

## 8주 1강

# 실전 적용 및 최신 프로그램 언어 소개



# 이번 주차에는...

## 실전 적용 시뮬레이션

- 계획서 작성 예시
- 산출물의 종류
- UML의 기본 개념 2
- 기본 자바 소개 1

# 1. 요구 분석 명세서

## ■ 요구 분석 명세서의 기재 내용

항목	내용
ID	내부 규칙에 따라 식별자를 부여한다. 하나의 요구사항에 하나의 식별자이다.
화면 명	어느 화면에서 구현할 기능인지 기재한다. 화면에 속하지 않는 요구사항도 있을 수 있다.
요구 사항 항명	요구사항의 설명을 요약하여 기재한다.
내용	요구 사항의 상세한 내용을 기재한다.
중요도	상중하, 1~5 등 내부 규칙에 따라 부여한다.
부서/작성자	요구사항을 기재한 담당자를 기재한다. 부서가 없는 경우 생략해도 무방하다.
날짜	요구사항을 기재한 날짜를 명시한다.
진행사항(구현 여부)	검토 예정, 진행 확정, 진행 불가, 추후 진행 등 내부에서 결정된 사항을 기재한다. 진행 불가이거나 추후 진행 일 경우 이유를 같이 적는 것이 좋다.
버전 명	요구사항이 변경될 수 있으므로 버전(혹은 다른 방식)으로 표기하여 타인이 알 수 있게 해야 한다.
그 외	유형(기능, 비 기능, 제약사항, 인터페이스, 기타 등), 출처(관련자 이름, 11/15 회의, 사업계획서 등)

## 2. 좋은 요구 사항 명세서의 조건

- 요구사항 명세서를 읽는 작업자(개발자, 디자이너 등)가 이해하기 쉬워야 한다.
- 무엇을 어떻게 구현되어야 하는지 명확하게 작성한다.
- 하나의 요구사항에 여러가지(복수) 요구사항을 작성하지 않는다.
- 다른 요구사항 모순 또는 중복되지 않게 한다.
- 애매한 단어를 사용하지 않고 명확하게 기재한다. ( ~ 있으면 좋겠다 → ~ 기능 필요)  
의견을 모호 하게 하지 말고 명확하게 표현 해야 한다.
- 꼭 필요하고 중요한 요구사항은 표시하는 것이 좋다. 다만 모든 요구사항에 남발 해서  
는 안된다.
- 동일한 용어를 사용하며 통일성이 있어야 한다. 댓글, 코멘트, 덧글과 같이 모두 같은  
의미를 다르게 표현하지 말라는 의미이다.
- 난이도가 있는 기능이거나 프로젝트 기간이 짧아 모든 요구사항을 구현하기 어려울 상  
황일 경우, 우선 순위를 정하고 대체 가능한 다른 방법도 함께 기술하여 전달하는 것이  
좋다.

### 3. 요구 분석의 문제점

- 의사 소통의 문제
  - 샘플이 없어 요구 설명이 어려움
  - 사용자가 요구 사항에 대한 설명 방법을 잘 모름
  - 일관성이 없거나 불안정한 요구 분석 명세서
- 요구 사항의 증가 및 변경
  - 초기 애매 모호한 요구사항
  - 분석가가 없이 시작한 프로젝트
  - 분석가 : 사용자와 개발자 사이에서 둘이 원활한 협의를 할 수 있도록 도와주는 사람

## 4. 프로젝트 계획서(1)

### ■ 목차

- 개요
  - 프로젝트 개요
  - 프로젝트의 산출물
  - 정의, 약어
- 자원 및 일정 예측
  - 자원
    - ✓ 인력
    - ✓ 비용
  - 프로젝트 일정
- 조직 구성
- 기술 관리 방법
  - 변경 관리
  - 위험 관리
  - 문제점 해결 방안

## 5. 프로젝트 계획서(2)

### ■ 목차

- 조직 구성
- 기술 관리 방법
  - 변경 관리
  - 위험 관리
  - 문제점 해결 방안
- 검토 회의
  - 검토 회의 일정
  - 검토 회의 진행 방법
  - 검토 회의 후속 조치
- 개발 환경
- 성능 시험 방법
- 인수 시험
- 참고 문헌

## 6. 개요

### ■ 프로젝트 개요

- 제목
  - 파이썬 자격 검증 시험 사이트
- 개요
  - 현재 IT 프로그램 언어 중에서 가장 핫한 언어인 파이썬에 대한 자격 검증 시험 사이트를 CBT 개념으로 개발하여 파이썬 자격증 시험을 실행 하고자 한다. 파이썬 자격 검증 시험은 1급과 2급으로 구분하며 각각에 대하여 필기와 실기를 진행하며 시험 시간은 2급인 경우 필기 60분, 실기 60분이며 1급인 경우는 필기 60분, 실기 120분을 진행한다.



