



2025-2학기


UNIFIED  
MODELING  
LANGUAGE™



객체 지향 설계와 분석을 위한  
**UML  
기초와 응용**



DevOps  
**소프트웨어 공학  
이론과 실제**



# 게임소프트웨어공학(GSE)

## [2주차-1] 이론 1장 소프트웨어공학(SE) 개요

1. 소프트웨어 고장 사례
2. 소프트웨어 위기
3. 소프트웨어공학 기술의 적용
4. 과제/진도


 한국공학대학교


★이 강의자료는 여러분을 위한 담당교수의 경험/생각과 한빛아카데미㈜에서 제공한 교재 자료를 활용해서 작성하였습니다★

1

## [이론 1장] 소프트웨어공학(SE) 개요

- 학습 목표
  1. 최근 소프트웨어의 오류로 인하여 발생한 대표적인 사고(고장) 사례 이해
  2. 소프트웨어 개발에는 어떠한 문제들이 존재하는지, 왜 소프트웨어 개발이 어려운지 이해
  3. 소프트웨어공학이 무엇인지 이해
  4. 개발과제 수행방법, 주제 선정 및 문제기술서(SOP) 작성



 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

2



UNIFIED  
MODELING  
LANGUAGE™

이론교재

**1장**

소프트웨어공학  
개요



1. 소프트웨어 고장 사례

2. 소프트웨어 위기

3. 소프트웨어공학 기술의 적용

4. 과제/진도



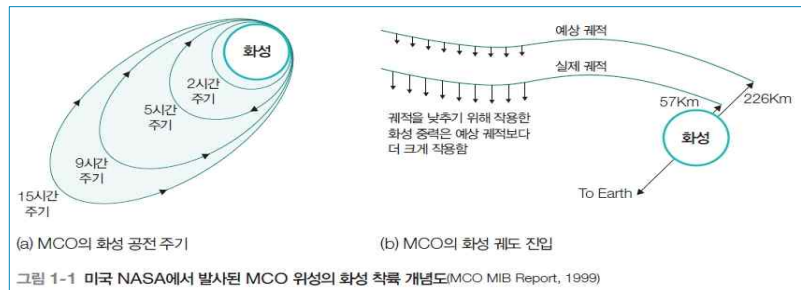
한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

3

## 화성 기후 관측 위성 결함 사례

- MCO(Mars Climate Orbiter) 위성
  - 1998년, 화성의 대기 탐사 목적으로 개발, 화성 궤도 진입 과정에서 통신 두절
- MCO 위성 결함 분석 결과
  - 발견된 사실
    - 위성 속도 변화 계산 프로그램(SM\_FORCE)에 사용된  $\Delta V$ (순간속도) 변수가 정상 값보다 4.45배 작게 계산된 사실 발견
  - 원인 분석
    - MCO의 추력 데이터 단위를 국제 표준단위인  $1N \cdot s$ (뉴턴초)로 표시하라고 명세
    - SM\_FORCE 프로그램에 미국식  $4.45lb \cdot s$ (파운드·초)를 사용하여 구현



## 유사 사례: 프랑스의 아리안 5호 로켓 발사 실패 사례

### ■ 프랑스의 아리안 5호 로켓 발사 실패는 SW 코드 복사가 불러온 재앙!

- 1996년 4월 프랑스령 남미 기아나의 우주기지에서 발생
- 약 5억 달러(약6,000억원) 프로젝트 유럽 상업용 우주선인 아리안 5호가 발사 후 약 40초 만에 공중에서 폭발한 사고
- 유럽우주기구(Europe Space Agency)와 프랑스 우주연구센터의 사고 조사결과 폭발 원인은 에이다(Ada) 언어로 작성된 SW 내부에 존재하는 버그로 밝혀짐
- 사고는 64비트 숫자 값을 16비트 정수로 변환하는 과정에서 발생, 문제가 된 모듈은 데이터 타입 오류로 산술적 오버플로우(Numeric Overflow)가 원인이며,
- 오버플로우 값은 고도값으로 우주선을 통제하는 중앙 SW에 전달되어 통제 SW는 자동발사장치를 작동하여 발생(프로그래머의 일상적인 Copy & Paste가 문제)
- 주목할 점은 폭발 사고를 일으킨 SW가 우주선의 운항과 직접 연관성이 없다는 데서 더 큰 충격!
- 이 사고는 SW에 숨어 있는 버그의 위력 실감할 수 있는 사례!



