

2022학년도 2학기-응용SW실무

1주차 : 프로젝트 계획하기

2022. 08. 29 (월)

Prepared by DaeKyeong Kim

Ph.D.

Dept. of CSE
University of YUHAN



1

Orientation

2

1교시: 소프트웨어개발 방법론 선정하기

3

2교시: 소프트웨어개발 방법론 선정하기

4

3교시 : 소프트웨어개발 방법론 테일러링하기

Orientation





Course Introduction

과목 개요

구분	세부 내용
교과목명	응용SW실무
교과목 개요 및 특징	<p>웹페이지를 구성하기 위한 구성 요소들과 기법들을 학습하고 웹사이트 개발을 위한 팀 프로젝트를 진행한다.</p> <p>HTML5, JavaScript, JQuery, Ajax, DBMS, Server-side Script 등을 활용하여 웹사이트를 기획하고 구축함으로써 실무역량을 함양하기 위한 과목이다.</p>
교과목표 (학습목표)	Java 프로그래밍 및 JSP 등의 응용소프트웨어를 구현하는 능력을 통합하여 실무 현장에 적용하기 위한 수단으로 프로젝트의 범위 및 일정을 체계적으로 계획하여 프로젝트를 완성하고 발표할 수 있다.
교재	코드로 배우는 스프링 웹 프로젝트 : 현업 개발을 위한 단계별 실습서 [개정판] 구멍가게 코딩단 지음 남가람북스 2018년 09월 (미개발)



Course Introduction

평가 개요

구분	배점	평가 개요
진단평가	0%	웹페이지를 구성하기 위한 구성 요소들과 기법들을 학습하고 웹사이트 개발을 위한 팀 프로젝트를 진행하기 위한 기본사항 진단.
출석평가	20%	※ 출석점수는 결석일수(日數)가 아닌 결석시수(時數)에 따라 평가·부여하므로 유의하시기 바람 ※ 교과목별 총 수업시수의 1/4을 초과하여 결석한 자는 해당 교과목의 시험에 응시할 수 없으며, F학점(미취득)
역량 평가1	30%	중간고사 서술형시험
역량 평가2	30%	기말고사 구두발표
역량 평가3	20%	과제평가



Course Introduction

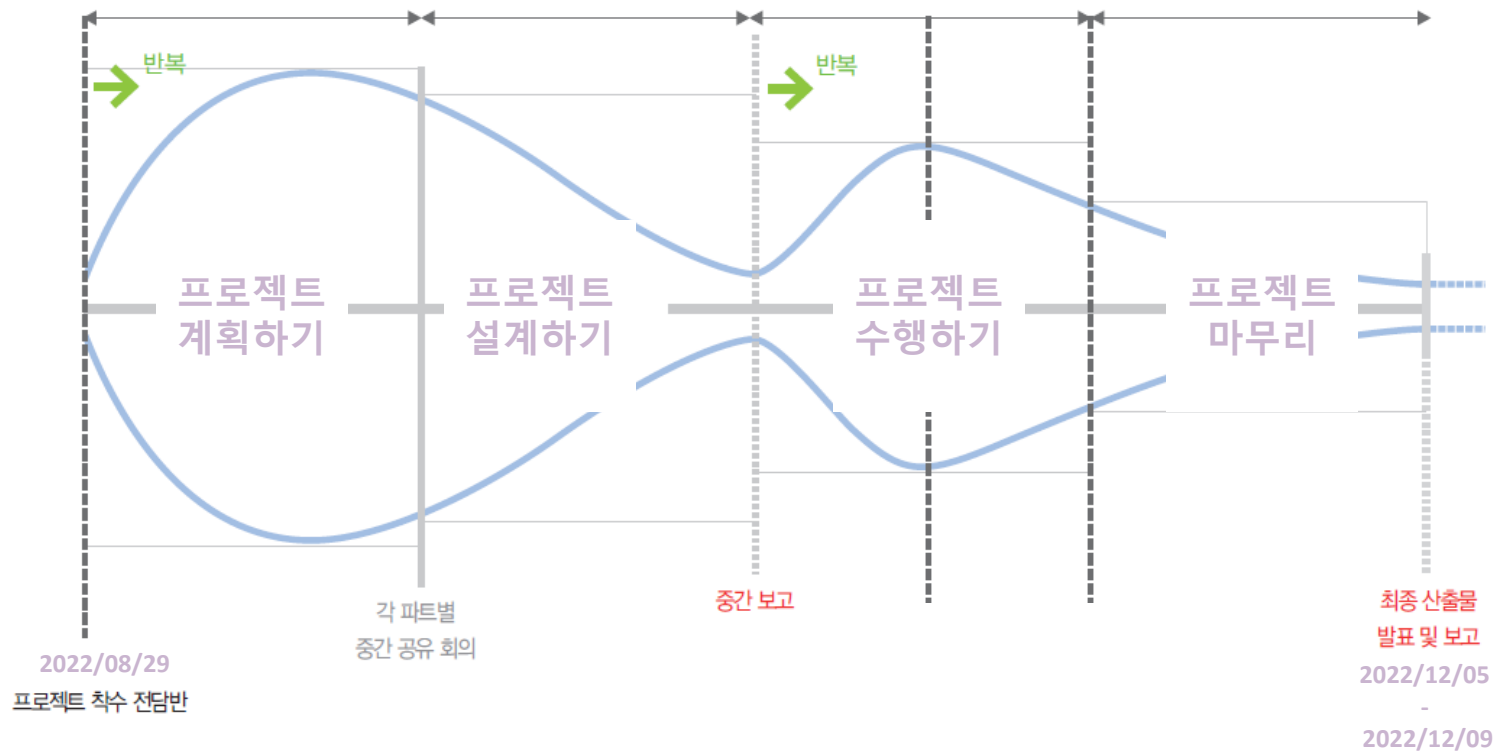
휴강에 따른 보강일

구분	추석연휴	추석연휴	개천절	한글날(대체공휴일)
휴강일	9/9(금)	9/12(월)	10/3(월)	10/10(월)
보강일(변경 전)	12/5(월)	12/9(금)	12/6(화)	12/7(수)
보강일(변경 후)	12/9(금)	12/5(월)	12/6(화)	12/7(수)



Course Introduction

수업 일정 수립



프로젝트 계획하기

1교시-2교시 :

소프트웨어개발 방법론 선정하기





1-1. 응용 소프트웨어 특성 작성

학습목표

- 개발하여야 할 응용 소프트웨어의 특성을 파악하여 정형화된 개발 방법론 선정 기준으로 활용 할 수 있다.