## 7. 프로젝트 산출물(1)

- 요구 분석
  - 요구 사항 정의서
  - 자료 사전
  - 자료 흐름도
  - 사용자 인터페이스
  - 소단위 명세서
  - 자원, 인력에 대한 제약 조건
- 기본 설계
  - 매뉴얼 작성
  - 출력 설계
  - 시스템 구조 설계
  - 프로그램 설계
  - DB 설계

## 8. 프로젝트 산출물(2)

---

- 인터페이스 설계
  - 디자인
  - 레이아웃
  - 링크 구성
- 구현
  - 데모 사이트

## 9. 자원 및 일정 예측

### ■ 자원

#### • 인력

- 팀 구성원 5명(김장현, 홍길동, 최진영, 유혁)

#### • 비용

- 인건비 :  $3,000,000/\text{월} * 4\text{명} * 3\text{달} * 0.5 = 18,000,000\text{원}$
- 기자재 비 : 노트북 \* 4대 \* 2,000,000원 = 8,000,000
- 인쇄비 : 20,000원 \* 30회 = 600,000원
- 여비 : 50,000원 \* 12회 = 600,000원
- 회의비 : 40,000원 \* 12회 = 480,000원

#### • 총 비용

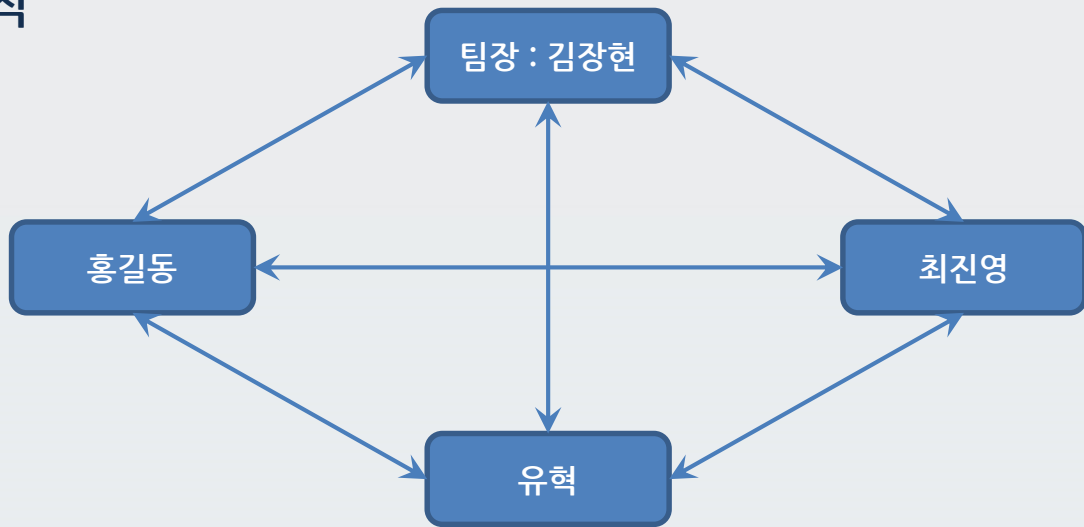
- 인건비+여비+기자재비+인쇄비+회의비
- → 27,680,000원

# 10. 프로젝트 일정 관리표

소작업명	담당자	10월				11월				12월				비고
		1	2	3	4	1	2	3	4	1	2	3	4	
1. 계획														10/1 ~ 10/23
프로젝트 팀 구성	김장현													
목표 및 문제정의	팀 전원													
비용, 일정예측	팀 전원													
계획서 검토회의	팀 전원													
2. 요구분석														10/25 ~ 10/27
요구정의	팀 전원													
자료사전	팀 전원													
소단위 명세서	팀 전원													
요구분석 검토회의	팀 전원													
3. 설계														10/27 ~ 10/30
매뉴얼 작성	김장현													
Form 설계	홍길동													
출력설계	유혁													
시스템 구조설계	최진혁													
프로그램 설계	팀 전원													
UI 설계	팀 전원													
분석서 및 검토회의	팀 전원													

