중에서 조치된 건수를 측정한다.

(사용 예시)

- 부적합 조치율(%) :  $(\sum \text{부적합 조치건수} / \sum \text{부적합 건수}) \times 100$

## 6. 프로세스 조정

- "품질보증 담당자"는 프로젝트 규모, 특성 및 환경에 따라 특정 직원 또는 그룹에게 그 역할이 부여되지 않고 "개발자"들이 품질보증 역할을 나누어서 수행할 수도 있다. 이 경우, "품질보증 담당자"는 역할을 부여 받은 각각의 개발자가 된다.
- "[08-13]품질 계획서"는 프로젝트 규모, 특성 및 환경에 따라 "[08-12] 프로젝트 계획서"에 통합하여 작성할 수 있다. 이 경우, "[08-12] 프로젝트 계획서"에 품질보증과 관련된 내용을 모두 포함하여 정의한다.

## 7. 참고 문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- CMMI-DEV (Capability Maturity Model Integration for Development, Version 1.2
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

## SUP.8 형상 관리 프로세스

### 1. 개요

#### 1.1. 목적

형상 관리 프로세스의 목적은 프로세스 또는 프로젝트의 작업산출물/항목의 무결성을 확립하고, 유지하고, 관련 이해관계자들에게 활용 가능하도록 하는 것이다.

#### 1.2. 설명

본 프로세스는 프로젝트의 착수단계부터 완료단계까지의 프로젝트 생명주기 전 과정에 걸쳐 적용된다.

형상 관리 프로세스는 다음과 같은 활동들을 포함한다.

- 특정 시점에서의 베이스라인을 구성하도록 선정된 작업산출물의 형상을 식별
- 형상항목의 변경 통제
- 형상관리시스템으로부터 작업산출물 명세서의 작성 또는 제공
- 베이스라인의 무결성 유지
- 개발자, 최종사용자, 고객에게 정확한 형상 상태 및 현재의 형상 데이터 제공

형상 관리 프로세스는 형상 식별, 형상 통제, 형상 상태 보고, 형상 감사 등의 방법으로 작업산출물의 무결성을 확보하고, 유지하는 것이다.

#### 1.3. 주요 관련 프로세스

ID	프로세스	설명
SUP.9	문제 해결 관리 프로세스	프로젝트 생명주기동안에 발생하는 문제점을 조치하기 위해, 베이스라인이 설정된 소스 코드나 문서 등에 대한 변경과 관련하여 형상관리 프로세스가 수행된다.
SUP.10	변경 요청 관리 프로세스	프로젝트 생명주기 동안에 발생하는 새로운 요구사항이나 요구사항 변경, 설계 변경, 시스템 개선 등과 관련하여, 베이스라인이 설정된 소스 코드나 문서 등을 변경하기 위해 형상관리 프로세스가 수행된다.



## 1.4. 용어 정의

- 릴리스 (Release): 승인된 버전(version)의 공식적인 통지 및 배포
- 버전 (Version): 형상항목에 대한 초기 릴리스 또는 재 릴리스
- 베이스라인 (Baseline): (1)공식적으로 검토되고 합의된 하나의 명세서(specification) 또는 제품(product). 이러한 것들은 베이스라인 설정 이후 향후 개발을 위해 근간으로 제공되고, 공식적인 변경 통제 절차를 통해서만 변경될 수 있음. (2) 형상 항목의 생명주기 동안 규정된 시점에서 지정되고 확정된 하나의 문서 또는 문서들의 집합
- 이해관계자 (Stakeholders): 프로젝트에 직접 또는 간접적으로 관련이 있는 그룹 또는 개인
- 형상 (Configuration): 기술적인 문서에 정의되거나 제품에 포함되는 하드웨어 또는 소프트웨어에 대한 기능적, 물리적 특성
- 형상 감사 (Configuration Audit): 형상 관리의 일부 요소로서, 개발된 형상 항목이 만족하게 완료되었는지를 검증하기 위해 수행되는 감사 활동. 감사의 관점에 따라 기능적 형상 감사(Functional Configuration Audit; FCA)와 물리적 형상 감사(Physical Configuration Audit; PCA)로 구분할 수 있음
- 형상 관리 (Configuration Management; CM): 형상항목에 대한 기능적, 물리적 특성을 식별하여 문서화하고, 문서화된 특성에 대한 변경을 통제하고, 변경 과정 및 상태를 기록하여 보고하고, 규정된 요구사항에 부합함을 검증하기 위해 적용하는 기술적/행정적 관리 및 감독 활동과 관련된 엄격한 규율
- 형상 상태 보고 (Configuration Status Accounting): 형상 관리의 일부 요소로서, 형상을 효과적으로 관리하기 위해 요구되는 정보를 기록하고 보고하는 활동으로 구성됨. 이 정보는 승인된 형상 항목에 대한 목록, 형상을 변경하기 위해 제안된 상태, 승인된 변경에 대한 구현 상태를 포함함
- 형상 식별 (Configuration Identification): 형상 관리의 일부 요소로서, 시스템에 대한 형상 항목을 선정하고, 선정된 형상 항목에 대한 기술적, 물리적 특성들을 기술적인 문서에 기록하는 활동들로 구성됨
- 형상 통제 (Configuration Control): 형상 관리의 일부 요소로서, 공식적인 베이스라인이 설정된 이후의 형상 항목에 대한 변경을 평가하고, 조정하고, 승인 또는 반려하고, 구현하는 활동들로 구성됨
- 형상 항목 (Configuration Item; CI): (1)하드웨어나 소프트웨어, 또는 그 모두에 대한 집적체로서, 형상 관리를 위해 지정되며, 형상 관리 프로세스에서 하나의 개체로서 취급됨. (2)명세서(specification), 도면(drawings), 관련 목록, 그리고 이러한 것들에서 언급하는 문서로 정의되는 형상 항목에 대한 현재 승인된 기술적인 문서. 하드웨어 형상 항목(Hardware Configuration Item; HWCI)과 컴퓨터 소프트웨어 형상 항목 (Computer Software Configuration Item; CSCI)로 구분하기도 함

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트가 착수되었다. 형상관리 담당자가 선임되었다.
입력물	[17-03]고객 요구사항 [08-12]프로젝트 계획서 [08-06]프로젝트 활동도 [14-09]작업 분해 구조도 [14-06]일정 [16-03]형상 관리 라이브러리 [13-07]문제 기록 [13-16]변경요청서 [17-11]소프트웨어 요구사항 [17-08]인터페이스 요구사항 [04-04]상위수준 소프트웨어 설계 [04-05]하위수준 소프트웨어 설계 [11-02]소프트웨어 구성요소 [08-25]단위시험 계획서 [11-05]소프트웨어 단위 [08-10]소프트웨어 통합 시험 계획서 [08-21]소프트웨어 시험계획서 [10-02]시험절차서 [17-14]시험사례명세서 [11-01]소프트웨어 제품
수행활동	형상 관리 전략 수립 형상항목 식별 브랜치 관리 전략 수립 베이스라인 설정 형상항목 설명서 유지 변경 및 릴리스 통제 형상항목 이력 유지 형상 상태 보고 형상항목 정보 검증 형상항목 백업, 저장, 파일보관, 취급 및 인도 관리
출력물	[08-04]형상 관리 계획서 [14-01]변경 이력 [15-03]형상 상태 보고서 [15-24]감사 보고서 [13-19]검토 기록 [13-03]백업/복구 기록 [13-06]인도 기록
완료조건	형상 항목이 성공적으로 고객에게 인도되었다. 프로젝트가 종료 되었다.

## 2.2. 책임과 역할

역할 \ 활동	형상관리 담당자	개발자	프로젝트 관리책임자	이해관계자
3.1 형상관리 전략 수립	●	○	○	
3.2 형상항목 식별	●	○	○	
3.3 브랜치 관리 전략 수립	●	○	○	
3.4 베이스라인 설정	○	○	●	○
3.5 형상항목 설명서 유지	○	●		
3.6 변경 및 릴리스 통제	●	○	○	
3.7 형상항목 이력 유지	○	●		
3.8 형상 상태 보고	●	○	○	○
3.9 형상항목 정보 검증	●	○	○	○
3.10 형상항목 백업, 저장, 파일보관, 취급 및 인도 관리	●	○	○	○

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 형상관리 전략 수립

형상관리 활동과 이러한 활동을 수행하기 위해 형상관리 전략을 수립한다.

☞ 형상관리 전략은 일반적으로 아래와 같은 내용 등을 포함하며, 형상관리 계획서 또는 프로젝트 계획서의 일부 내용으로 통합되어 문서화된다.

- 형상관리 목적
- 형상관리 조직 및 역할
- 형상항목 식별 및 정의
- 형상 베이스라인 설정
- 형상 관리 라이브러리 체계
- 형상 변경 통제 절차
- 브랜치(branch) 관리 전략

- 형상 상태 보고 주기 및 절차
- 형상 감사 주기 및 절차

☐ 산출물: [08-04]형상관리 계획서

### 3.2. 형상항목 식별

독립적으로 식별, 저장, 검토, 시험, 사용, 변경, 인도 및 유지될 필요가 있는 형상항목을 식별한다.

☞ 식별된 형상항목의 내용은 일반적으로 형상관리 계획서에 정의되며, 형상항목에 접근하고 저장하는 효율적인 수단을 제공하기 위해 자동화된 형상관리 시스템과 같이 파일 및 디렉토리 구조와 계층적 체계(hierarchies)가 수립될 수도 있다.

형상항목 식별은 아래에 기술한 예시와 같은 사항들을 선정, 생성, 명세화하는 것이다.

- 고객에게 인도되는 제품
- 지정된 내부 작업산출물
- 외부로부터 획득되는 제품
- 개발 환경 및 도구
- 상기 작업산출물들을 생성하고 설명하는 데에 활용되는 기타 항목

형상항목 식별을 위해 고려되어야 하는 기준의 예시는 아래와 같다.

- 두 개 이상의 그룹에 의해 사용되는 작업산출물
- 결함이나 요구사항 변경으로 인해 여러 차례 변경이 예상되는 작업산출물
- 한 작업자의 변경이 또 다른 작업자의 변경을 초래하는, 서로 영향을 미치는 상호 의존적인 작업산출물
- 프로젝트에서 아주 중요하다고 판단되는 작업산출물

형상항목이 될 수 있는 작업산출물의 예시는 아래와 같다.

- 프로세스 기술서
- 요구사항
- 요구사항명세서
- 설계명세서
- 인터페이스명세서

- 소프트웨어 소스 코드
- 시험 계획서 및 명세서
- 사용자 매뉴얼
- 개발 환경 및 도구

☐ 산출물: [08-04]형상관리 계획서

### 3.3. 브랜치 관리 전략 수립

동일한 소스 코드를 기준으로 사용하여 동시에 개발을 진행할 수 있도록 브랜치 (branch) 관리 전략을 수립한다.

☞ 브랜치 관리 전략에는 브랜치 관리, 머징(merging), 브랜치 시스템 상에서 파일 버전 생성 및 관리, 브랜치 병합(parenting) 및 태깅(tagging) 전략 등이 포함된다. 또한 브랜치 전략은 브랜치가 왜 그리고 언제 생성될 것인지, 브랜치에서 무슨 활동들이 수행되는지, 브랜치가 어떻게 완료되어 주(main) 소스 코드로 이전되는지가 정의된다. 이러한 브랜치 관리 전략은 형상관리 계획서에 정의된다.

☐ 산출물: [08-04]형상관리 계획서

### 3.4. 베이스라인 설정

내부 베이스라인 및 인도 베이스라인을 설정한다. 상위 수준의 베이스라인은 모든 필요한 형상항목을 각 하위 수준의 베이스라인에 공식적으로 축척함으로써 달성된다.

☞ 베이스라인에는 "3.2 형상항목 식별" 조항에서 정의된 적절한 요구사항, 설계 문서, 사용자 문서 및 시험 명세를 포함한 관련된 모든 작업산출물들이 포함되며, 조직 차원에서 제공된 형상 관리 라이브러리를 통해 등록되고, 관리된다.

☞ 베이스라인은 공식적인 검토를 거쳐 합의된 명세서 또는 작업산출물들의 집합으로서, 이후 개발의 근간이 되며, 변경 통제 절차에 의해서만 변경이 가능하다.

다음은 프로젝트 생명주기를 기반으로 일반적으로 설정되는 베이스라인의 예시이다.

- 기능 베이스라인 (Functional Baseline): 사용자의 요구사항이 분석되어 기술된 시스템 요구사항 명세서나, 이 문서를 검토하여 확정하는 시점을 의미한다.

- 할당 베이스라인 (Allocated Baseline): 사용자의 요구사항들이 하드웨어와 소프트웨어 사이에 어떻게 할당되는 가를 정의하는 시스템 설계명세서나, 이 문서를 검토하여 확정하는 시점을 의미한다.
- 설계 베이스라인 (Design Baseline): 코딩에 임하기 위한 설계명세서나, 이 문서를 검토하여 확정하는 시점을 의미한다.
- 시험 베이스라인 (Test Baseline): 소프트웨어의 기능 수행과 성능 충족도를 평가할 수 있는 소스 코드(source code), 실행 코드(execution code) 및 시험명세서를 포함한 시험계획서나, 이 문서를 검토하여 확정하는 시점을 의미한다.
- 제품 베이스라인 (Product Baseline): 하나의 시스템으로 개발 완료된 소프트웨어의 상품이나, 이에 대한 품질을 보증하는 시점을 의미한다.
- 운용 베이스라인 (Operational Baseline): 사용자 환경에 설치되고 운용되기 시작한 소프트웨어나, 이에 대한 품질을 사용자 입장에서 평가하는 시점을 의미한다.