눈높이 체크

- 소프트웨어 종류를 알고 계신가요?

1-1. 응용 소프트웨어 특성 작성

상용 소프트웨어

- 개요
 - 응용 소프트웨어는 정부에서 2016년 9월 6일 발표한 상용 소프트웨어 백서에 의하면 상용 소프트웨어의 산업 범용 소프트웨어에 속한다.
 - 상용 소프트웨어는 산업 범용 소프트웨어와 산업 특화 소프트웨어로 구분된다. 산업 범용 소프트웨어는 시스템 소프트웨어, 미들웨어, 응용 소프트웨어로 구분되고, 산업 특화 소프트웨어는 자동차 등 13개로 구분된다. 응용 소프트웨어는 산업 범용 소프트웨어이다.



[그림 1-1] 상용 소프트웨어의 분류

1-1. 응용 소프트웨어 특성 작성

산업 범용 소프트웨어 분류 체계

● 시스템 소프트웨어의 분류 체계 파악

중분류	소분류
운영체제	임베디드와 리얼타임 운영체제, 모바일 운영체제, PC와 서버 운영체제
데이터 관리	DBMS, 데이터 통합, 빅데이터 관리, 데이터 분석
스토리지 소프트웨어	파일 시스템, 데이터 백업, 스토리지 아카이빙
소프트웨어공학 도구	프로그래밍 언어, 시험 품질 검증, 소프트웨어개발 도구
가상화 소프트웨어	서버와 PC 하이퍼바이저, 임베디드 하이퍼바이저, 자원 가상화, 애플리케이션 가상화
시스템 보안	인증과 암호, 접근 제어, 침입 탐지 및 방어

출처:

과학기술정보통신부 보도자료(2016. 9. 6), “상용 소프트웨어 현재와 미래 조망”에서 재수정

1-1. 응용 소프트웨어 특성 작성

산업 범용 소프트웨어 분류 체계

● 미들웨어의 분류 체계 파악

중분류	소분류
분산 시스템 소프트웨어	웹 애플리케이션 서버, 연계 통합 솔루션, 실시간 데이터 처리, 분산 병렬 처리, TP 모니터
IT 자원 관리	네트워크 관리, 시스템 관리, SW 실행 관리, IT 서비스 운영 관리
서비스 플랫폼	IoT 플랫폼, 클라우드 서비스, UX/UI 프레임워크, CDN
네트워크 보안	접근 제어, 보안 통신, 침입 방지, 보안 관리

출처:

과학기술정보통신부 보도자료 (2016. 9. 6), “상용 소프트웨어 현재와 미래 조망 정리”

1-1. 응용 소프트웨어 특성 작성

산업 범용 소프트웨어 분류 체계

● 응용 소프트웨어의 분류와 종류

분류	소프트웨어 종류
기업용 소프트웨어	오피스웨어, ERP, SCM, BI, CRM 소프트웨어 (5종)
영상 처리 소프트웨어	영상 인식과 분석, 영상 코덱과 스트리밍, 영상 저작·편집·합성 소프트웨어 (3종)
CG/VR 소프트웨어	3D 스캐닝과 프린팅, 모델링·애니메이션·렌더링, 가상 현실과 증강 현실, 홀로그램, 가상 시뮬레이션 소프트웨어 (5종)
콘텐츠 배포 소프트웨어	콘텐츠 보호, 콘텐츠 관리, 콘텐츠 유통 소프트웨어 (3종)
자연어 처리 소프트웨어	정보 검색과 질의응답, 의사 결정 지원, 언어 분석과 변환, 대화 처리 소프트웨어 (4종)
음성 처리 소프트웨어	음성 인식, 음성 합성, 음향 처리 소프트웨어 (3종)

출처:

과학기술정보통신부 보도자료 (2016. 9. 6), “상용 소프트웨어 현재와 미래 조망”에서 재수정

1-1. 응용 소프트웨어 특성 작성

산업 특화 소프트웨어 분류 체계

- 산업 특화 소프트웨어는 자동차, 항공, 패션 의류, 조선, 건설, 농업, 의료, 국방, 게임, 공공, 교육, 물류, 인터넷 서비스로 구분한다. 소분류는 편의 장치 SW가 1개, 안전 장치 SW가 1개, 커넥티드 SW가 2개, 개발 검증 SW가 2개, 비행 운용 SW가 4개, 항공 시스템 SW가 4개, 패션/의류 제조가 3개, 솔루션 매니지먼트가 2개, 조선이 3개, 건설이 4개, 스마트 원예가 4개, 스마트 축산이 3개, 스마트 유통 경영이 3개, 병원 정보시스템이 1개, u-Health 서비스가 1개, 무기 체계가 2개, 전력 지원 체계가 1개, 게임 툴이 2개, 게임 미들웨어가 2개, 공공 안전이 1개, 생활 안전이 2개, 산업 안전이 1개, 콘텐츠 저작 도구가 1개, 학습 관리 시스템이 1개, 서비스 플랫폼이 2개, SCM이 2개, 해운이 2개, 항만이 2개, 항공이 2개, 인터넷 서비스가 5개로 66개이다.

1-1. 응용 소프트웨어 특성 작성

산업 특화 소프트웨어 분류 체계

대분류	중분류	소분류
자동차	편의 장치 SW	인포테인먼트 SW 플랫폼/도구
	안전 장치 SW	전장 SW 플랫폼/도구
	커넥티드 SW	차량내부 통신 SW, 차량외부 통신 SW
	개발 검증 SW	설계개발 SW, 시험평가 SW
항공	비행 운용 SW	임무제어 비행운용프로그램, 시현 비행운용프로그램, 비행제어 비행운용프로그램, 비행관리시스템 프로그램
	항공 시스템 SW	항공표준 실시간 운영체제, 항공표준 기능지원 미들웨어, 항공소프트웨어 개발환경, 통합시험장치 SW
패션의류	패션/의류 제조	기획/디자인, 디자인 입력/출력, 3D/CAD/Visualizing
	솔루션 매니지먼트	가상솔루션 매니지먼트

1-1. 응용 소프트웨어 특성 작성

산업 특화 소프트웨어 분류 체계

대분류	중분류	소분류
조선	조선	설계SW, 시뮬레이션SW, 운영SW
건설	건설	계획설계, 구조설계, 시공, U-City
농업	스마트 원예	식물병 측정, 식물병 진단, 식물공장 시스템, 식물 육종
	스마트 축산	생산이력 시스템, 양돈생산·경영관리, 자동착유 시스템
	스마트 유통 경영	공급사슬 관리정보, 전자상거래, 스마트 농업경영 시스템
의료	병원 정보시스템	OCS/EMR, PACS
	u-Health 서비스	u-Health
국방	무기 체계	무기체계 내장형 SW, 전장관리 정보체계 SW
	전력 지원 체계	M&S 체계 SW