## Jeep 체로키 차량의 해킹 사례

### ■ Jeep 체로키 해킹 사례

- 2014년 밀러와 발라섹이 **운행 중인 체로키 Jeep 차량에 무선 원격 접속**
- Jeep 체로키 제어 시스템의 BCM과 CAN-C간의 버스 데이터 모니터링, 수집 및 분석
- **차량의 자세 제어 및 동작 명령을 전송해서 운행 중인 차량의 급정지를 시도**

### ■ Jeep 체로키 차량의 결함 분석

- 제어시스템의 Radio 모듈에 앱(App) 다운로드 설치
- **온라인 모니터링 포트를 이용한 프로그램 수정 가능**
- 소프트웨어 패치를 위해 140만대의 자동차를 리콜 조치

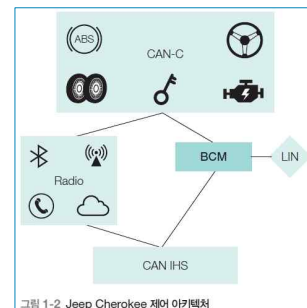


그림 1-2 Jeep Cherokee 제어 아키텍처

보잉 737 MAX8 추락 사례

- 보잉 737 MAX8의 탄생
  - 승객 탑승 공간을 확장하기 위해 기존 보잉 737 기종을 형상 변경
  - 기체 변동으로 엔진의 위치 변경하였지만 기존 소프트웨어는 변경없이 재사용
  - MCAS(Maneuvering Characteristics Augmentation System) 설치를 통한 자세 제어
- 보잉 737 MAX8의 추락
  - 2018년 자카르타를 출발한 아이언 에어 610편 추락
    - 이륙 12분만에 자바 해변에 추락, 189명 사망
  - MCAS에 입력되는 노즈 센서 값의 오류로 노즈 하강
  - 항공기가 조종사에 의한 수동 조작 거부로 인한 자세 제어 불가
    - 위급 상황에서 조종사로의 제어 복귀 기능 부재

MCAS 작동원리

이론 1장 소프트웨어공학(SE) 개요

7

기타 소프트웨어 결함 사고

- 테슬라 자율주행 자동차의 주행 사고
  - 2016년 5월, 테슬라 자율주행 자동차가 미국 플로리다주 고속도로에서 트럭의 측면과 충돌하는 사고 발생
  - 흰색 트럭과 구름 낀 하늘과의 인식 오류로 발생
- 폭스바겐 자동차의 조립 공정 사고
  - 차량 조립 과정에서 로봇 팔의 오동작으로 엔지니어의 사망사고 발생
  - 작업 동작 범위 내에서의 충돌 예방 기능에 대한 프로그램이 부재
- 도요다 자동차의 급발진 사고
  - 2009년 8월, 미국 캘리포니아 샌디에고 고속도로에서의 급발진 사고
  - 자동차 연료주입 제어 장치의 프로그래밍 오류로 브레이크 기능 차폐

이론 1장 소프트웨어공학(SE) 개요

8



UNIFIED  
MODELING  
LANGUAGE™

이론교재

1장

소프트웨어공학  
개요



1. 소프트웨어 고장 사례
2. 소프트웨어 위기
3. 소프트웨어공학 기술의 적용
4. 과제/진도/Q&A


 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

9

## 소프트웨어 위기의 등장

- 소프트웨어 위기(Crisis)
  - 1968년 NATO 소프트웨어공학 학회에서 처음 등장
  - 컴퓨터에 의한 계산 용량과 문제의 복잡도가 급증하면서, 새로운 소프트웨어 개발 방법의 필요성을 인식
- 소프트웨어 위기의 원인
  - 소프트웨어 규모의 대형화 및 복잡도 증가에 따른 개발 비용 증대
  - 소프트웨어 유지보수의 어려움과 개발 정체 현상 발생
  - 소프트웨어 개발 프로젝트 기간 및 소요 예산에 대한 정확한 예측의 어려움
  - 신기술에 대한 교육 및 훈련의 부족
  - 사용자의 소프트웨어에 대한 기대치 증가
  - 소프트웨어에 대한 사용자 요구사항의 빈번한 변경 및 반영