# 11. 조직 구성

## ■ 분산 형 팀 조직



- 프로젝트 조직 구조는 분산 형 팀 조직으로 구성함
- 팀장 1명, 팀원 3명으로 구성
- 팀장을 포함한 팀원 모두 개발자로 설계와 프로그래밍을 담당

## 12. 기술 관리 방법

### ■ 변경 관리

- 의사 소통의 창구를 통해 일정 기간 의견을 수렴하고 우선 순위 순으로 변경 및 업그레이드를 지향한다.
- 파이썬의 기능 및 버전이 계속적으로 업그레이드 됨에 따라 그에 맞는 정보를 제공하고 이번 버전과 현 버전의 차이점을 확인 시켜주며 지속적인 업데이트가 중요하다.

### ■ 위험 관리

- 최대한 개발 시 발생할 수 있는 문제점을 분석하여 위험 요소를 차단한다.
- 실제적으로 시험에 따른 다양한 위험 요소를 분석한다.
- 비용 및 인력에 대한 위험 요소를 분석하고 해결 방안을 준비한다.

### ■ 문제점 해결 방안

- 위험 요소 발견 시 팀원들과의 회의를 통해 최선의 해결책을 찾아 해결한다.
- 위험 요소의 우선 순위를 정해 우선 순위가 높은 것부터 제거해 나간다.

## 13. 검토 회의

- 검토 회의 일정
  - 회의 일정 : 매주 월요일 오전 10시
  - 회의 장소 : 회사내 제 1 회의실
  - 회의 내용 : 개발 이슈, 클라이언트 이슈, 팀 문제, 팀원들 개개인의 문제 순으로 진행
- 검토 회의 진행 방법
  - 이전 주에 진행했던 개발에서의 위험 요소를 분석하고 이슈화 한다.
  - 각각의 개발 진척도를 발표하고 일정보다 늦어졌으면 늦어진 이유와 해결 방안을 논의한다.
- 검토 회의 후속 조치
  - 검토 회의 후 나온 안건은 그 주 최대한 마무리한다.
  - 이후 해결 방안을 자료로 남기며 이 자료를 토대로 산출물을 만들어 둔다.

# 14. 개발 환경(1)

- 서버
  - 웹 서버 : Linux 서버(Ubunto 18.04.3 LTS)
  - DB 서버 : MariaDB(MariaDB 10.4.8 Stable)
  - 프로그램 언어 : JAVA, JSP, Servlet
- 데이터베이스 모델링
  - DB 모델링 툴
    - 종류 : Erwin, DA#
    - 장점
      - ✓ 데이터베이스 Schema 설계를 진행 할 때 손쉬움
      - ✓ 논리적 설계에서 물리적 설계 시 사용자 편의성과 가독성을 극대화 한다.
    - 단점
      - ✓ 대부분 사용 되어 있는 툴로 가격이 고가로 구입하기 쉽지 않다
    - 제품들
      - ✓ Erwin : <https://erwin.com/> , <https://www.ca.com/kr/company.html>
      - ✓ eXERD : <https://techdom.tomatosystem.co.kr/p/00016/>
      - ✓ DA# : [http://dataware.kr/solution/new\\_idx](http://dataware.kr/solution/new_idx)