1-1. 응용 소프트웨어 특성 작성

산업 특화 소프트웨어 분류 체계

대분류	중분류	소분류
게임	게임 툴	게임관리 툴, 게임 리소스 툴
	게임 미들웨어	게임 엔진, 게임 미들웨어
공공	공공 안전	재난재해 예방 및 대처
	생활 안전	범죄치안, 의료복지
	산업 안전	환경 에너지
교육	콘텐츠 저작 도구	교육 콘텐츠 저작 도구
	학습 관리 시스템	학습 관리 시스템
	서비스 플랫폼	교육 콘텐츠 유통 플랫폼, 교육 콘텐츠 서비스

1-1. 응용 소프트웨어 특성 작성

산업 특화 소프트웨어 분류 체계

대분류	중분류	소분류
물류	SCM	Supply Chain Planning, Supply Chain Execution
	해운	Container Carrier Operation System, Bulk Carrier Operation System
	항만	Terminal Operating System, Terminal Automation System
	항공	Airport Information System, Airline System
인터넷 서비스	인터넷 서비스	정보 및 검색 서비스, 전자상거래, 소셜미디어, 협력 도구, 사물인터넷 서비스

출처:

과학기술정보통신부 보도자료 (2016. 9. 6), "상용 소프트웨어 현재와 미래 조망"에서 재수정



1-2. 개발방법론 선정

학습목표

- 정형화된 개발방법론의 특징을 고려하여 응용 소프트웨어 특성에 맞는 개발방법론을 선정할 수 있다.

눈높이 체크

- 소프트웨어의 개념을 알고 계신가요?



1-2. 개발방법론 선정

소프트웨어개발 방법론

- 소프트웨어개발 방법론은 소프트웨어 개발 전 과정에 지속적으로 적용할 수 있는 방법, 절차, 기법 등이다. 소프트웨어를 하나의 생명체로 간주하고 소프트웨어 개발의 시작부터 시스템을 사용하지 않는 과정까지의 전 과정을 형상화한 것이다. 정보시스템을 개발하는 절차와 개발 단계의 반복 현상이다. 일명 시스템 개발 주기라고도 한다.
- 소프트웨어개발 방법론은 소프트웨어를 개발함에 있어 생산성과 소프트웨어 품질을 향상시키는데 목적이 있다.

1-2. 개발방법론 선정

Software

- "소프트웨어"라는 용어는 1957년에 존 터키(John W. Tukey)가 처음 사용
- 영국 시인 조지 고든 바이런의 딸로 영국의 수학자이자 세계 최초의 프로그래머로 알려져 있는 러브레이스 백작부인 어거스터 에이다 킹(Augusta Ada King, Countess of Lovelace, 1815년 12월 10일~1852년 11월 27일)은 찰스 배비지의 연구에 대한 좋은 이해자이자 협력자로서, 배비지가 고안한 해석기관에서 처리될 목적으로 작성된 알고리즘을 통해 소프트웨어의 많은 개념을 제시
- 오늘날 우리가 인식하는 컴퓨터의 발명 이전의 소프트웨어에 대한 최초의 이론은 앨런 튜링이 1935년 수필 《Computable numbers with an application to the Entscheidungsproblem》에서 제안되었다. (결정 문제)

출처:

<https://ko.wikipedia.org/wiki/%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4>



ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO
THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable numbers, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbersome technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers e , π , etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In §16 I examine certain arguments which would seem to prove the contrary. The correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel.[†] These results

[†] Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I", *Monatsh. Math. Phys.*, 38 (1931), 173-198.



1-2. 개발방법론 선정

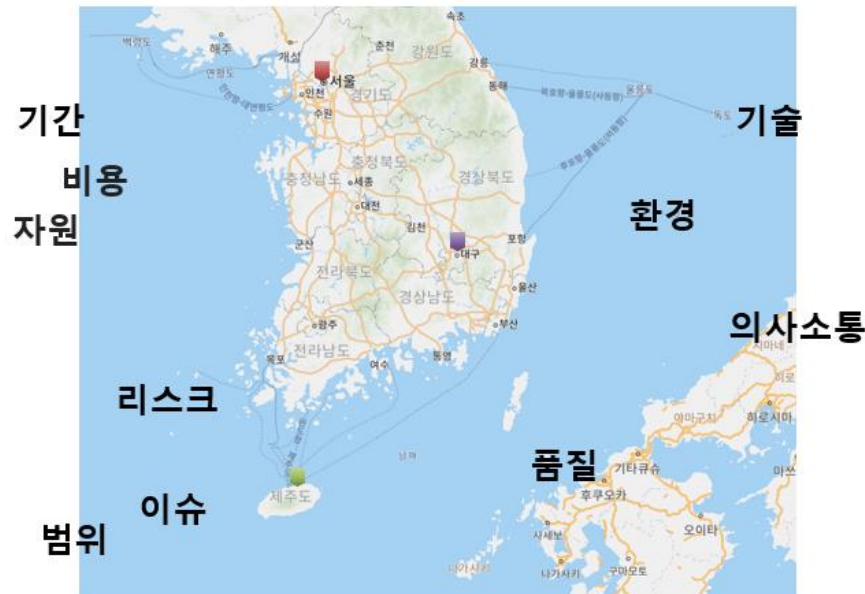
Software

- 컴퓨터 소프트웨어(computer software, 혹은 간단히 소프트웨어)는 컴퓨터에게 동작 방법을 지시하는 명령어 집합의 모임, 임베디드 분야에서는 변경 가능한 부품, 기업 환경에서는 다양한 업무 프로세스의 처리, 전산 분야에서는 까다로운 계산이나 복잡한 정보의 흐름 처리 등을 의미한다. 따라서, 일상적으로 접하는 빵이나 옷, 버스, 책상과 의자와는 상당히 많은 부분이 다르다. Software는 사람들이 보여주는 행위만큼 이해할 수 없는 복잡함이 있으며, 가끔 software는 매우 단순할 수도 있다.

1-2. 개발방법론 선정

Software

- 그렇기 때문에 이런 software를 개발하는 방법론은 아주 단순하거나 아니면 무척이나 복잡할 수 밖에 없다. 마치 다음 지도에서 보듯 서울, 대전, 부산에서 한라산까지 가능 방법이 같을 수 없듯이 소프트웨어 개발을 위한 방법론도 한결 같을 수는 없는 것이다.