☐ 산출물: 해당사항 없음

※ 베이스라인 설정 활동의 산출물은 명세서, 설계서, 소스 코드 등과 같이 해당 베이스라인을 구성하는 형상항목들을 지정된 형상관리 라이브러리에 등록된 상태 그 자체를 의미한다.

### 3.5. 형상항목 설명서 유지

각 형상항목에 대한 최신 설명서를 유지한다.

형상항목 설명서는 일반적으로 아래에 기술된 예시 등과 같은 정보를 포함한다.

- 항목 유형
- 관련된 형상관리 라이브러리 또는 파일 시스템
- 책임자
- 형상관리하에 들어온 일자
- 상태 정보 (예 개발 중, 베이스라인 설정됨, 배포됨 등)
- 변경 관리 기록
- 변경 이력

☐ 산출물: 해당사항 없음

※ 형상항목에 대한 설명서는 일반적으로 형상항목을 구성하는 한 부분으로서 존재한다. 예를 들면, 명세서나 설계서와 같은 문서의 경우 그 문서의 양식이 포함하고 있는 정보와 개정이력 등에 기술되며, 소스 코드의 경우 주로 헤더(header) 부분과 주석 문으로 기술된다.

### 3.6. 변경 및 릴리스 통제

항목을 기록하고, 등록하고, 릴리스하기 위한 체계를 구축한다.

☞ 소스 코드에 대한 통제에는 체크아웃/체크인, 파일 접근 승인, 버전 식별 및 증가, 변경에 대한 주석, 파일 잠금/해제(file locking/commit) 등이 포함될 수도 있다.

☞ 프로젝트 생명주기 전반에 걸쳐 베이스라인이 설정된 이후의 형상항목 변경은 통제된다. 변경을 통제한다는 것은 형상항목의 무결성 및 일관성을 유지하면서 변경을 처리하기 위해 체크아웃/체크인 과정이 필요하며, 베이스라인에 의도하지 않은 영향을 미치지 않게 하기 위해 관련 이해관계자와 검토를 수행한다. 또한, 변경된 형상항목이 베이스라인으로 통합되기 이전에 적절한 권한이 부여된 자로부터 승인을 득한다. 이러한 과정이 성공적으로 수행된 이후에 변경된 형상항목을 릴리스하며, 릴리스된 이후의 변경은 또다시 공식적으로 처리한다.

☞ 이러한 모든 활동들을 변경 및 릴리스 통제 활동이라 하며, 변경의 범위 및 특성에 따라 적절한 수준으로 조정될 수 있다. 예를 들면, 다른 형상항목에 영향을 미치지 않는 형상항목 변경에 대한 승인을 고려할 때에는 이 보다는 좀 더 덜 엄격해질 수도 있다.

☐ 산출물: [14-01]변경 이력

### 3.7. 형상항목 이력 유지

필요시, 이전 베이스라인에 놓인 버전으로 복귀할 수 있도록 충분하고 자세하게 각 형상항목의 이력(history)을 유지한다.

☞ 형상항목 이력은 최신 버전 이전의 구 버전들을 의미하며, 이러한 구 버전들을 어느 정도까지 유지해야 하는지에 대한 기준은 형상관리 계획서에 정의된 바에 따른다.

☞ 형상항목 이력은 형상 관리 라이브러리 내에서 별도의 지정된 공간에 저장되어

유지되며, 좀 더 나은 자동화된 형상관리 시스템을 활용하는 경우에는, 이러한 시스템의 일부 기능으로서 제공될 수도 있다.

☐ 산출물: 해당사항 없음

※ 형상항목 이력 유지 활동의 산출물은 지정된 형상관리 라이브러리에 저장된 상태 그 자체를 의미한다.

### 3.8. 형상 상태 보고

각 형상항목의 상태와 현 시스템 통합 관점에서의 형상항목들의 관계를 보고한다.

☞ 형상 상태 보고는 소프트웨어 형상 및 형상 관리의 식별, 통제, 감사 활동의 수행 결과를 기록하고, 유지하고, 보고하는 활동이다. 즉, 언제, 무슨 일이 프로젝트에 일어났는가를 상세히 기록하여 보고하는 것으로, 요약하여 정의하면 아래와 같다.

- 소프트웨어 개발과 관련된 중요한 사건(event)을 기록
- 소프트웨어 개발 중 수집된 모든 사건에 대한 완벽한 정보를 관리
- 소프트웨어 개발 과정에 대한 완벽한 보고서 작성 및 배포

☐ 산출물: [15-03]형상 상태 보고서

### 3.9. 형상항목 정보 검증

형상 상태 보고를 통해 제공된 형상항목 및 그 구조에 관한 정보의 완전함을 검증하고, 항목들의 일관성을 보장한다.

☞ 형상항목에 대한 정보 검증 활동은 일반적으로 형상 감사 활동의 형태로 수행되며, 형상 감사의 주요 내용은 아래에 기술한 예시와 같다.

- 베이스라인의 무결성을 평가
- 변경 이력, 형상 상태 보고서 등과 같은 형상관리 관련 기록이 형상항목의 형상을 올바르게 식별하고 있는지를 확인
- 형상 관리 라이브러리에 등록된 형상항목들의 구조와 무결성을 검토
- 형상 관리 라이브러리에 등록된 형상항목들의 완전성과 정확성을 확인
- 적용해야 할 형상관리 프로세스, 표준 및 절차 등에 부합하는지를 확인



형상항목 검증 결과 발견된 시정조치 사항들은 종결될 때까지 추적한다.

☐ 산출물: [15-24]감사 보고서, [13-19]검토 기록

### 3.10. 형상항목 백업, 저장, 파일보관, 취급 및 인도 관리

적절한 일정과 백업, 저장 및 파일보관을 위한 자원을 통해 형상항목의 무결성 및 일관성을 보장하고, 형상항목에 대한 취급 및 인도를 통제한다.

☞ 형상항목에 대한 백업/복구, 저장, 파일보관 등의 활동은 일반적으로 조직 차원에서 제공되는 형상 관리 라이브러리를 통해서 이루어진다. 좀 더 적극적인 방법으로는 자동화된 형상관리 시스템(configuration management system)을 사용하는 경우이며, 이 경우에는 이 조항에서 이야기하는 활동뿐만 아니라 변경 통제, 형상 상태 기록 등과 같이 형상관리에서 요구되는 많은 활동들이 이 시스템을 통해서 이루어질 수도 있다.

☐ 산출물: [13-03]백업/복구 기록, [13-06]인도 기록

## 4. 수행 결과

- 형상 관리 전략이 개발된다.
- 프로세스 또는 프로젝트에 의해 생성된 작업산출물/항목이 식별되고, 정의되고, 기준선에 놓인다.
- 작업산출물/항목의 수정 및 릴리스가 통제된다.
- 변경 및 릴리스가 영향 받는 당사자들에게 가용하게 된다.
- 작업산출물/항목의 상태와 수정사항이 기록되고 보고된다.
- 작업산출물/항목의 완전성 및 일관성이 보장된다.
- 작업산출물/항목의 저장, 처리 및 인도가 통제된다.

## 5. 측정 항목

측정항목(단위)	측정 방법
형상항목 변경요청 제안건수	베이스라인 설정 이후의 형상항목에 대한 변경이 요청된 건수를 측정한다.
형상항목 변경요청 승인건수	형상항목 변경요청 제안건수 중에서 변경이 승인된 건수를 측정한다.
형상항목 변경요청 완료건수	형상항목 변경요청 승인건수 중에서 변경이 완료된 건수를 측정한다.

(사용 예시)

- 형상항목 변경 승인율(%):  $(\sum \text{변경요청 승인건수} / \sum \text{변경요청 제안건수}) \times 100$
- 형상항목 변경 조치율(%):  $(\sum \text{변경요청 완료건수} / \sum \text{변경요청 승인건수}) \times 100$

## 6. 프로세스 조정

- "형상관리 담당자"는 프로젝트 규모, 특성 및 환경에 따라 보다 더 공식적인 수준의 "형상통제위원회(CCB: Configuration Control Board)"가 구성되어 그 역할을 수행할 수도 있다. 또한 약식화된 방법으로 특정 직원 또는 그룹에게 그 역할이 부여되지 않고 "개발자"들이 형상관리 역할을 나누어서 수행할 수도 있으며, 이 경우, "형상관리 담당자"는 역할을 부여 받은 각각의 개발자가 된다.
- "[08-04]형상관리 계획서"는 프로젝트 규모, 특성 및 환경에 따라 프로젝트 계획서에 통합하여 작성할 수 있다. 이 경우, "[08-12]프로젝트 계획서"에 "[08-04]형상관리 계획서"의 내용을 모두 포함하여 정의한다.
- "[14-01]변경 이력"은 형상항목을 변경할 때 그 형상항목의 내용에, 예를 들면, 명세서나 설계서와 같은 문서의 경우 그 문서가 포함하고 있는 개정이력에 기술될 수도 있으며, 소스 코드의 경우 주로 헤더(header) 부분과 주석 문으로 기술될 수도 있다. 따라서 이 경우, "[14-01]변경 이력" 작성을 생략할 수도 있다.
- "형상항목 정보 검증" 활동은 일반적으로 형상 감사 활동의 형태로 수행되나, 좀 더 약식화된 검토 또는 검증 활동의 형태로 수행될 수도 있다. 이 경우, "[15-24]감사 보고서" 작성을 생략하고, 대신에 "[13-19]검토 기록"으로 작성한다.

## 7. 참고 문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- CMMI-DEV(Capability Maturity Model Integration for Development), Version 1.2
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

# SUP.9 문제 해결 관리 프로세스

## 1. 개요

### 1.1. 목적

문제 해결 관리 프로세스의 목적은 발생된 모든 문제점들이 식별되고, 분석되고, 관리되고, 해결될 때까지 통제됨을 보장하는 것이다.

### 1.2. 설명

본 프로세스는 프로젝트의 착수단계부터 완료단계까지의 프로젝트 생명주기 전 과정에 걸쳐 적용된다.

문제 해결 관리 프로세스는 소프트웨어나 문서 등에서 발견된 문제점 등에 대해 문제를 식별하고 보고하는 것에서부터 조치가 종료될 때까지 추적하는 활동을 지원한다. 이러한 문제점들은 주로 시험 및 운영 단계에서 발견되는 소프트웨어에 대한 결함(defect)이나 오류(Error)들을 의미한다.

### 1.3. 주요 관련 프로세스

ID	프로세스	설명
SUP.1	품질 보증 프로세스	품질보증활동 결과 발생하는 부적합사항을 처리하기 위해, 변경 요청 관리 프로세스가 수행될 수도 있다.
SUP.8	형상 관리 프로세스	프로젝트 생명주기 동안에 발생하는 문제점을 조치하기 위해, 베이스라인이 설정된 소스 코드나 문서 등에 대한 변경과 관련하여 형상관리 프로세스가 수행된다.
SUP.10	변경 요청 관리 프로세스	프로젝트 생명주기 동안에 발견되는 결함이나 문제점들을 해결하기 위하여 요구사항을 조정하거나, 기타 변경에 대한 요청이 이루어질 수도 있다.
MAN.3-2	프로젝트 진행 관리 프로세스	프로젝트 진행 중에 식별되는 문제를 해결하기 하기 위해 문제 해결 관리 프로세스가 수행될 수도 있다.

## 1.4. 용어 정의

- 문제점(Problem): 소프트웨어에서 발견되는 결함(defect)이나 오류(error)로서, 주로 소프트웨어나 문서 등에서 요구사항을 충족시키지 못하는 기능이나 명세(specification)
- 이해관계자 (Stakeholders): 프로젝트에 직접 또는 간접적으로 관련이 있는 그룹 또는 개인
- 형상 관리 (Configuration Management; CM): 형상항목에 대한 기능적, 물리적 특성을 식별하여 문서화하고, 문서화된 특성에 대한 변경을 통제하고, 변경 과정 및 상태를 기록하여 보고하고, 규정된 요구사항에 부합함을 검증하기 위해 적용하는 기술적/행정적 관리 및 감독 활동과 관련된 엄격한 규율

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트 수행 중에 문제점이 발생되었다.
입력물	[13-16]변경요청서
수행활동	문제 해결 전략 수립 문제 식별 및 보고 초기 지원 및 문제 분류 문제 원인 조사 및 분석 문제 영향 평가 응급 조치 수행 문제 내용 통보 문제 조치 수행 변경 요청 착수 문제 조치 상태 추적
출력물	[08-27]문제관리 계획서 [13-07]문제 기록 [13-04]의사소통 기록 [13-16]변경요청서 [15-12]문제 상태 보고서
완료조건	발생된 문제점이 조치되었다.

## 2.2. 책임과 역할

역할 활동	프로젝트 관리책임자	개발자	조치담당자	이해관계자
1. 문제 해결 전략 수립	●	○		
2. 문제 식별 및 보고	○	●		
3. 초기 지원 및 문제 분류	●	○		
4. 문제 원인 조사 및 분석	○	○	●	○
5. 문제 영향 평가	○	○	●	○
6. 응급 조치 수행	○	●	○	○
7. 문제 내용 통보	○	○	●	○
8. 문제 조치 수행	○	○	●	○
9. 변경 요청 착수		○	●	
10. 문제 조치 상태 추적	●	○	○	○

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 문제 해결 전략 수립

문제점을 식별하고, 기록하고, 분석하고, 조치함을 보장하기 위해 문제 관리 전략을 수립한다.

☞ 문제 해결 전략은 일반적으로 아래에 기술한 예시 등과 같은 내용을 포함하며, 문제관리 계획서에 정의한다.

- 문제가 발견되었을 때부터 조치가 종료될 때까지의 문제 해결 절차
- 문제 분류 방법
- 문제 해결 접근방법
- 문제 추적 방법

☐ 산출물: [08-27]문제관리 계획서

### 3.2. 문제 식별 및 보고

발견된 문제점을 유일하게 식별하고, 기록하고, 보고한다.