# 15. 개발 환경(2)

- DB 관리 툴

- 종류 : Toad, DBA Master2000, HeidiSQL, 등
- 장점 : 대상 DB의 물리 Schema를 손쉽게 접근하여, 해당 결과를 빠르게 확인 할 수 있다.
- 단점 : Toad와 같은 상용 프로그램에서 프리웨어나 무료 사용시 기능 제한이 있다.
- 제품들
  - ✓ Toad : <http://www.toadworld.com/> (사용, 트라이얼버전, 제한적 무료 버전)
  - ✓ HeidiSQL : <https://www.heidisql.com/> (무료)
  - ✓ Golden6 : <https://www.benthicsoftware.com/downloads.html> (상용, 트라이얼버전)
  - ✓ DbVisualizer : <https://www.dbvis.com/> (상용, 무료 버전)
  - ✓ Aqua data studio : <https://www.aquafold.com/aquadatastudio> (상용, 트라이얼버전)
  - ✓ orange : <https://www.warevalley.com/solutions/orange.asp> (상용, 트라이얼버전)
  - ✓ sqlgate : <https://www.sqlgate.com/?language=ko> (상용, 제한적 무료 버전)
  - ✓ pl/sql developer : <https://www.allroundautomations.com/> (상용, 트라이얼버전)
  - ✓ Dbeaver : <https://dbeaver.io/download/>

# 16. 개발 환경(3)

- 프로그램 개발 툴

- 종류 : Eclipse, Visual Studio, VSCode, Android Studio, UltraEdit, EditPlus 등

- 특징

- ✓ 프로그램 개발을 위해 다양한 툴을 사용할 줄 알아야 한다.

- ✓ 각각의 프로그램 언어마다 맞는 개발 툴이 존재한다.

- ✓ 가벼운 텍스트툴을 다룰 수 있어야 함

- 제품들

- ✓ Visual Studio : <https://visualstudio.microsoft.com/ko/> (상용, 무료)

- ✓ Android Studio : <https://developer.android.com/studio/intro?hl=ko> (무료 : 안드로이드 개발 툴)

- ✓ Eclipse : <https://www.eclipse.org/downloads/> (무료 : 자바 개발 및 다양한 언어 개발툴, Plug-in)

- ✓ VSCode : <https://code.visualstudio.com/> (자바 스크립트 계열 개발, Vue.js, Node.js, React.js 등)

- ✓ UltraEdit : <https://www.ultraedit.com/> (텍스트 에디터)

- ✓ EditPlus : <https://www.editplus.com/> (텍스트 에디터)

# 17. 성능 시험 방법(1)

## ■ 시험 방법

- 자격 검증 시험의 특성에 따라 문제 은행에 문제를 저장하고 실제적으로 실행하듯이 아래와 같이 테스트 한다.
  - 시험 신청 → 시험 시작 → 시간 측정 → 시험 마침 → 시험 제출 → 결과 보기
- 요구 분석 명세서의 내용과 비교하여 테스트한다.
- 사용자의 요구에 맞게 SW가 만들어 졌는지 테스트한다.
- 코드에서 대상 시스템이 아닌 다른 쪽의 부하가 발생하거나 에러가 발생하는지, 정확한 페이지가 호출되는지를 Log를 통해 확인한다.
- 서버 응답시간, 데이터베이스 질의 응답시간등을 테스트한다.
- 어떠한 성능 시험 도구를 쓸것인가?
- 대상 방문자는 누구인가, 어떤 브라우저를 쓸 것인가,
- 클라이언트 사이드는 어느 정도의 성능을 기대하는가?

## 18. 성능 시험 방법(2)

- 서버와 콘텐츠를 업그레이드 하기 위한 정지 시간은 허용되는가? 허용된다면 얼마동안인가?
- 어떤 종류의 보안이 적용되는가? (방화벽, 암호화, 패스워드 등) 그리고 각각에 기대되는 기능은 무엇이며, 어떻게 시험할 것인가?
- 해당 사이트의 인터넷 접속의 신뢰도는 어떤 수준이 요구되는가?
- 백업시스템이나 보조 시스템이 필요한가? 이들도 시험하는가?
- 웹사이트의 콘텐츠를 관리하는 절차는 무엇인가?
- 페이지의 콘텐츠와 그래픽등을 추적/관리하는데 요구되는 사항은 무엇인가?
- 어떤 HTML사양을 적용할 것인가?
- 대상 브라우저에 따라 얼마의 변형이 허용되는가?
- 전체 사이트 또는 특정 페이지 그룹에 적용되는 페이지의 표현이나 그래픽에 관한 표준이나 요구사항이 있는가?

## 19. 성능 시험 방법(3)

- 어떻게 내부와 외부의 링크를 검증하는가? 그리고 얼마나 자주 하는가?
- testing이 작동하는 시스템에서 수행되는가? 아니면 별도의 testing 시스템이 요구되는가?
- 서버 로깅과 Reporting에 대한 요구는 어느 정도인가? 그리고 이러한 사항들이 전체 통합 시스템의 일부로 간주 되는가?
- 이 부분도 시험되어야 하는가?