1-2. 개발방법론 선정

소프트웨어개발 방법론의 종류

● 세대별 종류

세대	구분	내용
1970년대	구조적 방법론	구조화 프로그래밍
1980년대	정보공학 방법론	관리 절차와 작업 기법을 체계화
1990년대	객체지향 방법론	데이터와 그 데이터에 관련되는 동작을 모두 포함
2000년대	컴포넌트 기반 방법론	컴포넌트를 조립해서 새로운 응용 프로그램을 작성
2000년대	Agile 방법론	요구사항, 설계, 구현, 테스트의 과정으로 개발
2010년대	제품 계열 방법론	특정 제품에 수용하고 싶은 공통된 기능을 정의

출처:

소프트웨어개발 방법론 <https://ko.wikipedia.org>에서 2017. 6.13 검색



1-2. 개발방법론 선정

소프트웨어개발 방법론의 종류

❖ 관점별 종류

관점	종류
개발방법론	구조적 방법론, 정보공학 방법론, 객체지향 방법론, CBD, Agile
통합방법론	Waterfal, Prototyping, Spiral, RAD
운영방법론	ITIL(Information Technology Infrastructure Library), 서비스수준계약(SLA: Service-Level Agreement), 형상관리
조직관리방법론	프로젝트관리

1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

- 개발 준비 단계
 - 사업 팀 구성 및 테일러링의 활동목적은 사업 수행 시 발생하는 이슈 사항 처리 및 개발 방법론을 조정한다.
 - 개발 사전 준비의 활동목적은 개발에 필요한 전산 자원 사용 신청 및 발주 기관 표준을 숙지하여 개발 시작이 원활히 진행된다.

단계	활동	작업
개발 준비	사업 팀 구성 및 테일러링	사업 팀 구성
		방법론 테일러링
	개발 사전 준비	정보화 개발 준비

1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

● 분석 단계

단계	활동	작업
분석	요구사항 분석	요구사항 수집
		요구사항 정의
		유스케이스 기술
		요구사항 추적
	업무/데이터 분석	업무 분석
		데이터 분석
	아키텍처 분석	현행 아키텍처 분석
	분석 단계 테스트 계획	총괄 테스트 계획
	분석 단계 점검	분석 단계 산출물 점검



1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

● 분석 단계

- 요구사항 분석의 활동목적은 목표 시스템 구현을 위해 요구되는 사항을 분석하여 요구사항을 명확하게 정의하고 기술하기 위함이다.
- 업무 및 데이터 분석의 활동목적은 현행 업무 및 데이터를 분석하여 요구되는 비즈니스 프로세스를 정의하기 위함이다.
- 아키텍처 분석의 활동목적은 사업 대상 시스템의 현행 소프트웨어 및 시스템 아키텍처를 분석하여 목표 아키텍처 설계를 위한 기초 자료로 활용하기 위함이다.
- 분석 단계 테스트 계획의 활동목적은 개발 결과물에 대해 결함을 찾아내고 요구사항의 충족 여부를 테스트하기 위하여 단계별 테스트 일정, 방법 및 환경 등 전반적인 계획을 수립하기 위함이다.
- 분석 단계 점검의 활동목적은 분석 단계 산출물을 점검하고 미흡한 점을 사전에 조치하기 위함이다.

1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

- 분석 단계
 - 요구사항 정의서 작성 예시

순번	구분	요청내역	요청일	요청자	상태
1	사무소 관리	숫자 3자리 고유 사무소 번호, 주소, 전화 번호 관리 필요	9/1	홍길동	접수
2	차량 관리	트럭 종류 (1t, 5t, 10t, 24t, 36t), 마지막 정비 일자, 등록 말소일 관리 필요	9/1	홍길동	접수
3	고객 관리	개인 고객은 휴대폰 번호로 식별, 고객명, 주소 관리 필요	9/1	홍길동	접수
4	고객 관리	법인 고객은 사업자 등록 번호로 식별, 법인명, 주소, 연락처 관리 필요	9/1	홍길동	접수
5	렌탈 계약 관리	렌탈 계약은 차량 1대만 가능하며 렌탈 계약 시 고유 계약 번호 자동 발급 필요	9/1	홍길동	접수
6	렌탈 계약 관리	렌탈 계약 시 렌탈 사무소와 차량의 주행 거리 기록 필요	9/1	홍길동	접수
7	렌탈 계약 관리	반납 시 반납 사무소와 차량의 변경된 주행 거리 기록 필요	9/1	홍길동	접수

1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

● 설계 단계

단계	활동	작업
설계	아키텍처 설계	SW 아키텍처 설계
		시스템 아키텍처 설계
	애플리케이션 설계	클래스 설계
		사용자 인터페이스 설계
		컴포넌트 설계
		인터페이스 설계
		배치 프로그램 설계
		사용자 웹 구성 설계
	DB 설계	개념 DB 모델 설계
		논리 DB 설계
		물리 DB 설계
		데이터 흐름도(DFD) 작성
		데이터 검증식(BR) 작성

1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

● 설계 단계

- 아키텍처 설계의 활동목적은 개발 대상 시스템에 대한 응용 소프트웨어 구조와 시스템 환경 등 시스템의 구성 요소를 정의하기 위함이다.
- 애플리케이션 설계의 활동목적은 분석 단계 결과를 바탕으로 요구사항을 어떻게 구현할 것인지를 상세하게 설계하기 위함이다.
- DB 설계의 활동목적은 시스템에서 지속적으로 관리되어야 하는 데이터의 저장 DB 구조를 설계하기 위함이다
- 데이터 전환 설계의 활동목적은 DB 설계 및 구조 변경에 따른 정합성 오류 데이터 발생 방지 및 운영 중인 데이터의 품질 유지를 보장하기 위함이다.
- 설계 단계 테스트 계획의 활동목적은 구현 결과 시스템을 테스트하기 위해 총괄 테스트 계획서를 기반으로 테스트 종류별 상세 테스트 시나리오를 작성하기 위함이다.
- 설계 단계 점검의 활동목적은 설계 단계 산출물을 점검하고 미흡한 점을 사전에 조치하여 시스템 구현 시 요구사항을 누락 없이 모두 반영하기 위함이다.