☞ 문제점과 관련하여 문제점 보고자 관련 정보(성명, 조직, 연락처 등), 문제점 내용(가능한 한 상세하게 기술), 문제점 중요도[심각(critical), 주요(major), 사소(minor)], 문제점 보고일자 및 조치 희망일자, 문제점 관련 산출물 정보 등의 정보를 기록한다.

☐ 산출물: [13-07]문제 기록

### 3.3. 초기 지원 및 문제 분류

보고된 문제점에 대해 초기 지원 및 피드백(feedback)을 제공하고, 심각도(severity)에 따라 문제점을 분류한다.

☞ 보고된 문제점의 특성에 따라 초기 지원 및 피드백이 필요할 수도 있고, 아닐 수도 있으며, 문제점은 일반적으로 중요도, 시급성, 연관성 등에 따라 분류할 수 있다.

☞ 중요도는 심각(critical), 주요(major), 사소(minor)로 구분할 수 있으며, 문제점의 심각성 정도에 대한 정보를 제공한다. 시급성은 공식적인 문제점 조치를 위한 활동 수행 이전에 응급 조치가 필요한지, 아닌지에 대한 정보를 제공하며, 연관성은 이미 보고받은 문제점이 있는지, 타 이해관계자와 관련이 있는지에 대한 정보를 제공한다.

☐ 산출물: [13-07]문제 기록

### 3.4. 문제 원인 조사 및 분석

문제점의 원인을 식별하기 위해 문제점을 분석한다.

문제점은 알려진 오류이거나, 다수의 플랫폼에 설치된 어플리케이션에 영향을 미칠 수도 있다.

문제점의 원인이 되는 일반적인 사항들은 아래에 기술한 예시 등과 같다.

- 잘못된 요구사항
- 설계 오류
- 잘못 코딩(coding)된 소스 코드
- 부적절한 시험 명세
- 다른 문제점 조치에 대한 정보 부족

☐ 산출물: [13-07]문제 기록

### 3.5. 문제 영향 평가

문제점의 영향을 평가하고, 적절한 조치 활동을 결정한다.

☞ 문제 영향 평가는 산출물에 대한 무결성 및 일관성을 유지하기 위하여, 문제점을 해결하기 위해 조치가 필요한 모든 산출물을 사전에 식별하여 동시에 조치하기 위한 활동으로, 일반적으로 프로젝트 계획에 미치는 영향과 타 산출물에 미치는 영향 등 크게 두 가지 측면에서 평가가 이루어진다. 이를 구체적으로 기술하면 아래에 기술한 예시 등과 같다.

- 타 시스템 및 다른 구성요소에 미치는 영향
- 프로젝트 일정에 미치는 영향
- 추가 자원 및 예산의 필요성
- 관련 조직과의 관계에 있어, 동시 변경의 필요성 여부

☐ 산출물: [13-07]문제 기록

### 3.6. 응급 조치 수행

문제점에 대한 실제 변경을 통해 즉각적으로 해결이 필요한 경우, 적절한 조직 구성원에게 응급 조치를 위한 권한을 위임하고, 보고된 문제점에 대한 응급 조치 활동을 수행한다.

☞ 문제점은 분류된 시급성에 따라 완전한 조치를 위한 해결 활동 이전에 응급 조치가 필요할 수도 있다. 이러한 경우, 먼저 필요한 응급 조치 활동을 수행한 후, 완전한 문제점 해결 조치 활동을 수행한다.

☐ 산출물: 해당사항 없음

※ 응급 조치 수행의 산출물은 해당 산출물의 조치된 상태 그 자체를 의미한다.

### 3.7. 문제 내용 통보

문제점에 대한 심각도가 높고, 타 시스템 또는 사용자에게 영향을 미치는 경우, 문제점에 대한 내용 및 조치 내용을 관련 이해관계자에게 통보하여 문제점이 해결 또는 변경될 때까지 주의할 수 있도록 한다.

☞ 문제점은 분류된 심각도 및 연관성에 따라 타 이해관계자에게 해당 문제점에 대한 정보를 제공하여 또 다른 문제점이 발생하지 않도록 조치한다.

☐ 산출물: [13-04]의사소통 기록

### 3.8. 문제 조치 수행

문제점을 해결하기 위해 조치 활동을 수행하고, 수행 내용을 검토한다.

☞ 문제점 영향 평가 결과를 고려하여 사전에 결정된 문제점 조치 활동을 수행하고, 조치 결과를 관련 이해관계자와 함께 검토한다.

☐ 산출물: 해당사항 없음

※ 문제 조치 수행의 산출물은 해당 산출물의 조치된 상태 그 자체를 의미한다.

### 3.9. 변경 요청 착수

분석된 오류(errors)에 대한 변경 요청을 착수한다.

☞ 문제점 해결을 위해 기존의 요구사항에 대한 변경이 필요하거나, 설계 등의 작업산출물에 대한 변경이 필요한 경우, 그리고 내부적인 소프트웨어 개선을 위해 관련 작업산출물 변경 등이 필요한 경우에 변경 요청 관리 프로세스에 따라 변경 요청을 착수한다.

☐ 산출물: [13-16]변경요청서

### 3.10. 문제 상태 추적

식별된 문제점의 상태를 모니터링하고, 조치가 종료될 때까지 추적한다.



☞ 조치가 종료될 때까지 적절한 주기로 문제점 상태를 모니터링하고, 모니터링 결과는 관련 이해관계자와 함께 검토하여, 필요한 경우 조치에 필요한 지원을 제공할 수 있도록 한다.

☞ 산출물: [15-12]문제 상태 보고서

#### 4. 수행 결과

- 문제 관리 전략이 수립된다.
- 문제점이 식별되고, 기록되고, 분류된다.
- 수락할 수 있는 해결방법을 식별하기 위해 문제점이 분석되고, 평가된다.
- 문제 해결방법이 구현된다.
- 문제점이 종료될 때까지 추적된다.
- 보고된 모든 문제점에 대한 상태가 공유된다.

#### 5. 측정 항목

측정항목(단위)	측정 방법
문제 보고건수	보고된 문제점 건수를 측정한다.
문제 해결건수	보고된 문제점 건수 중에서 조치된 문제점 건수를 측정한다.

(사용 예시)

- 문제점 조치율(%):  $(\sum \text{문제 해결건수} / \sum \text{문제 보고건수}) \times 100$

#### 6. 프로세스 조정

- "[08-27]문제관리 계획서"는 프로젝트 규모, 특성 및 환경에 따라 "[08-12]프로젝트 계획서"에 통합하여 작성할 수 있다. 이 경우, "[08-12]프로젝트 계획서"에 "[08-27]문제 관리 계획서"의 내용을 모두 포함하여 정의한다.
- "응급 조치 수행" 활동은 보고된 문제점의 특성, 즉 분류된 시급성에 따라 수행될 수도 있고, 생략될 수도 있다.
- "변경 요청 착수" 활동은 보고된 문제점을 해결하기 위해 기본 요구사항 또는 설계 변경 등의 조치가 필요한 경우 수행된다. 그렇지 않은 경우에는 생략한다.

## 7. 참고 문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

# SUP.10 변경 요청 관리 프로세스

## 1. 개요

### 1.1. 목적

변경 요청 관리 프로세스의 목적은 변경 요청이 관리되고, 추적되고, 통제됨을 보장하는 것이다.

### 1.2. 설명

본 프로세스는 프로젝트의 착수단계부터 완료단계까지의 프로젝트 생명주기 전 과정에 걸쳐 적용된다.

변경 요청 관리 프로세스는 새로운 요구사항이 발생되었거나 기존의 요구사항이 변경된 경우로 인한 설계 등의 작업산출물 변경, 그리고 내부적인 소프트웨어 개선을 위한 변경 요구가 발생 했을 경우에 관련 작업산출물 변경 등에 대해 변경 요청에서부터 변경이 완료될 때까지 추적하는 활동을 지원한다.

### 1.3. 주요 관련 프로세스

ID	프로세스	설명
SUP.1	품질 보증 프로세스	품질보증활동 결과 발생하는 부적합사항을 처리하기 위해, 변경 요청 관리 프로세스가 수행된다.
SUP.8	형상 관리 프로세스	프로젝트 생명주기 동안에 발생하는 새로운 요구사항이나 요구사항 변경, 설계 변경, 시스템 개선등과 관련하여, 베이스라인이 설정된 소스 코드나 문서 등을 변경하기 위해 형상관리 프로세스가 수행된다.
SUP.9	문제 해결 관리 프로세스	프로젝트 생명주기 동안에 발견되는 결함이나 문제점들을 해결하기 위하여 요구사항을 조정하거나, 기타 변경에 대한 요청이 이루어질 수도 있다.
MAN.3-2	프로젝트 진행 관리 프로세스	프로젝트 진행 중에 식별되는 문제를 해결하기 하기 위해 요구사항을 조정하거나, 기타 변경에 대한 요청이 이루어질 수도 있다.

## 1.4. 용어 정의

- 릴리스 (Release): 승인된 버전(version)의 공식적인 통지 및 배포
- 버전 (Version): 형상항목에 대한 초기 릴리스 또는 재 릴리스
- 이해관계자 (Stakeholders): 프로젝트에 직접 또는 간접적으로 관련이 있는 그룹 또는 개인
- 형상 관리 (Configuration Management; CM): 형상항목에 대한 기능적, 물리적 특성을 식별하여 문서화하고, 문서화된 특성에 대한 변경을 통제하고, 변경 과정 및 상태를 기록하여 보고하고, 규정된 요구사항에 부합함을 검증하기 위해 적용하는 기술적/행정적 관리 및 감독 활동과 관련된 엄격한 규율

## 2. 프로세스 구성

### 2.1. 프로세스 요약

구 성	내 용
선행조건	프로젝트가 착수 되었다. 형상관리 담당자가 선임되었다.
입력물	[08-04]형상관리 계획서 [16-03]형상 관리 라이브러리
수행활동	변경 관리 전략 수립 변경 요청 기록 변경 요청 상태 기록 타 변경 요청과의 의존성 및 연관성 정의 변경 영향 평가 변경 결과 검증 및 확인 활동 식별 변경 승인 변경 구현 변경 결과 검토
출력물	[08-28]변경관리 계획서 [13-16]변경요청서 [13-21]변경관리기록
완료조건	요청된 변경이 구현되었다.

## 2.2. 책임과 역할

역할	형상관리 담당자	개발자	프로젝트 관리책임자	이해관계자
3.1 변경 관리 전략 수립	●	○	○	
3.2 변경 요청 기록		●		
3.3 변경 요청 상태 기록	●	○		
3.4 타 변경 요청과의 의존성 및 연관성 정의	●	○	○	○
3.5 변경 영향 평가	●	○	○	○
3.6 변경 결과 검증 및 확인 활동 식별	○	●		
3.7 변경 승인	○	○	●	
3.8 변경 구현	○	●	○	
3.9 변경 결과 검토	●	○	○	○

(●: 주관, ○: 지원)

## 3. 프로세스 활동

### 3.1. 변경 관리 전략 수립

변경을 기술하고, 기록하고, 분석하고, 조치함을 보장하기 위하여 변경관리 전략을 수립하고, 이행한다.

☞ 변경 관리 전략은 일반적으로 아래에 기술한 예시 등과 같은 내용을 포함하며, 변경관리 계획서에 정의한다.

- 변경이 요청되었을 때부터 변경이 종료될 때까지의 변경 통제 절차
- 변경 상태를 추적하는 방법
- 변경에 대한 승인권자
- 변경이 완료되었을 때 검증하고 확인하는 활동에 대한 정의

☐ 산출물: [08-28]변경관리 계획서

### 3.2. 변경 요청 기록

각각의 변경 요청 사항들을 유일하게 식별하고, 기록한다.

변경 요청과 관련하여 기록되어야 하는 정보들은 아래에 기술한 예시 등과 같다.

- 변경 요청자 관련 정보 (성명, 조직, 연락처 등)
- 변경 요청 대상 산출물 정보
- 변경 요청 내역 (가능한 한 상세하게 기술)
- 변경 요청일자 및 조치 희망일자

☐ 산출물: [13-16]변경요청서

### 3.3. 변경 요청 상태 기록

변경에 대한 추적을 용이하게 하기 위해 변경 요청 및 변경에 대한 상태를 기록한다.

☐ 변경 요청에 대한 원인(배경)까지의 추적성을 제공할 수 있도록 기록한다. 예를 들면, 문제에 대한 해결 또는 결함 보고로 제출된 변경 요청은 원인이 된 문제(또는 결함) 기록과 연결될 수 있도록 그 연관성을 유지한다.

☐ 산출물: [13-16]변경요청서

### 3.4. 타 변경 요청과의 의존성 및 연관성 정의

의존성 정의(예를 들면, 동일한 소프트웨어 구성요소에 대한 다른 변경이 발생되었거나, 또는 계획된 릴리스와 관련된 여러 변경의 발생 등)를 위해 변경 요청과 타 변경 요청과의 연관성을 식별한다.

이 활동의 목적은 아래에 기술된 예시 등과 같은 사항을 사전에 식별하여 산출물에 대한 무결성과 일관성을 유지하기 위함이다.

- 중복된 변경 요청이 있다면 이를 그룹화 하여 동시에 처리
- 상충된 변경 요청이 있다면 이에 대한 문제를 해결한 후 처리
- 타 변경 요청과 상호 의존적인 관계가 있다면 우선순위를 설정하여 순차적으로 처리

☐ 산출물: [13-16]변경요청서

### 3.5. 변경 영향 평가

변경 요청에 대한 영향, 자원, 위험 및 잠재적인 이득 등을 평가하고, 구현을 확인하기 위한 기준을 정의한다.