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

10

## 소프트웨어 개발 프로젝트 통계

### ■ 소프트웨어 개발 프로젝트의 성공과 실패율



그림 1-7 2000년대 소프트웨어 개발 프로젝트의 통계 분석 결과(출처: Standish Group, CHAOS Report, 2015, USA)

## 왜 소프트웨어 개발은 어려운가?

### ■ 의사소통의 문제

- 소프트웨어 개발 과정에 참여하는 다양한 역할자(프로젝트 관리자, 품질 관리자, 개발자, 사용자 등)간의 의사 소통 오류

### ■ 시스템의 순차 특성

- 3차원적인 실세계의 서비스를 2차원 평면에서 프로그램 코드로 나열

### ■ 개발에 의한 결과물

- 조립이나 공식에 의한 문제 풀이가 아닌 개발자의 지적 활동에 의한 산출물

### ■ 프로젝트 복잡성

- 프로젝트마다 개발 기간, 개발자 수, 사용자 수준 등의 차이로 인하여 유연한 관리 필요

### ■ 다양한 관리 이슈

- 요구사항 변경관리, 일정관리, 코드 버전관리 등 다수의 활동 간의 오케스트레이션

## 소프트웨어 고장 그래프

### ■ 하드웨어 고장 그래프

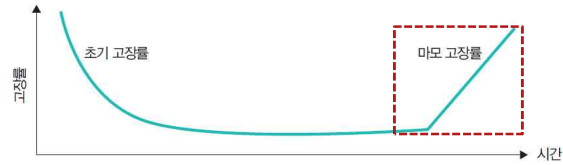


그림 1-11 하드웨어 수명주기에 따른 고장률 그래프

### ■ 소프트웨어 고장 그래프

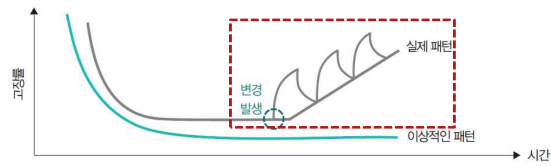


그림 1-12 소프트웨어 수명주기에 따른 고장률 그래프

## 인공지능(AI) 시대의 소프트웨어 위기

### ■ 4차 산업혁명 시대

- 초연결(Hyper-connected), 초융합(Hyper-convergence), 초지능(Hyper-Intelligence)
- 인공지능, 빅데이터, 가상현실, IoX(Internet of Everything), 사이버물리 시스템

### ■ 인공지능(AI) 소프트웨어의 범람

- 인공지능을 기반으로 하는 다양한 학습 모델 출현
  - 지도학습, 준지도학습, 비지도학습, 강화학습 등 다양한 기계학습 알고리즘 활용 증대
- 인공지능과 기계학습 기반의 소프트웨어 개발 및 활용 증대
  - 정보 서비스 분야, 자율 제어 분야, 모바일 분야 등 활용 영역 증대
- 인공지능 소프트웨어의 위기
  - 기계학습 결과에 대한 정확성, 신뢰성, 안전성 등의 문제
  - 윤리 및 도덕적인 영역에서의 인공지능의 편향성 문제



이론교재

1장

소프트웨어공학  
개요



1. 소프트웨어 고장 사례

2. 소프트웨어 위기

3. 소프트웨어공학 기술의 적용

4. 과제/진도/Q&A

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

15


소프트웨어공학적 기법

■ 구조적 프로그래밍: 1970년대

- 포트란, 코볼, C 등의 절차적 프로그래밍 언어의 등장
- 이해하기 쉽고, 체계적인 논리를 표현할 수 있는 공학적 접근 방법
- 구조적 분석 및 구조적 설계 방법론: 자료흐름도, 구조차트 등
- 모듈화 개념과 단계적 상세화 개념

■ 객체지향 프로그래밍: 1980년대

- 클래스 개념의 출현: 캡슐화, 정보은닉, 상속, 다형성
- C++, C#, Java, Python 등의 언어 출현
- 실세계의 묘사가 직관적이며, 재사용을 강조하는 공학적 접근 방법
- UML 기반 객체지향 분석 및 설계