1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

- 구현 단계

단계	활동	작업
구현	구현 준비	개발 환경 구성
	개발	프로그램 개발
	단위 테스트	단위 테스트
	구현 단계 점검	웹 표준 점검
		소스 품질 검사
		구현 단계 산출물 점검



1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

● 구현 단계

- 구현 준비의 활동목적은 설계 단계 산출물을 바탕으로 목표 시스템을 구현 시 필요한 개발 환경을 준비하기 위함이다.
- 개발의 활동목적은 설계 단계 산출물을 바탕으로 사용자 인터페이스, 컴포넌트, DB를 구현하여 요구사항을 충족하는 목표 시스템을 실현하기 위함이다.
- 단위 테스트의 활동목적은 구현 시스템의 단위 컴포넌트, 사용자 인터페이스의 기능 오류를 찾아 안정적인 시스템을 구축하기 위함이다.
- 구현 단계 점검의 활동목적은 목표 시스템의 요구사항을 구현 완료한 구현 단계 산출물을 점검하고, 미흡한 점을 사전에 조치하여 통합 테스트 수행을 준비하기 위함이다.

1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

● 시험 단계

- 테스트의 활동목적은 구현 완료된 시스템에 대해 요구사항이 올바르게 반영되었는지를 통합적인 관점에서 검증하기 위함이다.
- 시험 단계 점검의 활동목적은 구현 완료된 시스템의 테스트 결과를 점검하고 미흡한 점을 사전에 조치하여 전개 수행을 준비하기 위함이다.

단계	활동	작업
시험	테스트	테스트 준비 작업
		통합 테스트
		사용자 테스트
	시험 단계 점검	시험 단계 산출물 점검

1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

- 전개 단계
 - 리허설의 활동목적은 구현 완료된 시스템의 전개 전 운영 환경에 대한 설치 테스트 수행을 통해 문제 발생 요소를 사전에 해소함으로써 시스템 전개를 성공적으로 수행하기 위함이다.
 - 전개의 활동목적은 구현 완료된 시스템을 운영 환경에 설치하기 위한 계획을 수립하고, 설치 결과를 테스트하기 위함이다.

단계	활동	작업
전개	리허설	리허설 준비 작업
		최종 점검 및 리허설
	전개	전개 준비 작업
		최종 점검 및 리허설



1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

- 인도 단계

단계	활동	작업
인도	인수인계	인수인계 계획
		EA 현행화
		매뉴얼 작성
		산출물 현행화
		산출물 인수인계
	교육	교육 준비 및 교육

출처:

특허청(2014. 12) 소프트웨어 개발 방법론.



1-2. 개발방법론 선정

소프트웨어개발 방법론의 절차

- 인도 단계
 - 인수인계의 활동목적은 수행 업체가 단계별로 개발한 모든 산출물을 운영 부서에 인수인계하기 위함이다.
 - 교육의 활동목적은 설치 완료된 시스템에 대한 교육을 수행함으로써 사용자 및 운영자가 시스템을 사용하고 운영할 수 있는 역량을 갖추도록 하기 위함이다.

1-2. 개발방법론 선정

개인별 평가표

- 개인별 평가서는 평가대상 방법론, 개인별 직책, 근무 부서, 평가항목, 정량평가, 정성평가 결과로 구성한다.

개발 방법론 선정을 위한 개인별 평가서						
평가 대상 방법론 : ○○○ 방법론						
직책 : 부장 ○○○						
근무 부서 : 정보화 부서						
평가항목	정량 평가(적합, 적합하지 않다)					정성 평가
	5	4	3	2	1	
정량 평가 점수	점					

1-2. 개발방법론 선정

개발 방법론 선정을 위한 평가서

- 평가 회의를 통해 응용 소프트웨어 개발에 활용할 방법론을 선정한다.

개발 방법론 선정을 위한 평가서					
평가자 :					
직책 :					
근무 부서 :					
평가 항목	○○○ 방법론	◎◎◎방법론	◇◇◇ 방법론	△△△방법론	비고
정량 평가 총점					
정성 평가 의견					
평가 결과					
※ 평가 항목별 평가 값은 적용 가능성에 대해 최대 5점에서 1점까지 평가					

프로젝트 계획하기

3교시 :

소프트웨어개발 방법론
테일러링하기





2-1. 개발방법론 결정

학습목표

- 프로젝트의 일정, 비용, 투입 자원과 재사용 현황을 고려하여 반복 전략을 개발 방법론에 반영 할 수 있다.
- 확정된 생명 주기와 개발 방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결 기준을 정의할 수 있다.

눈높이 체크

- SDLC의 개념을 알고 계신가요?



2-1. 개발방법론 결정

개발 방법론 결정하기 흐름도

- 프로젝트의 일정, 비용, 투입 자원과 재사용 현황을 고려하여 반복 전략을 개발 방법론에 반영하고, 확정된 생명 주기와 개발 방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결 기준을 정의하는 절차는 다음과 같다.

프로젝트의 일정, 비용, 투입 자원과 재사용 현황을 개발 방법론에 반영한다.

1. 개발 방법론에 프로젝트 관리를 반영하는 방법을 참여자에게 설명한다.
2. 개발 방법론에 프로젝트 관리기준을 반영한 결과를 설명하여 확정한다.

생명 주기에 맞춰 개발 단계별 작업 및 절차를 수립한다.

1. 소프트웨어 생명 주기의 주요 프로세스를 파악한다.
2. 소프트웨어의 개발프로세스, 개발 생명 주기, 프로세스 모델을 정리한다.
3. 응용 소프트웨어 특성에 맞는 정형화된 개발 방법론을 평가하여 선정한다.

확정된 개발 방법론의 개발 단계별 활동목적과 작업내용 및 산출물에 대한 매뉴얼을 작성한다.



2-1. 개발방법론 결정

프로젝트 관리기준

- 개발 방법론에 프로젝트의 일정관리 활동을 반영한다. 일정관리 활동은 활동순서, 활동 기간산정, 일정개발, 일정통제이다.
- 개발 방법론에 프로젝트의 비용관리 활동을 반영한다. 비용관리 활동은 원가 산정, 예산 편성, 원가 통제이다.
- 개발 방법론에 프로젝트의 품질관리 활동을 반영한다. 품질관리 활동은 품질계획, 품질 보증수행, 품질통제이다.
- 개발 방법론에 프로젝트의 투입자원관리를 반영한다. 투입자원관리는 프로젝트 팀 편성, 자원 산정, 프로젝트 조직 정의, 프로젝트 팀 개발, 자원 통제, 프로젝트 팀 관리이다. 특히 프로젝트 팀 관리를 통해 팀원의 성과 추적과 피드백을 제공하여 문제를 해결하고 변경을 조정함으로써, 프로젝트 성과를 향상시킬 수 있다.
- 개발 방법론에 프로젝트의 위험관리 활동을 반영한다. 위험관리 활동은 위험식별, 위험평가, 위험대처, 위험통제로 구성된다.