☞ 변경요청위원회(Change Request Board; CRB)는 변경 요청을 평가하기 위해 활용되는 일반적인 방법이다. 영향 및 자원을 평가할 때에는 비용, 투입공수 및 가용성, 구현을 위해 필요한 시간 등과 같은 사항들을 포함하여, 변경을 구현하는 데 필요한 자원과 함께, 기반 구조(infrastructure) 및 사용자에게 대한 영향도 고려되어야 한다.

☞ 변경 영향 평가는 일반적으로 프로젝트 계획에 미치는 영향과 타 산출물에 미치는 영향 등 크게 두 가지 측면에서 평가가 이루어진다. 이를 구체적으로 기술하면 아래에 기술한 예시 등과 같다.

- 변경 요청 사유 및 변경 대상 범위의 타당성
- 타 시스템 및 다른 구성요소에 미치는 영향
- 프로젝트 일정에 미치는 영향
- 추가 자원 및 예산의 필요성
- 관련 조직과의 관계에 있어, 동시 변경의 필요성 여부

☐ 산출물: [13-16]변경요청서

### 3.6. 변경 결과 검증 및 확인 활동 식별

변경을 구현하기 이전에, 수행되어야 할 검증 및 확인 활동의 범위를 식별한다.

☞ 변경 결과에 대한 검증 및 확인 활동은 프로젝트 진척도에 따라 그 범위가 결정된다. 예를 들면, 변경 범위에 따라서 달라지겠지만 시험단계에서 요구사항 변경이 요청되었다면 일반적으로 요구명세 검토, 설계 검토, 코드 검사, 시험명세 검증, 시험 등의 여러 가지 다양한 검증 및 확인 활동이 필요하지만, 설계 단계에서 요구사항 변경이 요청되었다면 요구명세 검토만의 검증 및 확인 활동이 필요하다.

☐ 산출물: [13-16]변경요청서

### 3.7. 변경 승인

변경을 구현하기 이전에, 모든 변경을 승인한다.

☞ 변경에 대한 승인권자는 변경의 영향에 따라 다양한 수준으로 정할 수 있다. 예를 들면, 특정 프로젝트 내부에만 영향을 미치는 변경이라면 프로젝트 관리자의 승인을 득하면 되겠지만, 만약 타 프로젝트까지 영향을 미치는 변경이라면 그 프로그램 관리자의 승인까지 득해야 할 것이다. 또한, 프로젝트 일정이나 비용에 영향을 미치는 변경이라면 고객이나 상위관리자의 승인을 득하거나, 이러한 이해관계자들이 조직되어 있는 특정 회의체(예를 들면, 형상통제위원회 등)의 승인을 득해야 할 것이다. 이러한 사항은 변경관리 계획서에 정의한다.

☐ 산출물: [13-16]변경요청서

### 3.8. 변경 구현

승인된 변경을 구현한다.

☞ 계획된 변경은 목표 릴리스(target release)에 포함할 수도 있다. 패키징된 릴리스(packaged release)는 시정(corrective) 및 적응(adaptive)을 위한 변경을 포함할 수도 있다.

☞ 변경 요청서에 기록된 변경대상 산출물을 변경한다. 프로젝트 진척도에 따라서 어떠한 경우에는 요구명세만을 변경하는 경우도 있겠지만, 또다른 경우에는 요구명세, 설계명세, 소스 코드, 시험명세 등의 여러 가지 산출물을 변경해야 하는 경우도 있다.

☐ 산출물: 해당사항 없음

※ 변경 구현 활동의 산출물은 변경 대상 산출물의 변경된 상태 그 자체를 의미한다.

### 3.9. 변경 결과 검토

모든 변경이 기대하는 효과 및 효과를 달성하였음을 보장하기 위하여 변경 종료 직전에 모든 변경을 검토한다.

☞ 사전에 식별된 검증 및 확인 방법에 따라 검증 및 확인 활동을 수행하며, 변경 결과와 변경에 대한 검증 및 확인 결과를 관련 이해관계자와 함께 검토한다. 검토 결과에 따라 "3.8 변경 구현" 활동이 다시 수행될 수도 있다.

☐ 산출물: [13-21]변경관리기록



#### 4. 수행 결과

- 변경 관리 전략이 수립된다.
- 변경 요청이 기록되고, 식별된다.
- 타 변경 요청과의 의존성 및 연관성이 식별된다.
- 변경 요청 구현을 확인하기 위한 기준이 정의된다.
- 변경 요청의 우선순위가 정해지고, 자원 요구사항이 추정된다.
- 우선순위 및 자원 가용성을 기초로 변경이 승인된다.
- 승인된 변경이 구현되고, 종료 시까지 추적된다.
- 모든 변경 요청의 상태가 공유된다.

#### 5. 측정 항목

측정항목(단위)	측정 방법
변경요청 건수	요청된 변경 건수를 측정한다.
변경승인 건수	요청된 변경 건수 중에서 승인된 변경 건수를 측정한다.
변경조치 건수	승인된 변경 건수 중에서 조치된 변경 건수를 측정한다.

(사용 예시)

- 변경요청 승인율(%):  $(\sum \text{변경승인 건수} / \sum \text{변경요청 건수}) \times 100$
- 변경조치 완료율(%):  $(\sum \text{변경조치 건수} / \sum \text{변경승인 건수}) \times 100$

#### 6. 프로세스 조정

- "[08-28]변경관리 계획서"는 프로젝트 규모, 특성 및 환경에 따라 "[08-04]형상 관리 계획서" 또는 "[08-12]프로젝트 계획서"에 통합하여 작성할 수 있다. 이 경우, 해당 계획서에 "[08-28]변경관리 계획서"의 내용을 모두 포함하여 정의한다.

#### 7. 참고 문헌

- KSXISO/IEC15504-2 정보기술 - 소프트웨어 프로세스 심사 - 제2부 : 프로세스 및 프로세스 능력 참조 모델, 2006
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

## 부속서 A. 작업산출물 특성

이 부속서에 나열된 작업산출물 특성은 프로세스 구현에 따른 예상되는 입력물 및 출력물을 검토할 때 사용될 수 있도록 첨부되었다. 작업산출물 특성은 특정 프로세스의 심사를 지원하고 객관적인 증거를 제시하기 위해, 작업산출물 내에서 찾아야 하는 속성들에 대한 지침으로서 제공된다.

작업산출물 ID	작업산출물을 참조하기 위해 사용되는 작업산출물의 식별자 번호 (예: 01-03)
작업산출물 이름	작업산출물 특성과 관련된 전형적인 이름의 예를 제공한다. 이 이름은 프랙티스나 프로세스에서 나오는 작업산출물 유형에 대한 식별자로 제공되나 작업산출물들은 조직에 따라 다른 이름으로 불릴 수 있으므로 작업산출물의 이름은 중요하지 않다. 어떤 한 작업산출물에서 정의된 특성은 조직에 따라 여러 개의 작업산출물에 포함될 수도 있다. 또한 작업산출물의 양식은 다를 수 있다. 여기에서 주어진 예제들과 조직의 실제 작업산출물을 대응시키는 것은 심사원과 심사대상 조직단위(OU)의 조정자가 할 일이다.
작업산출물 특성	작업산출물 유형과 관련된 특성으로 여길 수 있는 예를 제공한다. 심사원은 심사대상 조직단위(OU)가 제공하는 문서에서 작업산출물 특성들을 찾을 수 있다.

부속서 A는 A.1(공통 작업산출물)과 A.2 (공통작업 산출물, 특정 작업산출물)로 구성되어 있다. 특정 작업산출물이란 본문에서 작업산출물이라고 명명한 기본 프랙티스 수행과 관련된 입출력물이다.

## A.1 공통 작업산출물

아래의 표는 프로세스 속성인 PA를 달성한 결과인 공통 작업산출물에 나타날 것이라 기대되는 특성들을 나타내고 있다. 공통 작업산출물은 프로세스 수행지표인 작업산출물들을 분류하는 기반이 된다. 즉, 공통 작업산출물에는 - 이하가 모두00 번호가 부여되며(예 03-00), 모든 프로세스 수행지표인 작업산출물은 공통 작업산출물의 00번호 자리에 번호를 부여한다.

작업 산출물 ID	작업산출물 이름	작업산출물 특성
03-00	데이터	<ul style="list-style-type: none"> <li>- 측도를 적용한 결과</li> <li>- 정의된 기간 내에 알 필요가 있는 사람에게 가용해야 함</li> </ul>
07-00	측도	<ul style="list-style-type: none"> <li>- 제품이나 프로세스에 대한 정량적이거나 정성적인 속성</li> <li>- 데이터 수집을 위한 방법을 정의함</li> <li>- 사용자들이 이해할 수 있어야 함</li> <li>- 조직/프로젝트에 가치를 제공함</li> <li>- 관련된 목표를 참조함</li> <li>- 작업 흐름과 일관성이 있음</li> <li>- 조직, 생명주기모델, 프로세스에 적절함</li> <li>- 사용자들이 의미있는 해석을 할 수 있도록 적절한 분석과 주석을 가짐</li> </ul>
08-00	계획서	<p>(응용이나 목적에 적합한)</p> <ul style="list-style-type: none"> <li>- 계획의 주체를 식별함, 다음사항을 포함함: <ul style="list-style-type: none"> <li>o 달성되어야 할 것들에 대한 범위와 목적</li> <li>o 가정사항</li> <li>o 제약사항들</li> <li>o 위험</li> <li>o 수행되어야 할 과업</li> <li>o 일정, 이정표와 목표 날짜</li> <li>o 주요한 의존관계</li> <li>o 계획을 위한 유지관리 배치</li> </ul> </li> <li>- 계획 달성을 위한 방법/접근법, 다음 사항을 식별함: <ul style="list-style-type: none"> <li>o 다른 당사자(예, 공급자, 고객)들이 수행해야 할 과업을 포함한 과업 책임자</li> <li>o 품질 기준</li> <li>o 요구된 작업산출물</li> </ul> </li> <li>- 계획 목적을 달성하기 자원을 포함함. <ul style="list-style-type: none"> <li>o 일정</li> <li>o 직원(주요 역할과 권한, 예, 후원자)</li> <li>o 자원과 장비</li> <li>o 예산</li> </ul> </li> <li>- 미 달성된 과업을 위한 임시계획을 포함</li> <li>- 계획이 승인됨</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
09-00	정책	<ul style="list-style-type: none"> <li>- 권한을 부여 받음</li> <li>- 정책의 영향을 받는 모든 사람들에게 사용가능함</li> <li>- 준수해야 할 프랙티스/규칙 수립</li> </ul>
10-00	프로세스 기술서	<ul style="list-style-type: none"> <li>- 프로세스와 절차에 대한 자세한 기술서는 다음 사항을 포함: <ul style="list-style-type: none"> <li>o 표준 프로세스의 조정(적용가능한)</li> <li>o 프로세스 목적</li> <li>o 프로세스 수행결과</li> <li>o 수행되어야 할 과업, 활동 및 과업의 순서</li> <li>o 과업 활동들 사이의 주요한 의존관계</li> <li>o 과업을 수행하기위해 필요한 예상시간</li> <li>o 입력/출력 작업산출물</li> <li>o 작업 입력물과 출력물 사이의 연결</li> </ul> </li> <li>- 프로세스 시작과 종료 기준을 식별함</li> <li>- 프로세스에 대한 내외부 인터페이스를 식별함</li> <li>- 프로세스 측도를 식별함</li> <li>- 기대 품질을 식별함</li> <li>- 기능별 역할과 책임을 식별함</li> <li>- 책임있는 사람에 의해 승인됨</li> </ul>
13-00	기록	<ul style="list-style-type: none"> <li>- 달성된 결과를 설명하거나 프로세스에서 수행된 활동에 대한 증거를 보여주는 작업산출물</li> <li>- 검색 및 식별가능한 데이터 집합의 일부인 항목</li> </ul>
14-00	등록	<ul style="list-style-type: none"> <li>- 정의된 순서에 따라 아래와 같은 것들이 가능하도록 하는 데이터와 정보의 편집 <ul style="list-style-type: none"> <li>o 활동들이 수행되고 있다는 증거를 보여주는 전반적인 관점</li> <li>o 감시와 분석</li> <li>o 지속적으로 프로세스가 수행되고 있다는 증거를 제공함</li> </ul> </li> </ul>
15-00	보고서	<ul style="list-style-type: none"> <li>- 다음과 같은 상황을 설명하는 작업산출물: <ul style="list-style-type: none"> <li>o 결과와 상태를 포함함</li> <li>o 적용가능하고 연관된 정보를 식별함</li> <li>o 고려사항과 제약사항을 식별함</li> <li>o 증거와 확인 데이터를 제공함</li> </ul> </li> </ul>
16-00	저장소	<ul style="list-style-type: none"> <li>- 구성요소에 대한 저장소</li> <li>- 저장과 검색가능한 능력</li> <li>- 내용을 검색할 수 있는 능력</li> <li>- 속성 설명을 포함한 내용물을 목록화함</li> <li>- 관련 그룹들 사이에 구성요소를 이전하고 및 공유함</li> <li>- 효율적인 접근 통제</li> <li>- 구성요소 설명을 유지관리함</li> <li>- 구성요소의 저장버전의 복구</li> <li>- 구성요소의 상태를 보고하는 능력</li> <li>- 구성요소의 변경은 변경/사용자 요청까지 추적 가능함</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
18-00	표준	<ul style="list-style-type: none"> <li>- 적용 대상 및 사람을 식별함:</li> <li>- 구성요소에 대한 기대값을 식별함</li> <li>- 요구사항에 대한 적합성을 제시함</li> <li>- 요구사항에 대한 예외사항 처리나 조정 방안을 포함</li> </ul>
19-00	전략	<ul style="list-style-type: none"> <li>- 만족시켜야 하는 니즈, 목적과 목표를 식별함</li> <li>- 니즈, 목표, 목적을 달성시키기 위한 선택과 접근방법을 수립함</li> <li>- 어떠한 전략적 선택에 대해 평가되는 평가 기준을 수립함</li> <li>- 다루어져야 할 제약사항이나 위험을 식별함</li> </ul>
20-00	템플릿	<ul style="list-style-type: none"> <li>- 프로세스 실행결과로서 생성된 작업산출물과 관련된 속성들을 정의함</li> <li>- 이 제품 유형과 전형적으로 관련된 기술적 구성요소를 식별함</li> <li>- 예상되는 형식과 스타일을 정의함</li> </ul>
21-00	작업산출물	<ul style="list-style-type: none"> <li>- 프로세스 실행에서 나온 산출물과 관련된 속성을 정의함 <ul style="list-style-type: none"> <li>o 작업산출물에서 나타나는 주요한 구성요소</li> </ul> </li> </ul>

## A.2 공통 및 고유 작업산출물

아래의 표는 공통 및 고유 작업산출물을 나타내고 있다. 아래의 고유 작업산출물의 유형들은 특정 프로세스의 수행결과를 만족시키기 위해 만들어지고 적용된다.