## 20. 인수 테스트

### ■ 인수 테스트

- 플랫폼 설치를 하기 전 인수 시험을 실시한다.
- 각 모듈들이 정의된 인터페이스에 따라 잘 동작하는지 통합 시험 (integration test)을 수행한다.
- 여러 모듈을 조금씩 결합하면서 단계적으로 시험하는 방법을 취한다.
- 각각의 주어진 기능들이 오류 없이 수행 되는가의 시험한다.
- 스크립트에서 대상시스템이 아닌 다른 쪽의 부하가 발생하거나 에러가 발생하는지, 정확한 페이지 호출이 되고 있는지를 Log를 통하여 확인 한다.

# 21. 산출물의 종류(1)

## 1. 분석 단계

### 1. 요구사항 정의서

- 고객의 니즈를 담고 있는 문서, 고객의 니즈에 따라서 프로젝트의 스펙이 결정될 것임

### 2. 기능차트

- 니즈를 베이스로 큰 카테고리를 만들어서 프로젝트의 역할을 보여준다.

### 3. 프로세스 정의서

- 기능 차트를 기준으로 각각의 프로세스를 보여준다.

### 4. 인터페이스 정의서

- 위 문서를 베이스로 웹서비스에서 어떤식으로 인터페이스를 한다는 정의서

## 22. 산출물의 종류(2)

### 1. 설계 단계

#### 1. 화면 설계서

- 웹 어플리케이션 혹은 어플리케이션에서 고객이 사용하는 샘플 화면

#### 2. ERD(DB 모델링 자료)

- 데이터베이스를 생성하고 테이블 간의 관계를 나타내는 문서

#### 3. 테이블 정의서

- 각 테이블의 필드명 값, 바이트등의 정보 표시

#### 4. 개발 표준 정의서

- 변수명, 클래스명, 파일명의 규칙등 코딩에 관련된 규칙을 담은 문서

#### 5. 단위 테스트 시나리오

- 프로세스별 단위 테스트를 위한 단위 테스트 문서

#### 6. 통합 테스트 시나리오

- 단위 테스트를 근간으로 고객의 니즈를 보안할 수 있는 테스트 문서



## 23. 산출물의 종류(3)

### 1. 개발 단계

#### 1. 소스 코드

- 오류 수정까지 끝난 원시 코드 자체

#### 2. 단위 테스트 결과

- 단위 테스트 시나리오를 기준으로 한 테스트 결과

#### 3. 결함/오류 보고서

- 단위테스트를 통해 발견된 에러/버그의 원인과 수정에 대한 내용

#### 4. 통합 테스트 결과

- 단위테스트의 보안된 내용을 포함하면서, 통합 테스트 시나리오를 기준으로 한 결과로서 개발완료 여부의 기준이 되는 문서

## 24. 산출물의 종류(4)

### 1. 구현 단계

#### 1. 시스템 이행 결과서

- 시스템 이행 계획서를 통해 이행된 결과를 확인 받는 문서

#### 2. 사용자 매뉴얼

- 사용자 화면이 존재할 경우, 일반적인 조작법과 화면등의 예시를 들어 산출되는 문서

#### 3. 운영자 매뉴얼

- 개발된 프로젝트 시스템 전방에 대한 내용을 담는 문서

#### 4. 교육(인수)명세서

- 사용자 매뉴얼, 운영자 매뉴얼을 중심으로 담당자에게 세부사항을 교육/인수한 수 받는 문서

#### 5. 개발 산출물 별 검사 리스트

- 산출물들의 이상여부, 인수 여부 체크 후 받는 문서

#### 6. 프로젝트 완료 보고서

- 최종적으로 개발된 내용, 인도물, H/W, 고객사 대표, 개발사 대표의 확인이 들어간 중요 문서



다음 시간

# UML의 기본 개념 2



송실사이버대학교

송실사이버대학교의 강의콘텐츠는  
저작권법에 의하여 보호를 받는바, 무단  
전재, 배포, 전송, 대여 등을 금합니다.

\* 사용처: 나눔글꼴