2-1. 개발방법론 결정

소프트웨어개발의 생명 주기를 정리

- 소프트웨어개발의 생명 주기(Software Development Life-Cycle: SDLC)는 고객의 요구에 의해서 소프트웨어 시스템이 탄생하고, 가동, 운용되는 가운데에 유지 보수가 반복되고, 최종적으로 수명이 다하여 파기할 때까지의 전 공정을 체계 화한 개념이다. 시스템이 개발될 때부터 운용과 보수를 거쳐 생애를 마칠 때까지 어떠한 순서를 밟는지에 대한 작업 프로세스를 모델화한 것이다.
- 소프트웨어의 프로세스 모델은 개발 조직이 해야 하는 일의 종류, 순서 및 각 단계에서 생성되어야 하는 결과물을 정리해 놓은 것이다. 소프트웨어의 개발이 체계적이고 반복적으로 수행되기 위해 꼭 필요하다. 각 단계의 작업을 완료하고 그 다음 단계로 가기 위해서 필요한 조건을 명시해야 한다. 개발되는 소프트웨어의 종류, 요구되는 신뢰도, 난이도, 조직의 규모 또는 문화 등에 따라 적합한 프로세스 모델이 다르다. Waterfall model, prototyping model, spiral model이 가장 많이 알려진 프로세스 모델이다. 우수한 프로세스 모델은 소프트웨어 개발이 좀 더 짧은 시간에, 좀 더 높은 품질의 소프트웨어를, 좀 더 적은 비용으로 개발할 수 있도록 도와준다.



2-2. 작성 산출물 적용

학습목표

- 응용 소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법, 적용 도구를 정립할 수 있다.
- 테일러링된 적용 개발 방법론에 따른 작성 산출물 유형을 확정할 수 있다.

눈높이 체크

- 소프트웨어개발 방법론 테일러링을 위한 품질 관리 표준을 알고 계신가요?

2-2. 작성 산출물 적용

응용 소프트웨어개발 방법론의 프레임워크

단계(Phase)	활동(Activity)
개발 준비	팀 구성 및 테일러링
	개발 사전 준비
	요구사항 분석
분석	업무/ 데이터 분석
	아키텍처 분석
	분석 단계 테스트 계획
	분석 단계 점검
설계	아키텍처 설계
	애플리케이션 설계
	DB 설계
	데이터 전환 설계
	설계 단계 테스트 계획
구현	설계 단계 점검
	구현 준비
	개발
	단위 테스트
시험	구현 단계 점검
	테스트
전개	시험 단계 점검
	리허설
인도	전개
	인수인계
	교육

출처:

특허청(2014. 12) 소프트웨어 개발 방법론에서 요약하여 제시

2-2. 작성 산출물 적용

특정 정보시스템의 요구사항의 사례

구분	설명	요구 개수
시스템 장비 구성 요구사항	사업 수행을 위해 필요한 도입 장비 구성 요구사항을 기술	1
기능 요구사항	목표 시스템이 반드시 수행하거나 목표 시스템을 이용하여 사용자가 반드시 할 수 있어야 하는 기능(동작)에 대해 기술	43
성능 요구사항	목표 시스템이 특정 기능을 얼마나 빠르게, 멀게, 크게, 많이 수행해야 하는지에 대한 요건을 기술	8
인터페이스 요구사항	목표 시스템과 외부 세계 사이를 연결하는 시스템 인터페이스와 사용자 인터페이스에 대한 요건을 기술	4
데이터 요구사항	목표 시스템의 서비스에 필요한 DB 설계 등 데이터를 구축하기 위해 필요한 요건을 기술	3
테스트 요구사항	시스템의 기능을 테스트하고 점검하기 위한 요구사항을 기술	3
보안 요구사항	정보 자산의 기밀성과 무결성을 위해 목표 시스템의 데이터 및 기능, 운영 접근을 통제하기 위한 요건을 기술	12
품질 요구사항	목표 시스템이 가져야 하는 품질 항목, 품질 평가 대상 및 목표 값에 대한 요구사항을 기술	6
제약사항	기능, 비기능, 인터페이스, 데이터 요구사항 외에 시스템을 구축하기 위해 필요한 제약 및 요건을 기술	3
프로젝트 관리 요구사항	프로젝트 관리를 위해 필요한 요구사항으로 위험 관리, 품질 관리, 산출물 관리, 일정 관리 요구사항 등이 해당함.	13
프로젝트 지원 요구사항	프로젝트 수행 및 향후 지원을 위해 필요한 요구사항으로 표준화, 하자유지보수, 프로젝트 팀원 요구사항 등이 해당함.	13

2-2. 작성 산출물 적용

테일러링을 위한 품질 관리

- 소프트웨어 개발 단계에서 품질 관리를 한다. 품질 관리는 국제 표준으로 정한다. 대표 적인 국제 표준으로 ISO 12207 표준, CMMI 모델, SPICE 모델이 있다.

1. ISO 12207 표준

- ISO/IEC 12207 표준은 소프트웨어 생명 주기 프로세스이다. 소프트웨어와 관련된 조직과 사람, 소프트웨어 획득자, 공급자, 개발자, 운영자, 유지 보수자, 품질보증 관리자, 사용자 등의 이해관계자들이 각자의 입장에서 수행해야 할 일을 정의하고 지속적으로 개선시키기 위한 활동이다.

〈표 2-21〉 ISO 12207 표준의 프로세스

구분	주요 프로세스
기본 공정 프로세스	획득 프로세스, 공급 프로세스, 개발 프로세스, 운영 프로세스, 유지 보수 프로세스
지원 공정 프로세스	문서화 프로세스, 품질 보증 프로세스, 형상 관리 프로세스, 검증 프로세스, 확인 프로세스, 문제 해결 프로세스, 활동 검토 프로세스, 감사 프로세스
조직 공정 프로세스	기반 구조 프로세스, 관리 프로세스, 개선 프로세스, 훈련 프로세스

출처: 소프트웨어 공학, 2016. 이형주 편저, (주)지식과 미래, P.38~39



2-2. 작성 산출물 적용

테일러링을 위한 품질 관리

1. ISO 12207 표준

- ISO/IEC 12207 표준은 소프트웨어 개발 프로세스를 정의하고 향상시키기 위한 프로세스이다. 표준은 기본 공정, 지원 공정, 조직 공정으로 구성된다.