작업 산출물 ID	작업산출물 이름	작업산출물 특성
01-03	소프트웨어 항목	<ul style="list-style-type: none"> <li>- 통합 소프트웨어의 구성요소:               <ul style="list-style-type: none"> <li>o 소스 코드</li> <li>o 소프트웨어 컴포넌트</li> <li>o 실행 코드</li> <li>o 형상 파일</li> </ul> </li> <li>- 다음과 같은 문서화               <ul style="list-style-type: none"> <li>o 원시 코드를 기술하고 정의함</li> <li>o 소프트웨어 구성요소를 기술하고 정의함</li> <li>o 형상 파일을 기술하고 식별함</li> <li>o 실행코드를 기술하고 식별함</li> <li>o 소프트웨어 생명주기 상태 기술함</li> <li>o 저장소와 배포 기준을 기술함</li> <li>o 소프트웨어 단위의 컴파일을 설명함</li> <li>o 소프트웨어 항목 구축을 설명함</li> </ul> </li> </ul>
02-00	계약(서)	<ul style="list-style-type: none"> <li>- 구매 혹은 인도하여야 할 것을 정의함</li> <li>- 인도 혹은 계약된 서비스 일자를 위한 기간을 식별함</li> <li>- 법적 요구사항을 식별함</li> <li>- 재정적 고려사항을 식별함</li> <li>- 품질 보증 정보를 식별함</li> <li>- 저작권 및 라이선싱 정보를 식별함(예, 모든 관련 작업산출물에 대한 특허, 저작권, 비밀보장, 소유권, 라이선스 권리)</li> <li>- 고객 서비스 요구사항을 식별함</li> <li>- 서비스 수준 요구사항을 식별함</li> <li>- 모든 성능 및 품질 기대/ 제약사항/ 감시를 참조함</li> <li>- 사용되는 표준과 절차</li> <li>- 권한이 있는 사람의 검토 및 승인에 대한 증거</li> <li>- 적절한 계약을 위해서는 아래의 사항들이 고려되어야 함               <ul style="list-style-type: none"> <li>o 모든 인수 기준을 참조함</li> <li>o 고객의 모든 특별한 니즈를 참조함(예, 비밀협약 요구사항, 보안, 하드웨어 등)</li> <li>o 모든 변경관리 및 문제해결 절차를 참조함</li> <li>o 독립적 사업자 및 협력업체와 인터페이스를 식별함</li> <li>o 개발 및 유지보수 프로세스에서 고객의 역할을 식별함</li> <li>o 고객이 제공해야 할 자원을 식별함</li> </ul> </li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
03-00	데이터	<ul style="list-style-type: none"> <li>- 측도를 적용한 결과</li> <li>- 정의된 기간 내에 알 필요가 있는 사람에게 가용해야 함</li> </ul>
03-07	시험데이터	<ul style="list-style-type: none"> <li>- 시험 사례 및 절차와 관련된 입력 값들의 집합</li> <li>- 타당성의 범위를 지정함</li> <li>- 관련된 시험 사례를 식별함</li> </ul>
04-04	상위수준 소프트웨어 설계	<ul style="list-style-type: none"> <li>- 전체적인 소프트웨어 구조를 기술함</li> <li>- 필요한 소프트웨어 구성요소를 식별함</li> <li>- 소프트웨어 구성요소간의 관계를 식별함</li> <li>- 고려 사항들: <ul style="list-style-type: none"> <li>o 요구된 소프트웨어 성능 특성</li> <li>o 요구된 소프트웨어 인터페이스</li> <li>o 요구된 소프트웨어 보안 특성</li> <li>o 데이터베이스 설계 요구사항</li> <li>o 오류 대처법 및 회복 속성</li> </ul> </li> </ul>
04-05	하위수준 소프트웨어 설계	<ul style="list-style-type: none"> <li>- 상세 설계(프로토타입, 데이터흐름도, 개체관계도, pseudo 코드로 표현됨)를 제공함</li> <li>- 데이터 입/출력 형태를 제공함</li> <li>- 데이터 저장 니즈의 명세서를 제공함</li> <li>- 필요한 데이터 이름 부여 규칙을 수립함</li> <li>- 필요한 데이터 구조를 정의함</li> <li>- 필요한 각 데이터 구성요소들의 필드와 목적을 정의함</li> <li>- 프로그램 구조 명세서를 제공함</li> </ul>
04-06	시스템 아키텍처 설계	<ul style="list-style-type: none"> <li>- 모든 시스템 설계의 개요를 제공함</li> <li>- 시스템 구성요소간 내부관계를 기술함</li> <li>- 시스템 구성요소와 소프트웨어 사이의 관계를 기술함</li> <li>- 요구된 각 시스템 구성요소를 위한 설계를 명시, 고려사항은 다음과 같음: <ul style="list-style-type: none"> <li>o 메모리/능력 요구사항</li> <li>o 하드웨어 인터페이스 요구사항</li> <li>o 사용자 인터페이스 요구사항</li> <li>o 외부 시스템 인터페이스 요구사항</li> <li>o 성능 요구사항</li> <li>o 명령 구조</li> <li>o 보안/ 데이터 보호 특성</li> <li>o 시스템 매개변수 설정</li> <li>o 수동 운용</li> <li>o 재사용 컴포넌트</li> </ul> </li> <li>- 요구사항을 시스템 구성요소와 대응시킴</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
06-03	설치가이드	<ul style="list-style-type: none"> <li>- 실행 요구사항에 의해 순서에 따라 지시된 제품 로딩 및 설치를 위한 과업 <ul style="list-style-type: none"> <li>o 인도된 파일로부터 소프트웨어 다운로드</li> <li>o 소프트웨어, 파일, 폴더, 라이브러리에 대한 적절한 업로드</li> <li>o 적용가능한 부분의 부분 설치 및 업그레이드 방법</li> <li>o 초기화 절차</li> <li>o 전환 절차</li> <li>o 커스트마이징 및 형상 절차</li> <li>o 검증 절차</li> <li>o 업데이트 절차</li> <li>o 운영 설명서</li> </ul> </li> <li>- 식별된 운영 요구사항 <ul style="list-style-type: none"> <li>o 관련된 하드웨어, 소프트웨어, 고객 문서</li> <li>o 전환 프로그램 및 사용법</li> <li>o 프로그램 초기화 및 시스템 가동 정보</li> <li>o 구성요소와 설명서</li> <li>o 요구된 하드웨어와 소프트웨어의 최소 형상</li> <li>o 백업과 복구 설명서</li> <li>o 확인 프로그램</li> <li>o 형상 매개변수(예, 크기 요구사항, 메모리)</li> </ul> </li> <li>- 고객/기술 지원 계약</li> <li>- 문제해결 설명서</li> <li>- 롤백 계획</li> </ul>
07-00	측도	<ul style="list-style-type: none"> <li>- 제품이나 프로세스에 대한 정량적이거나 정성적인 속성</li> <li>- 데이터 수집을 위한 방법을 정의함</li> <li>- 사용자들이 이해할 수 있어야 함</li> <li>- 조직/프로젝트에 가치를 제공함</li> <li>- 관련된 목표를 참조함</li> <li>- 작업 흐름과 일관성이 있음</li> <li>- 조직, 생명주기모델, 프로세스에 적절함</li> <li>- 사용자들이 의미있는 해석을 할 수 있도록 적절한 분석과 주석을 가짐</li> </ul>
08-00	계획서	<p>(응용이나 목적에 적합한)</p> <ul style="list-style-type: none"> <li>- 계획의 주체를 식별함</li> <li>- 다음사항을 포함함: <ul style="list-style-type: none"> <li>o 달성되어야 할 것들에 대한 범위와 목적</li> <li>o 가정사항</li> <li>o 제약사항들</li> <li>o 위험</li> <li>o 수행되어야 할 과업</li> <li>o 일정, 이정표와 목표 날짜</li> </ul> </li> </ul>



작업 산출물 ID	작업산출물 이름	작업산출물 특성
		<ul style="list-style-type: none"> <li>○ 주요한 의존관계</li> <li>○ 계획을 위한 유지관리 배치</li> <li>- 계획 달성을 위한 방법/접근법</li> <li>- 다음 사항을 식별함: <ul style="list-style-type: none"> <li>○ 다른 당사자(예, 공급자, 고객)들이 수행해야 할 과업을 포함한 과업 책임자</li> <li>○ 품질 기준</li> <li>○ 요구된 작업산출물</li> </ul> </li> <li>- 계획 목적을 달성하기 자원을 포함함. <ul style="list-style-type: none"> <li>○ 일정</li> <li>○ 직원(주요 역할과 권한, 예, 후원자)</li> <li>○ 자원과 장비</li> <li>○ 예산</li> </ul> </li> <li>- 미 달성된 과업을 위한 임시계획을 포함</li> <li>- 계획이 승인됨</li> </ul>
08-04	형상 관리 계획서	<ul style="list-style-type: none"> <li>- 형상 항목들에 대한 변경을 통제하기 위한 절차를 참조하거나 정의함</li> <li>- 형상관리 활동의 상태를 결정하기 위해 활용된 측정 활동을 정의함</li> <li>- 형상관리 감사기준을 정의함</li> <li>- 형상관리 기능이 승인됨</li> <li>- 형상 라이브러리 도구, 절차 및 수단을 식별함</li> <li>- 통제 항목의 이력과 상태를 보여주는 관리 기록과 상태 보고서를 포함함</li> <li>- 형상관리 라이브러리를 위한 접근 절차와 수단의 위치를 명세화함</li> <li>- 저장, 조작 및 인도(저장과 검색을 포함한)을 위해 정의된 절차와 수단</li> </ul>
08-06	프로젝트 활동도	<ul style="list-style-type: none"> <li>- 모든 프로젝트의 활동, 속성 및 그들간의 관계를 보여주는 네트워크 다이어그램으로서 프로젝트의 도식법 가장 일반적인 형태가 PERT 차트임.</li> <li>- 활동 속성은 포함함: <ul style="list-style-type: none"> <li>○ 활동 이름</li> <li>○ 예상 기간</li> <li>○ 계획된 시작 날짜와 실제 시작 날짜</li> <li>○ 계획된 날짜와 실제 완성 날짜</li> <li>○ 자원 요구사항</li> </ul> </li> <li>- 활동들간의 관계를 포함함: <ul style="list-style-type: none"> <li>○ 선행 활동</li> <li>○ 후속 활동</li> <li>○ 의존적 지연</li> </ul> </li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
08-09	설치 및 유지보수 계획	<ul style="list-style-type: none"> <li>- 영향을 받는 장소를 식별함</li> <li>- 적절한 버전 정보와 함께 설치를 위한 요구된 구성요소를 식별함 (최소한 다음 사항은 고려되어야 함) <ul style="list-style-type: none"> <li>o 릴리스된 소프트웨어</li> <li>o 매체 유형</li> <li>o 요구된 유지관리 수정사항</li> <li>o 요구된(변환 프로그램, 확인절차, 결합시스템 인터페이스, 데이터 베이스 관리 시스템) 소프트웨어 지원</li> <li>o 요구된 고객 문서</li> <li>o 설치 지침(서)</li> <li>o 요구된 하드웨어 및 주변 장비</li> </ul> </li> <li>- 지원 정보 혹은 필요한 자료를 식별함: <ul style="list-style-type: none"> <li>o 파라미터 정보</li> <li>o 운영 및 유지관리 정보</li> <li>o 사전 전환 정보, 재료 및 설치된 장비</li> </ul> </li> <li>- 설치 유형(새 제품, 현 시스템의 전환, 유지관리)</li> <li>- 마스터 및 백업 복사본의 보관</li> <li>- 운영 및 운영 정지 의사결정 기준을 식별함</li> <li>- 검증 프로세스를 식별함: <ul style="list-style-type: none"> <li>o 요구된 인도물을 준비하기 위해 필요한 과업 검증</li> <li>o 사이트에서 요구한 컴포넌트의 검증</li> <li>o 설치 절차의 검증</li> <li>o 사전 설치 구조와 전환 활동의 검증</li> <li>o 시스템 통합, 릴리스 구축 등의 검증</li> </ul> </li> <li>- 고객 인수 요구사항을 식별함</li> <li>- 안전과 보안 요구사항을 식별함</li> </ul>
08-10	소프트웨어 통합 시험 계획서	<ul style="list-style-type: none"> <li>- 통합 목적을 정의함: <ul style="list-style-type: none"> <li>o 시스템의 하위 구성요소 확인(하위 시스템 작업, 향후 작업 등 필요한 모든 프로그램)</li> <li>o 다른 시스템 구성요소(하드웨어, 소프트웨어, 지원장비, 시스템 인터페이스)와 소프트웨어의 통합 확인</li> </ul> </li> </ul>
08-12	프로젝트 계획서	<ul style="list-style-type: none"> <li>- 다음을 정의함: <ul style="list-style-type: none"> <li>o 개발되어야 할 작업산출물</li> <li>o 사용될 생명주기 모델과 방법론</li> <li>o 프로젝트 관리와 관련된 고객 요구사항</li> <li>o 완성되어야 할 과업</li> <li>o 과업 책임자</li> <li>o 프로젝트 자원</li> <li>o 일정, 이정표 그리고 목표 날짜</li> <li>o 견적</li> <li>o 품질 기준</li> </ul> </li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
		<ul style="list-style-type: none"> <li>- 다음을 식별함:               <ul style="list-style-type: none"> <li>o 주요 의존 관계</li> <li>o 요구된 작업산출물</li> <li>o 프로젝트 위험 및 위험 완화 계획</li> <li>o 미 완성된 과업들에 대한 예외상황 활동</li> </ul> </li> </ul>
08-13	품질 계획서	<ul style="list-style-type: none"> <li>- 품질을 위한 목표/목적</li> <li>- 품질을 보장하기 위해 필요한 활동 과업을 정의함</li> <li>- 관련된 작업산출물을 참조함</li> <li>- 품질 심사 및 보증을 위한 방법</li> <li>- 규제적 요구사항, 표준, 고객 요구사항을 참조함</li> <li>- 예측되는 품질 기준을 식별함</li> <li>- 정의된 생명주기와 계획된 관련 활동을 위한 시간일정과 체크포인트를 명시함</li> <li>- 요구된 품질을 달성하기 위한 목표 시간 계획</li> <li>- 목적 달성을 위한 방법:               <ul style="list-style-type: none"> <li>o 수행되어야 할 과업</li> <li>o 과업 책임자</li> <li>o 수행되어야 하는 감사</li> <li>o 자원 제공 이행의지</li> </ul> </li> <li>- 작업산출물과 프로세스 과업에 대한 품질 기준을 식별함</li> <li>- 시정조치 이전에 허용 수준 혹은 경계값을 명시함</li> <li>- 품질 측정 및 벤치마킹 데이터를 정의함</li> <li>- 품질 기록을 수집하는 기간 및 절차와 수단을 정의함</li> <li>- 낮은 품질로 인해 영향을 받는 프로세스로 수집된 품질 기록을 제공하는 체계를 명시함</li> <li>- 품질에 책임 있는 부서나 기능에 의해 승인됨</li> </ul>
08-19	위험관리 계획서	<ul style="list-style-type: none"> <li>- 프로젝트 위험이 식별되고 우선순위가 매겨짐</li> <li>- 위험을 추적하는 절차와 수단</li> <li>- 시정 조치가 요구될 때 식별해야 할 임계값 기준</li> <li>- 위험을 완화하기 위해 제안된 방법:               <ul style="list-style-type: none"> <li>o 위험 완화기(mitigator)</li> <li>o 임시 방편(work around)</li> <li>o 시정조치 활동/과업</li> <li>o 감시 기준</li> <li>o 위험 측정 체계</li> </ul> </li> </ul>
08-20	위험완화 계획서	<ul style="list-style-type: none"> <li>- 계획된 위험 처리 활동과 과업:               <ul style="list-style-type: none"> <li>o 수용할 수 없는 위험 또는 위험 조합을 위해 선택된 위험 취급 명세를 기술함</li> <li>o 위험을 처리하는 과정에서 발견될 수 있는 모든 어려움을 기술함</li> </ul> </li> <li>- 처리 일정</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
		<ul style="list-style-type: none"> <li>- 처리 자원 및 그들의 할당</li> <li>- 책임과 권한: <ul style="list-style-type: none"> <li>o 위험처리가 이루어지고 있음을 보장할 책임이 있는 사람과 그들의 권한을 식별함</li> </ul> </li> <li>- 처리 통제 측도는: <ul style="list-style-type: none"> <li>o 위험 처리의 효과성을 평가하기 위해 사용될 측도를 정의함</li> </ul> </li> <li>- 처리 비용</li> <li>- 당사자들 사이의 인터페이스는: <ul style="list-style-type: none"> <li>o 처리가 적절히 구현될 때 발생하는 이해관계자들 사이 또는 프로젝트 총괄계획과의 조정을 기술함</li> </ul> </li> <li>- 환경/기반구조: <ul style="list-style-type: none"> <li>o 환경 혹은 기반구조 요구사항 또는 영향을 기술함(예, 조치가 가지는 안전 혹은 보안 효과)</li> </ul> </li> <li>- 위험 조치 계획 변경 절차와 이력</li> </ul>
08-21	소프트웨어 시험 계획서	<ul style="list-style-type: none"> <li>- 요구사항에 정의된 대로 특성 및/또는 기능이 운영됨을 검증하기 위한 전략을 식별함</li> </ul>
08-25	단위시험 계획서	<ul style="list-style-type: none"> <li>- 요구사항과 설계에 대해 단위 기능성과 비 기능성 요구사항(프로그램, 블록, 모듈, 루틴)을 검증하는 전략을 식별함</li> <li>- 요구사항이 검증되는 방법을 명세화함</li> </ul>
08-27	문제관리 계획서	<ul style="list-style-type: none"> <li>- 식별, 기록, 기술 및 분류를 포함하는 문제 해결 활동을 정의함</li> <li>- 문제 해결 접근법: 문제의 평가와 시정</li> <li>- 문제 추적을 정의함</li> <li>- 모든 시간 제약사항</li> <li>- 문제 해결책을 수집하고 배포하는 체계</li> </ul>
08-28	변경관리 계획서	<ul style="list-style-type: none"> <li>- 식별, 기록, 기술, 분석 및 구현을 포함한 변경관리 활동을 식별함</li> <li>- 변경관리 요청 상태를 추적하는 접근방법을 정의함</li> <li>- 검증 및 확인 활동을 정의함</li> <li>- 변경 승인 및 관계를 분석함</li> </ul>
09-00	정책	<ul style="list-style-type: none"> <li>- 권한을 부여 받음</li> <li>- 정책의 영향을 받는 모든 사람들에게 사용가능함</li> <li>- 준수해야 할 프랙티스/규칙 수립</li> </ul>
10-00	프로세스 기술서	<ul style="list-style-type: none"> <li>- 프로세스와 절차에 대한 자세한 기술서는 다음 사항을 포함: <ul style="list-style-type: none"> <li>o 표준 프로세스의 조정(적용가능한)</li> <li>o 프로세스 목적</li> <li>o 프로세스 수행결과</li> <li>o 수행되어야 할 과업, 활동 및 과업의 순서</li> <li>o 과업 활동들 사이의 주요한 의존관계</li> <li>o 과업을 수행하기위해 필요한 예상시간</li> <li>o 입력/출력 작업산출물</li> <li>o 작업 입력물과 출력물 사이의 연결</li> </ul> </li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
		<ul style="list-style-type: none"> <li>- 프로세스 시작과 종료 기준을 식별함</li> <li>- 프로세스에 대한 내외부 인터페이스를 식별함</li> <li>- 프로세스 측도를 식별함</li> <li>- 기대 품질을 식별함</li> <li>- 기능별 역할과 책임을 식별함</li> <li>- 책임있는 사람에 의해 승인됨</li> </ul>
10-01	생명주기 모델	<ul style="list-style-type: none"> <li>- 각 생명주기 단계에서 수행되어야 할 상위 수준의 활동 기술서</li> <li>- 생명주기 단계의 순서</li> <li>- 주요 생명주기 단계별 의존 관계를 식별함</li> <li>- 각 생명주기별 입력물과 결과물을 식별함</li> <li>- 모델에서 주요 의사결정 시점(이정표)를 식별함</li> <li>- 모델에서 품질 통제시점을 식별함</li> </ul>
10-02	시험절차서	<ul style="list-style-type: none"> <li>- 다음을 식별함: <ul style="list-style-type: none"> <li>o 시험 이름</li> <li>o 시험 기술(서)</li> <li>o 시험 완료 날짜</li> </ul> </li> <li>- 잠재적 구현 이슈를 식별함</li> <li>- 시험절차를 완성할 사람을 식별함</li> <li>- 선행조건을 식별함</li> <li>- 시험자와 결과에서 요구되는 단계의 수, 활동을 포함한 절차 단계를 식별함</li> <li>- 다음과 관련하여 시험에서 사용됨: <ul style="list-style-type: none"> <li>o 소프트웨어와 시스템 설치</li> <li>o 소프트웨어 통합</li> <li>o 소프트웨어</li> <li>o 시스템 통합</li> <li>o 시스템</li> </ul> </li> </ul>
11-01	소프트웨어 제품	<ul style="list-style-type: none"> <li>- 소프트웨어 항목들의 통합</li> <li>- 컴퓨터 프로그램의 집합, 절차, 그리고 관련된 문서와 데이터</li> </ul>
11-02	소프트웨어 구성 요소	<ul style="list-style-type: none"> <li>- 소프트웨어 설계에 정의된 소프트웨어 단위의 조합</li> </ul>
11-05	소프트웨어 단위	<ul style="list-style-type: none"> <li>- 수립된 코딩 표준을 준수함(언어와 응용 프로그램에 따라 적절하게)</li> <li>- 데이터 정의 표준을 따름(언어와 응용 프로그램에 따라 적절하게)</li> <li>- 정의된 개체 관계</li> <li>- 데이터베이스 레이아웃이 정의됨</li> <li>- 파일구조와 블록킹이 정의됨</li> <li>- 데이터 구조가 정의됨</li> <li>- 알고리즘이 정의됨</li> <li>- 기능적 인터페이스가 정의됨</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
12-01	제안 요청서	<ul style="list-style-type: none"> <li>- 요구사항 명세서 참조</li> <li>- 공급자의 선택 기준 식별</li> <li>- 다음과 같은 요구된 특성을 식별함: <ul style="list-style-type: none"> <li>o 시스템구조, 형상 요구사항이나 서비스 요구사항(컨설턴트, 유지 관리 등)</li> <li>o 품질 기준 또는 요구사항</li> <li>o 프로젝트 일정 요구사항</li> <li>o 요청 인도/서비스 일자</li> <li>o 비용/가격 기대치</li> <li>o 규제 표준/요구사항</li> </ul> </li> <li>- 제출(Submission) 제약사항을 식별함 <ul style="list-style-type: none"> <li>o 응답 재제출(resubmission) 일자</li> <li>o 응답 양식에 관한 요구사항</li> </ul> </li> </ul>
12-02	폐기 요청서	<ul style="list-style-type: none"> <li>- 폐기를 위한 컴포넌트나 프로젝트 이름을 식별함</li> <li>- 기본 기술서를 식별함</li> <li>- 제안된 폐기날짜를 식별함</li> <li>- 컴포넌트/프로젝트 지속기간을 식별함</li> <li>- 폐기를 승인할 사람을 식별함</li> </ul>
13-00	기록	<ul style="list-style-type: none"> <li>- 달성된 결과를 설명하거나 프로세스에서 수행된 활동에 대한 증거를 보여주는 작업산출물</li> <li>- 검색 및 식별가능한 데이터 집합의 일부인 항목</li> </ul>
13-01	인수 기록	<ul style="list-style-type: none"> <li>- 인도물 수취 기록</li> <li>- 받은 날짜를 식별함</li> <li>- 인도된 구성요소를 식별함</li> <li>- 정의된 모든 고객 인수 기준에 대한 검증을 기록함</li> <li>- 수령 고객이 서명함</li> </ul>
13-03	백업/복구 기록	<ul style="list-style-type: none"> <li>- 백업 날짜</li> <li>- 백업 내용과 연계 버전을 목록화</li> <li>- 백업 위치 목록화</li> <li>- 백업시 연관된 시스템 속성과 형상을 식별함</li> <li>- 연관된 백업 절차를 식별함</li> </ul>
13-04	의사소통 기록	<ul style="list-style-type: none"> <li>- 모든 형태의 개인간 통신 내역 <ul style="list-style-type: none"> <li>o 편지</li> <li>o 팩스</li> <li>o 이메일</li> <li>o 음성녹음</li> <li>o 텔레타이프</li> </ul> </li> </ul>
13-06	인도 기록	<ul style="list-style-type: none"> <li>- 고객에게 전자적으로 배송/인도된 품목의 기록</li> <li>- 다음을 식별함: <ul style="list-style-type: none"> <li>o 받는 사람</li> <li>o 배달 주소</li> <li>o 배송일자</li> </ul> </li> <li>- 배송 품목에 대한 송장 기록</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
13-07	문제 기록	<ul style="list-style-type: none"> <li>- 제출된 이름 및 관련된 자세한 연락처를 식별함</li> <li>- 수리 책임 그룹/사람을 식별함</li> <li>- 문제에 대한 기술을 포함함</li> <li>- 문제의 분류를 식별함 (중요도, 긴급성, 관련성 등)</li> <li>- 보고된 문제의 심각성을 식별함 (심각, 주요, 사소)</li> <li>- 보고된 문제의 상태를 식별함</li> <li>- 수정되어야 할 목표 릴리스 문제를 식별함</li> <li>- 보고된 날짜를 식별함</li> <li>- 예상 마감 일자를 식별함</li> <li>- 모든 종결 기준을 식별함</li> <li>- 재 검사를 식별함</li> <li>- 결함(의도된거나 특정한 사용과 연관된 요구사항 미달성) 기술함</li> <li>- 시스템 형상정보를 식별함 (릴리스 버전, 시스템 소프트웨어, 하드웨어 구성 등)</li> <li>- 관련된 결함 보고, 고객 요청, 반복된 문제, 관련 수정을 식별함</li> <li>- 제품에 영향을 준 구성요소를 식별함</li> <li>- 관련된 지원 정보를 식별함(덤프, 파일 등)</li> <li>- 적용가능한 소프트웨어 릴리스나 버전 정보를 식별함</li> </ul>
13-08	설치 기록	<ul style="list-style-type: none"> <li>- 설치된 것들에 대한 기록</li> <li>- 릴리스나 기록된 시스템 형상 정보</li> <li>- 기록된 특수 사이트에 특정 정보</li> <li>- 수행된 인수 테스트를 식별함</li> <li>- 수집된 설치 성능 정보 <ul style="list-style-type: none"> <li>o 변환 및 설치이후 발견된 결함 수</li> <li>o 설치 시간</li> </ul> </li> <li>- 설치 변환 후 시스템 가동 능력</li> <li>- 고객 승인 기록</li> </ul>
13-14	진척도 상태 기록	<ul style="list-style-type: none"> <li>- 계획 (계획에 대비한 실제) 상태의 기록, 예를 들면: <ul style="list-style-type: none"> <li>o 계획과업에 대비한 실제 과업의 상태</li> <li>o 설정된 목표/목적에 대비한 실제 결과의 상태</li> <li>o 계획 자원배분에 대비한 실제 자원배분 상태</li> <li>o 계획 예산에 대비한 실 비용의 상태</li> <li>o 계획 일정에 대비한 실제 진행기간의 상태</li> <li>o 계획 품질에 대비한 실제 품질의 상태</li> </ul> </li> <li>- 계획에 대비해 편차가 발생한 활동에 대한 원인 기록</li> </ul>
13-16	변경요청서	<ul style="list-style-type: none"> <li>- 변경의 목적을 식별함</li> <li>- 요청 상태를 식별함 (신규, 승인, 거부)</li> <li>- 요청자의 연락 정보를 식별함</li> <li>- 영향을 받는 시스템(들)</li> <li>- 기존 시스템(들)의 운영에 주는 영향이 정의됨</li> <li>- 관련 문서에 주는 영향이 정의됨</li> <li>- 요청의 중요성, 필요한 일자</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
13-17	고객요청서	<ul style="list-style-type: none"> <li>- 다음과 같은 요청 목적을 식별함: <ul style="list-style-type: none"> <li>o 신규 개발</li> <li>o 개선</li> <li>o 내부 고객</li> <li>o 운영</li> <li>o 문서화</li> <li>o 정보 차원</li> </ul> </li> <li>- 다음과 같은 요청 상대 정보를 식별함: <ul style="list-style-type: none"> <li>o 요청 일자</li> <li>o 현재 상태</li> <li>o 할당된 날짜와 책임있는 소유자</li> <li>o 확인된 날짜</li> <li>o 종결 일자</li> </ul> </li> <li>- 요청의 우선순위/심각성을 식별함</li> <li>- 다음과 같은 고객정보를 식별함: <ul style="list-style-type: none"> <li>o 요청을 시작한 회사/개인</li> <li>o 연락 정보와 세부사항</li> <li>o 시스템 사이트 형상 정보</li> <li>o 영향을 받는 시스템(들)</li> <li>o 기존 시스템 운영에 주는 영향</li> <li>o 요청의 중요성</li> <li>o 예상되는 고객 응답/종결 요구사항</li> </ul> </li> <li>- 요구되는 요구사항/표준을 식별함</li> <li>- 요청과 같이 보내진 정보를 식별함 (제안요청서, 덤프, 등)</li> </ul>
13-18	품질 기록	<ul style="list-style-type: none"> <li>- 유지할 정보를 정의함</li> <li>- 과업/행동/프로세스가 산출하는 정보를 정의함</li> <li>- 데이터 수집시기를 정의함</li> <li>- 연계된 데이터 출처를 정의함</li> <li>- 연계된 품질 범위를 식별함</li> <li>- 정보를 이용하는 연관 측정사항을 식별함</li> <li>- 기록을 위해 준수하거나 혹은 기록에 의해 만족되어야만 하는 요구사항을 식별함</li> </ul>
13-19	검토 기록	<ul style="list-style-type: none"> <li>- 검토에 대한 상황 정보를 제공함: <ul style="list-style-type: none"> <li>o 검토 대상</li> <li>o 참석한 검토자 목록</li> <li>o 검토 상태</li> </ul> </li> <li>- 검토범위에 대한 정보를 제공함: <ul style="list-style-type: none"> <li>o 체크리스트</li> <li>o 검토 기준</li> <li>o 요구사항</li> <li>o 표준 준수성</li> </ul> </li> </ul>