기본 공정

- 공급, 획득, 개발
- 운영, 유지보수

지원 공정

- 문서화, 형상관리, 문제해결
- 품질보증, 검증, 확인, 합동검토, 감리,

조직 공정

- 관리, 기반 구조
- 개선, 교육 훈련

테일러링을 위한 품질 관리

1. ISO 12207 표준

- 기본 공정을 파악하여 적용한다.

- ① 공급은 서비스를 제공하는 공급자 또는 공급 조직이 수행할 활동을 정의한다.
- ② 획득은 서비스를 획득하는 획득자 또는 획득 조직이 수행할 활동을 정의한다.
- ③ 개발은 소프트웨어를 개발하는 개발자 또는 개발 조직이 수행할 활동을 정의한다.
- ④ 운영은 서비스를 제공하는 운영자 또는 운영 조직이 수행할 활동을 정의한다.
- ⑤ 유지 보수는 유지 보수를 제공하는 보수자 또는 조직이 수행할 활동을 정의한다.

- 지원 공정을 파악하여 적용한다.

- ① 문서화는 산출되는 정보의 기록을 위한 활동을 정의한다.
- ② 형상 관리는 구성 관리 활동을 정의한다.
- ③ 문제 해결은 문제점을 분석하고 제거하기 위한 프로세스를 정의한다.
- ④ 품질 보증은 객관적으로 보증하기 위한 활동을 정의한다.
- ⑤ 검증은 소프트웨어 제품을 검증하기 위한 활동을 정의한다.
- ⑥ 확인은 소프트웨어 제품을 확인하기 위한 활동을 정의한다.
- ⑦ 합동 검토는 활동의 상태 및 제품을 평가하기 위한 활동을 정의한다.
- ⑧ 감리는 요구사항에 대하여 적합성을 결정하기 위한 활동을 정의한다.

- 조직 공정을 파악하여 적용한다.

- ① 관리는 프로젝트 관리를 포함한 기본적인 관리 활동을 정의한다.
- ② 기반 구조는 프로세스의 기반 구조를 확립하기 위한 기본 활동을 정의한다.
- ③ 개선은 프로세스의 확립, 측정, 통제, 개선을 위하여 수행하는 활동을 정의한다.
- ④ 교육 훈련은 적절하게 훈련된 요원을 제공하기 위한 활동을 정의한다.

2-2. 작성 산출물 적용

테일러링을 위한 품질 관리

2. CMMI 모델

- CMMI는 미국 카네기멜론대학의 소프트웨어기술연구소가 제안한 통합 모델이다. CMMI(Capability Maturity Model Integration) 모델은 조직의 개발 프로세스 역량 성숙도를 평가한다. CMMI 모델에는 6종이 있다.

〈표 2-22〉 CMMI 모델의 구분

구분	내용
SW-CMM	소프트웨어 능력 성숙도 모델
SECM	시스템 엔지니어링 능력 모델
IPD-CMM	통합 제품 개발 능력 성숙도 모델
People-CMM	인력의 개발과 관리
SA-CMM	소프트웨어 획득
SECAM	시스템 엔지니어링 능력 심사 모델

출처: 소프트웨어 공학, 2016년 1월 15일, 이형주 편저, (주)지식과 미래

2-2. 작성 산출물 적용

테일러링을 위한 품질 관리

2. CMMI 모델

- CMMI의 종류로는 단계적 모델과 연속적 모델이 있다. 단계적 모델은 조직의 시스템 개발과 관리 프로세스가 평가되어 1부터 5까지의 성숙도 수준을 할당한다. 연속적 모델은 더 세분화된 분류가 가능하며 24개의 프로세스 영역에 대해 0부터 5까지의 등급을 부여한다.
- CMMI의 프로세스 영역



테일러링을 위한 품질 관리

2. CMMI 모델

- **프로세스 관리 영역을 파악하여 적용한다.**
 - 프로세스 관리의 영역은 조직의 프로세스 정의, 조직의 프로세스 초점, 조직의 훈련, 조직의 프로세스 성과, 조직의 혁신과 배치로 구성된다.
- **프로젝트 관리 영역을 파악하여 적용한다.**
 - 프로젝트 관리의 영역은 프로젝트 계획 수립, 프로젝트 모니터링과 통제, 공급자와의 합의 관리, 통합된 프로젝트 관리, 위험 관리, 통합된 팀 구성, 정량적인 프로젝트 관리로 구성된다.
- **엔지니어링 영역을 파악하여 적용한다.**
 - 엔지니어링의 영역은 요구사항 관리, 요구사항 개발, 기술적 해결책, 제품 통합, 증명, 검증으로 구성된다.
- **지원 영역을 파악하여 적용한다.**
 - 지원의 영역은 형상 관리, 프로세스와 제품의 품질 경영, 측정과 분석, 의사 결정과 해결책, 통합을 위한 조직의 환경, 임시 분석과 해결책으로 구성된다.

2-2. 작성 산출물 적용

테일러링을 위한 품질 관리

3. SPICE 모델

- SPICE(Software Process Improvement and Capability dEtermination) 모델은 소프트웨어 프로세스 평가를 위한 국제 표준을 제정하는 프로젝트이다. SPICE의 목적은 3가지로 정리할 수 있다. 개발 기관이 프로세스 개선을 위하여 스스로 평가하는 것, 기관에서 정한 요구 조건을 만족하는 지 개발 조직 스스로 평가하는 것, 계약 을 맺기 위하여 수탁 기관의 프로세스를 평가하는 것이 그것이다.
 - SPICE은 CMM의 단점을 개선하기 위해 ISO 12207 소프트웨어 생명주기 프로세스를 참 고로 하며, 1995년에 ISO/IEC 155504라는 규격을 완성하였다. 프로세스 수행 능력 수준 단계는 6단계이다.
- ① CMM은 조직을 평가하므로 제품의 품질과는 직접적인 연관성이 없다.
 - ② CMM은 조직 전체에 대한 등급 판정이 비효율적, 비현실적이다.
 - ③ CMM은 소규모 업체에서는 적용이 곤란하다.