작업 산출물 ID	작업산출물 이름	작업산출물 특성
		<ul style="list-style-type: none"> <li>- 다음에 대한 정보를 기록함:               <ul style="list-style-type: none"> <li>o 검토의 용이성</li> <li>o 검토 준비시간</li> <li>o 검토시간</li> <li>o 검토자, 역할과 전문성</li> </ul> </li> <li>- 필요한 시정조치를 식별함:               <ul style="list-style-type: none"> <li>o 위험 식별</li> <li>o 발견된 차이와 문제에 대한 우선순위 목록</li> <li>o 활동, 문제를 고치기 위해 행해져야하는 과업</li> <li>o 시정조치를 위한 책임자</li> <li>o 식별된 문제를 위한 상태 및 목표 종결 일자</li> </ul> </li> </ul>
13-21	변경관리기록	<ul style="list-style-type: none"> <li>- 기준선 상의 제품/공식적인 프로젝트 릴리스 라이브러리의 제품에 대한 변경을 통제하는 체계로 사용</li> <li>- 변경되고, 기준선 상의 제품(작업제품, 소프트웨어, 고객 문서, 등)에 이루어진 변경 기록               <ul style="list-style-type: none"> <li>o 변경에 영향을 받는 시스템과 문서를 식별함</li> <li>o 변경 요청자를 식별함</li> <li>o 변경에 책임이 있는 부서를 식별함</li> <li>o 변경 상태를 식별함</li> <li>o 관련 고객 요청, 내부 변경 요청, 등과의 연결</li> </ul> </li> <li>- 적절한 승인</li> <li>- 중복된 요청이 식별되고 그룹화 됨</li> </ul>
13-22	추적 기록	<ul style="list-style-type: none"> <li>- 추적되어야 할 요구사항을 식별함</li> <li>- 생명주기의 작업산출물과 요구사항의 대응을 식별함</li> <li>- 요구사항과 작업산출물 분할과의 연결을 제공함 (요구사항 설계 코드 시험 인도, 등)</li> <li>- 요구사항과 생명주기의 모든 단계에 걸친 관련된 작업산출물과의 양방향 대응을 제공함</li> <li>- 비고: 다른 정의된 작업산출물의 기능으로 포함 될 수 있음 (예: 설계 분할을 위한 CASE 도구는 그 특성의 일부로 대응을 포함할 수도 있음)</li> </ul>
13-27	폐기 공지	<ul style="list-style-type: none"> <li>- 고객과 사용자에게 제품 폐기를 알림</li> </ul>
14-00	등록	<ul style="list-style-type: none"> <li>- 정의된 순서에 따라 아래와 같은 것들이 가능하도록 하는 데이터와 정보의 편집               <ul style="list-style-type: none"> <li>o 활동들이 수행되고 있다는 증거를 보여주는 전반적인 관점</li> <li>o 감시와 분석</li> <li>o 지속적으로 프로세스가 수행되고 있다는 증거를 제공함</li> </ul> </li> </ul>
14-01	변경 이력	<ul style="list-style-type: none"> <li>- 객체에 발생한 모든 변경에 대한 기록 (문서, 파일, 소프트웨어 모듈 등):               <ul style="list-style-type: none"> <li>o 변경의 기술</li> <li>o 변경된 객체에 관한 버전 정보</li> <li>o 변경 일자</li> <li>o 변경 요청자 기록</li> <li>o 변경 제어 기록 정보</li> </ul> </li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
14-02	시정조치 관리대장	<ul style="list-style-type: none"> <li>- 초기 문제를 식별함</li> <li>- 정의된 활동을 완료하기 위한 책임자를 식별함</li> <li>- 해결책을 정의함 (문제 수정을 위한 일련의 행동)</li> <li>- 개시일과 목표 완료일을 식별함</li> <li>- 상태 지표를 포함함</li> <li>- 후속 감사 조치를 표시함</li> </ul>
14-04	시험 로그	<ul style="list-style-type: none"> <li>- 소프트웨어 생명 주기에 따른 시험 결과 등록</li> <li>- 시험된 요소를 식별함</li> <li>- 시험 실행 일자를 식별함</li> <li>- 시험 결과에 대한 책임자를 식별함</li> </ul>
14-06	일정	<ul style="list-style-type: none"> <li>- 실행될 과업을 식별함</li> <li>- 해당 과업의 예상/실제 시작일과 종료일을 식별함</li> <li>- 주요 과업이나 과업 의존성을 허용함</li> <li>- 계획 일자에 대한 과업 완료상황을 식별함</li> <li>- 예정된 자원 데이터와 대응시킴</li> </ul>
14-09	작업 분해 구조도	<ul style="list-style-type: none"> <li>- 수행되어야 할 과업 및수정사항을 정의함</li> <li>- 과업에 대한 담당자를 문서화함</li> <li>- 과업간 주요한 의존성을 문서화함</li> <li>- 입력 및 출력 작업산출물을 문서화함</li> <li>- 정의된 작업산출물간의 중요한 의존성을 문서화함</li> </ul>
15-00	보고서	<ul style="list-style-type: none"> <li>- 다음과 같은 상황을 설명하는 작업산출물: <ul style="list-style-type: none"> <li>o 결과와 상태를 포함함</li> <li>o 적용가능하고 연관된 정보를 식별함</li> <li>o 고려사항과 제약사항을 식별함</li> <li>o 증거와 확인 데이터를 제공함</li> </ul> </li> </ul>
15-01	분석 보고서	<ul style="list-style-type: none"> <li>- 분석대상</li> <li>- 분석 인원</li> <li>- 사용된 분석 기준: <ul style="list-style-type: none"> <li>o 사용된 기준 선택이나 우선순위 구조</li> <li>o 의사 결정 기준</li> <li>o 품질 기준</li> </ul> </li> <li>- 다음 결과를 기록함: <ul style="list-style-type: none"> <li>o 결정 및 선택된 사항</li> <li>o 선택 이유</li> <li>o 설정된 가정</li> <li>o 잠재적 위험</li> </ul> </li> <li>- 분석할 정확성 측면은 다음을 포함함: <ul style="list-style-type: none"> <li>o 완전성</li> <li>o 이해가능성</li> <li>o 시험 가능성</li> <li>o 검증 가능성</li> <li>o 구현 가능성</li> </ul> </li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
		<ul style="list-style-type: none"> <li>○ 타당성</li> <li>○ 일관성</li> <li>○ 내용 적절성</li> </ul>
15-03	형상 상태 보고서	<ul style="list-style-type: none"> <li>- 형상 관리의 항목 수를 식별함</li> <li>- 형상 관리와 관련된 위험을 식별함</li> <li>- 분실된 형상 관리 항목의 수와 분실 이유를 식별함</li> <li>- 형상 관리와 연계된 문제점이나 이슈를 식별함</li> <li>- 형상항목을 받은 당사자를 식별함</li> <li>- 작성된 기준선을 식별함</li> </ul>
15-06	프로젝트 상태 보고서	<ul style="list-style-type: none"> <li>- 프로젝트의 현재 상태 보고서</li> <li>- 일정: <ul style="list-style-type: none"> <li>○ 계획된 진척도</li> <li>○ 실제 진척도</li> <li>○ 계획 진척도와 차이에 대한 설명</li> <li>○ 지속적 진행에 대한 위협</li> <li>○ 진척도를 유지하기 위한 부가 계획</li> </ul> </li> <li>- 예산: <ul style="list-style-type: none"> <li>○ 계획된 비용</li> <li>○ 실제 비용</li> <li>○ 계획 비용과 실제 비용과의 차이에 대한 설명</li> <li>○ 향후 예상 비용</li> <li>○ 예산 목표 달성하기 위한 돌발상황 계획</li> </ul> </li> <li>- 품질 목표: <ul style="list-style-type: none"> <li>○ 실제 품질 측정</li> <li>○ 계획 품질과 실제 품질과의 차이 이유</li> <li>○ 품질 목표를 달성하기 위한 계획</li> </ul> </li> <li>- 프로젝트 이슈: <ul style="list-style-type: none"> <li>○ 프로젝트 목표 달성에 영향을 미칠 수 있는 이슈</li> <li>○ 프로젝트 목표 달성에 대한 위협을 극복할 수 있는 임시 계획</li> </ul> </li> </ul>
15-08	위험 분석 보고서	<ul style="list-style-type: none"> <li>- 분석된 위험을 식별함</li> <li>- 분석 결과를 기록함: <ul style="list-style-type: none"> <li>○ 위험을 완화할 수 있는 잠재적 방법</li> <li>○ 설정된 가정사항</li> <li>○ 제약 사항</li> </ul> </li> </ul>
15-12	문제 상태 보고서	<ul style="list-style-type: none"> <li>- 문제 기록의 요약을 보여줌: <ul style="list-style-type: none"> <li>○ 문제 범주/분류에 따라</li> </ul> </li> <li>- 문제 해결의 상태: <ul style="list-style-type: none"> <li>○ 해결된 문제 대 미해결 문제</li> </ul> </li> </ul>
15-24	감사 보고서	<ul style="list-style-type: none"> <li>- 감사 목적을 선언함</li> <li>- 감사를 위해 사용된 방법</li> <li>- 감사의 기반이 되는 요구사항</li> <li>- 가정과 한계점</li> <li>- 필요한 상황 및 범위 정보를 식별함:</li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
		<ul style="list-style-type: none"> <li>○ 감사 일자</li> <li>○ 감사를 받은 심사대상 조직단위</li> <li>○ 후원자 정보</li> <li>○ 감사팀</li> <li>○ 참가자</li> <li>○ 범위/포함범위</li> <li>○ 참가자의 정보</li> <li>○ 사용된 심사도구(체크리스트, 도구)</li> <li>- 결과를 기록함: <ul style="list-style-type: none"> <li>○ 식별된 부적합성</li> <li>○ 필요한 시정조치</li> </ul> </li> </ul>
16-00	저장소	<ul style="list-style-type: none"> <li>- 구성요소에 대한 저장소</li> <li>- 저장과 검색가능한 능력</li> <li>- 내용을 검색할 수 있는 능력</li> <li>- 속성 설명을 포함한 내용물을 목록화함</li> <li>- 관련 그룹들 사이에 구성요소를 이전하고 및 공유함</li> <li>- 효율적인 접근 통제</li> <li>- 구성요소 설명을 유지관리함</li> <li>- 구성요소의 저장버전의 복구</li> <li>- 구성요소의 상태를 보고하는 능력</li> <li>- 구성요소의 변경은 변경/사용자 요청까지 추적 가능함</li> </ul>
16-03	형상 관리 라이브러리	<ul style="list-style-type: none"> <li>- 라이브러리로부터 제품을 정확히 생성함</li> <li>- 모든 릴리스 또는 시험 형상을 재생성할 수 있음</li> <li>- 형상 상태를 보고하는 능력</li> </ul>
17-03	고객 요구사항	<ul style="list-style-type: none"> <li>- 정의된 목적/목표</li> <li>- (계약) 검토의 이슈/요구사항을 포함함</li> <li>- 다음을 식별함: <ul style="list-style-type: none"> <li>○ 시간 일정/제약사항</li> <li>○ 요구된 특성과 기능적 특성</li> <li>○ 필요한 성능 고려사항/제약사항</li> <li>○ 필요한 내부/ 외부 인터페이스 고려사항/ 제약사항</li> <li>○ 요구된 시스템 특성/ 제약사항</li> <li>○ 인간공학 고려사항/ 제약사항</li> <li>○ 보안 고려사항/ 제약사항</li> <li>○ 환경 고려사항/제약사항</li> <li>○ 운영 고려사항/ 제약사항</li> <li>○ 유지관리 고려사항/ 제약사항</li> <li>○ 설치 고려사항/ 제약사항</li> <li>○ 지원 고려사항/제약사항</li> <li>○ 설계 제약사항</li> <li>○ 안전성/ 신뢰성 고려사항/ 제약사항</li> <li>○ 품질 요구사항/ 기대치</li> </ul> </li> </ul>