2-2. 작성 산출물 적용

테일러링을 위한 품질 관리

3. SPICE 모델

- 프로세스 수행 능력 수준 단계



2-2. 작성 산출물 적용

테일러링을 위한 품질 관리

3. SPICE 모델

- ① 수준 0, 불안정 단계: 프로세스가 구현되지 않았거나, 프로세스가 그 목적을 달성하지 못한 단계이다.
- ② 수준 1, 수행 단계: 프로세스의 목적이 전반적으로 이루어진 단계이다.
- ③ 수준 2, 관리 단계: 정의된 자원의 한도 내에서 그 프로세스가 작업 산출물을 인도한다.
- ④ 수준 3, 확립 단계: 소프트웨어공학 원칙에 기반하여 정의된 프로세스가 수행된다.
- ⑤ 수준 4, 예측 단계: 프로세스가 목적 달성을 위해 통제되고, 양적인 측정을 통해서 일관되게 수행된다.
- ⑥ 수준 5, 최적화 단계: 프로세스 수행을 최적화하고, 지속적으로 업무 목적을 만족시킨다.

2-2. 작성 산출물 적용

단계별 활동목적과 표준 산출물을 확정

- 개발준비단계의 활동목적과 표준 산출물 및 산출물의 필수여부를 확정한다.
- 개발준비단계의 활동목적은 사업을 수행할 때 발생하는 사항을 처리하고 발주 기관의 개발 방법론을 조정하기 위함이다.
- 개발준비단계는 사업 팀 구성과 방법론 테일러링의 작업으로 이루어진다.
- 필수는 ○, 선택은 △, 특성은 ▲

〈표 2-23〉 개발 준비 단계의 표준 산출물과 필수 여부

작업	입력물	산출물	필수 여부	
			대	소
사업 팀 구성	제안 요청서, 제안서, 사업 수행 계획서	팀 구성 계획서	○	○
방법론 테일러링	발주 기관 개발 방법론	방법론 테일러링 결과서	○	○

출처: 특허청(2014. 12) 소프트웨어 개발 방법론 p.23에서 요약

2-2. 작성 산출물 적용

단계별 활동목적과 표준 산출물을 확정

- 요구사항 분석의 활동목적과 표준 산출물 및 산출물의 필수 여부를 확정한다.
- 요구사항분석의 활동목적은 목표 시스템 구현을 위해 요구되는 사항을 분석함으로써 요구사항을 명확하게 정의하고 기술하기 위함이다.
- 요구사항분석 단계는 요구사항 수집, 요구사항 정의, 유스케이스 기술, 요구사항 추적을 한다.

〈표 2-24〉 요구사항 분석 단계의 표준 산출물과 필수 여부

작업	입력물	산출물	필수 여부	
			대	소
요구사항 수집	제안 요청서, 제안서, 사업수행 계획서	인터뷰 계획서, 인터뷰 결과서	○	○
요구사항 정의	제안서, 제안 요청서, 인터뷰 결과서	요구사항 정의서	○	○
유스케이스 기술	요구사항 정의서	유스케이스 명세서	○	○
요구사항 추적	요구사항 정의서, 유스케이스 명세서	요구사항 추적표	○	○

출처: 특허청(2014. 12) 소프트웨어 개발 방법론 p.31에서 요약

2-2. 작성 산출물 적용

단계별 활동목적과 표준 산출물을 확정

- 시험단계의 활동목적과 표준 산출물 및 산출물의 필수 여부를 확정한다.
- 시험단계의 활동목적은 구현 완료된 시스템에 대해 요구사항이 올바르게 반영되었는지를 통합적인 관점에서 검증하기 위함이다.
- 작업 내용은 테스트 준비 작업, 통합 테스트, 사용자 테스트이다.

〈표 2-25〉 시험 단계의 표준 산출물과 필수 여부

작업	입력물	산출물	필수 여부	
			대	소
테스트 준비 작업	총괄 테스트 계획서	-	○	○
통합 테스트	통합 테스트 시나리오	통합 테스트 결과서	○	○
사용자 테스트	요구사항 정의서, 사용자 인터페이스 설계서	사용자 테스트 결과서	○	○

출처: 특허청(2014. 12) 소프트웨어 개발 방법론 p.93에서 요약하여 제시

2-2. 작성 산출물 적용

방법론 테일러링 결과서를 확정

- 방법론 테일러링 결과서의 작성 예시
- 필수는 ○, 선택은 △, 특성은 ▲

개발 방법론				적용 여부	(사업명) 방법론				테일러링 사유
단계	활동	작업	산출물		단계	활동	작업	산출물	
업무/ 데이 터 분석			To-Be비즈니스 스 프로세스 정의서	○	○	○	○	○	필수
			To-Be비즈니스 스 업무 흐름 도	○	○	○	○	○	필수
			현행 데이터 분석서	△	△	△	△	△	선택
			현행 표준사 전 정의서	△	△	△	△	△	선택
데이터 분석			전환대상 업 무 및 범위 정의서	▲	▲	▲	▲	▲	특성

Unit A

참고자료



문헌

1. <http://www.ncs.go.kr>
2. 애자일SW개발101,nipa
3. 방법론 가이드,SW중소기업을 위한 경량 개발 방법론, 과학기술정보통신부, nipa
4. 김치수(2015). 『쉽게 배우는 소프트웨어 공학』. 한빛아카데미(주)
5. 김향곤, 소프트웨어개발 방법론, 대구 카톨릭대학 • 기술표준원(2013. 11)
6. 프로젝트 관리기준(ISO 21500) 이행가이드
7. 과학기술정보통신부 보도자료(2016. 9. 6)
8. 상용 소프트웨어 현재와 미래 조망
9. 보건복지부(2013).
10. 지식경제부(2010.2.26.). 소프트웨어 사업대가의 기준.
11. 정보통신기술진흥센터(2016. 9. 6). 글로벌 상용 소프트웨어 백서.
12. 특허청(2014). 소프트웨어개발 방법론.
13. 한국소프트웨어산업협회(2016). 소프트웨어 사업 대가 산정 가이드.
14. CMMI (CMMI(Capability Maturity Model Integration) 모델
15. ISO/ IEC 12207 모델 (소프트웨어 개발 생명주기 프로세스)
16. ISO/ IEC 14598 품질 평가 프로세스 • SPICE (ISO 15504) 모델 (프로세스 평가 표준)
17. 린 소프트웨어 개발의 적용
18. 속도 경쟁에서 승리하기
19. 메리 포펜딕, 톰 포펜딕 공저 / 엄위상, 심우곤, 한주영 공역 | 위키북스 | 2007년 08월 31일 | 원제 : Implementing Lean Software Development: From Concept To Cash
20. T 아카데미
21. <https://www.lesstif.com/jira/>
22. 제프랭어, 앤디 헌트, 데이브 토마스, 유동환 역,자바와 JUnit을 활용한 실용주의 단위 테스트,길벗

감사합니다.

❖ Mobile: 010-9591-1401

❖ E-mail: onlooker2zip@naver.com / dkkim@lklab.org