작업 산출물 ID	작업산출물 이름	작업산출물 특성
17-08	인터페이스 요구 사항	<ul style="list-style-type: none"> <li>- 두 제품, 프로세스 또는 프로세스 과업간 관계를 정의함</li> <li>- 둘 사이에 공통요소에 대한 기준과 형식을 정의함</li> <li>- 주요 시기적 의존성 또는 순차적 순서를 정의함</li> </ul>
17-11	소프트웨어 요구 사항	<ul style="list-style-type: none"> <li>- 사용된 표준을 식별함</li> <li>- 소프트웨어 구조 고려사항/제약사항을 식별함</li> <li>- 필요한 소프트웨어 구성요소를 식별함</li> <li>- 소프트웨어 요소간의 관계를 식별함</li> <li>- 다음을 고려함: <ul style="list-style-type: none"> <li>o 필요한 소프트웨어 성능 특성</li> <li>o 필요한 소프트웨어 인터페이스</li> <li>o 필요한 보안 특성</li> <li>o 데이터베이스 설계 요구사항</li> <li>o 필요한 오류 처리 및 복구 속성</li> </ul> </li> </ul>
17-14	시험사례 명세서	<ul style="list-style-type: none"> <li>- 시험사례를 식별함</li> <li>- 시험 항목</li> <li>- 입력 명세서</li> <li>- 출력 명세서</li> <li>- 환경적 니즈</li> <li>- 특별한 절차 요구사항</li> <li>- 사례간 의존성</li> </ul>
18-00	표준	<ul style="list-style-type: none"> <li>- 적용 대상 및 사람을 식별함:</li> <li>- 구성요소에 대한 기대값을 식별함</li> <li>- 요구사항에 대한 적합성을 제시함</li> <li>- 요구사항에 대한 예외사항 처리나 조정 방안을 포함</li> </ul>
19-00	전략	<ul style="list-style-type: none"> <li>- 만족시켜야 하는 니즈, 목적과 목표를 식별함</li> <li>- 니즈, 목표, 목적을 달성시키기 위한 선택과 접근방법을 수립함</li> <li>- 어떠한 전략적 선택에 대해 평가되는 평가 기준을 수립함</li> <li>- 다루어져야 할 제약사항이나 위험을 식별함</li> </ul>
19-06	유지관리전략	전략(Generic) 적용을 위한 요소
19-07	소프트웨어 개발 방법론	<ul style="list-style-type: none"> <li>- 소프트웨어 개발에 사용된 접근/방법을 식별함</li> <li>- 소프트웨어 개발에 사용된 생명주기 모델(waterfall, spiral, serial build, 등)을 식별함</li> <li>- 프로세스, 활동, 및 통제에 대한 상위수준의 설명을 제공함</li> </ul>
20-00	템플릿	<ul style="list-style-type: none"> <li>- 프로세스 실행결과로서 생성된 작업산출물과 관련된 속성들을 정의함</li> <li>- 이 제품 유형과 전형적으로 관련된 기술적 구성요소를 식별함</li> <li>- 예상되는 형식과 스타일을 정의함</li> </ul>
21-00	작업산출물	<ul style="list-style-type: none"> <li>- 프로세스 실행에서 나온 산출물과 관련된 속성을 정의함 <ul style="list-style-type: none"> <li>o 작업산출물에서 나타나는 주요한 구성요소</li> </ul> </li> </ul>

## 저자 소개

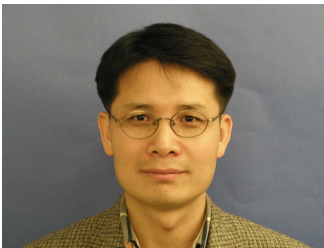
### 정호원 교수



고려대학교 경영대학 정호원 교수는 고려대학교 산업공학(학사), KAIST 석사(산업공학), University of Arizona(경영학박사, MIS 전공)를 졸업하였다. 1991년부터 ISO/IEC JTC1/SC7의 소프트웨어 품질과 프로세스 국제표준화에 참여해, SPICE Phase 2 Trials에서 ISO/IEC 15504 의 유용성에 대한 실증연구 데이터 분석가였으며, 현재는 SPICE Research International Coordinator이다. 또한 SPICE 심사원 자격 교육기관 지정과 심사원 교육 강사를 인증해주는 국제민간기구인 InterRSA(International Register Scheme for Assessors)의 창립 멤버이고 이사이다.

2004년에는 카네기멜론 대학의 SEI(Software Engineering Institute)가 전 세계에서 선정한 소프트웨어 프로세스 전문가 16명 중에 아시아에서 유일하게 선정되었고, 이들의 모임인 IPRC(International Process Research Consortium)는 "A Process Research Framework" 이라는 책을 2006년에 미국에서 공동 출판하였다. 2002에는 SEI에서 1년간 SEI에서 근무하며 SW-CMM 심사데이터를 분석하였으며 최근에는 SEI의 CMMI-DEV 심사 데이터 분석체계를 만들고 있다. 정호원 교수는 SEI 공인 CMMI-DEV Instructor로 Introduction to CMMI-DEV 공인 교육을 실시하고 있다. 또한 Software Quality Journal (Springer 발행)의 editor로 활동하고 있다. (주)데이콤, 한국전산원(현, 정보사회진흥원), Clemson University 등에서 근무한 경력을 가지고 있다.

### 김길조 대표 컨설턴트



현재 에이비엔아이(주)의 대표 컨설턴트로 근무하고 있는 김길조 박사는 산업공학(학사), 경영과학(석사), 소프트웨어공학(박사)을 공부하였다. 시스템공학연구소와 한국전자통신연구원에서 소프트웨어공학 분야를 연구하였으며, 특히 소프트웨어 프로세스 및 품질을 주제로 다양한 연구를 수행하였다. 에이비엔아이(주)에 합류한 후 SPICE, CMMI-DEV, ITSM 등을 기반으로 은행, 증권사, 소프트웨어개발업체, 방송국, 연구소 등 다양한 분야의 프로세스 개선 프로젝트를 컨설팅하였다. SPICE 선임심사원 및 강사로써 초창기부터 국내 SPICE 보급에 앞장섰으며, SPICE 국제표준화 작업에 참여하였다. 최근에는 중소기업들을 위한 소프트웨어 프로세스 표준화를 추진 중인 ISO/IEC JTC 1 SC7 WG24 활동에 참여하고 있다. 10여 회 이상에 걸쳐 SPICE 및 CMMI-DEV 공식심사에 참여하였다.

## 김도관 대표 컨설턴트



현재 비즈피어(주)의 대표 컨설턴트인 김도관 상무는 시스템 통합(SI) 업체인 포스데이타의 품질 보증팀에 근무하면서 국제품질규격인 ISO9001을 시작으로 ISO/IEC15504(SPICE), SW-CMM, CMMI-DEV 등과 같은 프로세스 개선을 위해 국제적으로 널리 활용되는 다양한 모델들을 소프트웨어 개발 및 유지보수 업무에 적용하여 지속적으로 프로세스 개선 활동을 수행하여 왔다. 그 경험을 기반으로 현재 비즈피어(주)의 대표 컨설턴트로서 2002년 05월부터 다양한 산업 분야의 대기업 및 중견기업들을 대상으로 프로세스 성숙도 수준을 향상시키기 위해 교육 및 컨설팅 등의 형태로 고객들을 지원하고 있다. 또한, 소프트웨어 및 시스템 엔지니어링 분야에서 약 10회 정도의 공식심사에 참여하여 왔으며, 특히 CMMI-DEV 모델에서 다루고 있는 성숙도 수준 2단계부터 5단계까지의 모든 프로세스 영역에 대한 컨설팅 경험을 보유하고 있다.

## 본 가이드라인이 나오기까지

ISO/IEC 15504 (프로세스 심사) 국제 표준화 작업에 참여하고 국내에 표준 보급을 목표로 1997년 설립된 KSPICE(Korea SPICE; 한국 SPICE 위원회). SPICE는 Software Process Improvement and Capability dEtermination의 약자로 통상 ISO/IEC 15504 표준을 통칭합니다. KSPICE는 심사원 자격 교육(자격분야: 소프트웨어 공학)을 시행해오고 있습니다. 또한 KSPICE는 비용이나 일정상 SPICE 심사원 자격 교육에 참여할 수 없는 중소기업 근무자를 대상으로 2004년부터 ISO/IEC 15504 표준 보급을 위한 무료 강의를 실시하고 있습니다. 이 강의는 심사원 자격 교육이라 수강자들이 실제 산업 현장에서 표준에 따라 일하기 위해서는 이 강의를 바탕으로 자신의 소속 기업에 맞는 프로세스를 정의한 '표준 프로세스(standard processes)'를 만들어 사용해야 했습니다. 일반적으로 ISO/IEC 15504 표준은 '무엇을 하라(What)' 만 정의하고 있으며, 기업 현장에서 필요한 '어떻게(How)'는 표준에서 정의하고 있지 않습니다. 기업 현장에 맞게 ISO/IEC 15504 표준에 맞는 '어떻게(How)'를 기술한 것이 표준 프로세스입니다.

ISO/IEC 15504 표준을 중소기업이 산업 현장에서 활용하는데 어려움을 해결하고자 2007년 기술표준원의 지원으로 중소기업에서 직접 사용이 가능한 표준 프로세스를 가이드라인으로 만들어 무료로 보급할 수 있게 되었습니다. 본 가이드라인은 ISO/IEC 15504 표준에 따라 작성되었으며, 중소기업에서 공통으로 사용할 수 있는 표준 프로세스입니다. 산업 현장에서는 본 가이드라인의 내용을 업무의 특성에 따라 조정을 하여 그대로 사용할 수 있습니다.

본 가이드라인의 기획과 출판에 재정적 지원을 제공한 기술표준원에 감사드립니다. 재정적 지원보다 더 중요한 것은 기술표준원이 소프트웨어 관련 중소기업이 진정으로 무엇이 필요한지를 이해하고 있다는 사실입니다. 또한 본 가이드라인이 나오기까지 힘들고 지루한 교정에 시간을 할애해 준 고려대학교 대학원 경영학과 이미원 학생과 유지석 학생에게 감사드립니다.

마지막으로 그 동안 KSPICE 중소기업 무료 강의에 참가해주신 심사인협회(KAPA) 회장이신 이경환 박사님, AB&I의 정학종 이사님과 김길조 박사님, 한국전자통신연구소의 박영식 책임연구원, LG CNS의 김도균 부장님, 한국외국어대학교의 윤재욱 교수님, 건양대학의 김진수 교수님께 감사드립니다. 본 가이드라인의 내용의 오류나 추가적으로 보완이 필요하다고 여겨지는 부분에 대한 지적은 저자에게 보내 주시기 바랍니다. 감사히 수용하겠습니다.

2007년 12월 저자 일동



높은 품질의 소프트웨어를. 더 빨리, 더 적은 비용으로 개발하기 위한

# ISO/IEC15504(SPICE) 기반의 중소기업을 위한 소프트웨어 표준 프로세스

---

2007년 12월 일 인쇄  
2007년 12월 일 발행

발행처 : 경기도 과천시 교육원길 96번지  
Tel : 590-7262~5

## 산업자원부 기술표준원

작성자 명단 :

저	자	정 호 원	고려대학교(교수)
		김 길 조	AB&I(대표컨설턴트)
		김 도 관	비즈피어(대표컨설턴트)
확 인 자	박 인 수	기술표준원(팀장)	
	노 학 엽	기술표준원(연구사)	