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

16



## 소프트웨어공학적 기법

### ■ 소프트웨어 컴포넌트와 재사용: 1990년대

- 소프트웨어 컴포넌트: 재사용 가능한 명세 및 코드의 묶음
- 재사용으로 인한 개발의 신속성 및 품질 향상
- 컴포넌트 기반 소프트웨어 개발 방법론
- 오픈소스 소프트웨어(OSS) 재사용을 통한 소프트웨어 개발 확대

### ■ 기타 공학적 기법들

- 통합 개발 환경: 개인 혹은 팀 개발 활동을 지원하는 통합된 개발 도구
- 소프트웨어 개발 프로세스: 개발 과정의 체계화 및 관리 효율성
- 소프트웨어 검사 및 검증: 오류, 결함의 탐지 및 제거를 통한 품질 향상
- 소프트웨어 형상관리: 빈번한 변경 발생에 대응하기 위한 개발 과정 지원

## 소프트웨어공학의 정의

### ■ 소프트웨어공학(SE)이란?

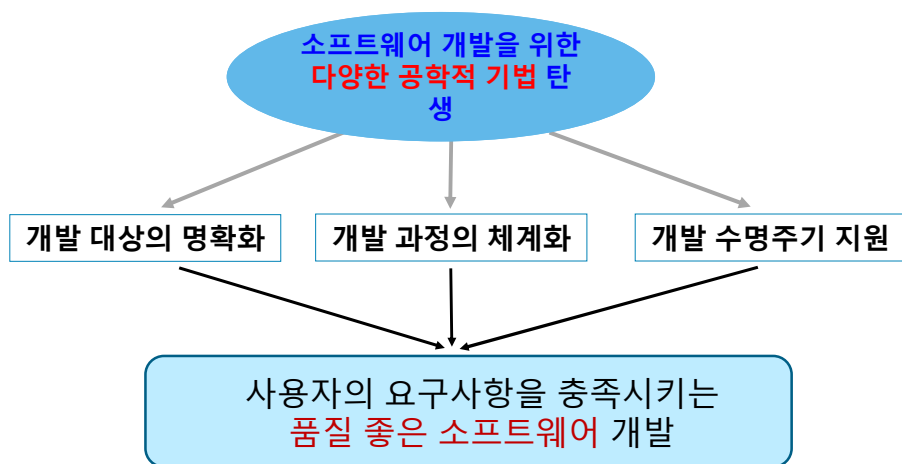
번호	제안자(연도)	소프트웨어 공학의 정의
1	Bauer (1972)	Deals with establishment of sound engineering principles and methods in order to economically obtain software that is reliable and works on real machines. (실제 환경에서 신뢰성 있는 소프트웨어를 경제적으로 확보하기 위한 건전한 공학적 원리와 방법의 구축 및 적용)
2	Boehm (1976)	Practical application of scientific knowledge in the design and construction of computer program and the associated documentation required to develop, operate, and maintain them. (컴퓨터 프로그램을 개발, 운영, 유지보수하기 위해 요구되는 프로그램의 설계 및 구축과 관련된 문서화에 대한 과학적 지식의 실질적인 적용)
3	Zelkowitz (1978)	Study on the principles and methodologies for developing and maintaining software systems. (소프트웨어 시스템을 개발하고 유지보수하기 위한 원리 및 방법에 대한 연구)

## 소프트웨어공학의 정의

### ■ 소프트웨어공학이란?


번호	제안자(연도)	소프트웨어 공학의 정의
4	Parnas (1978)	Multi-person construction of multi-version software. (소프트웨어의 여러 버전에 대한 여러 사람에 의한 개발)
5	Pfleeger (1990)	Methods and techniques to develop and maintain quality software to solve problem. (주어진 문제를 해결하는 품질 좋은 소프트웨어를 개발, 유지보수하기 위하여 적용 되는 방법 및 기법)
6	ISO 24765 (2010)	Systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software. (소프트웨어의 설계, 구현, 테스트, 문서화를 위한 과학적이고 기술적인 지식, 방법, 경험의 체계적인 적용)

## 소프트웨어공학의 목표



## 소프트웨어공학의 원리

- **엄격성과 정형성**
  - 소프트웨어는 개발자의 경험과 지식에 의존적인 창의적, 공학적 활동의 산출물
  - 소프트웨어 개발은 주어진 시간과 비용에서 명확하게 개발되어야 하는 엄격성
- **관심사의 분할**
  - 복잡한 문제를 단순한 문제로 분리하여 적용하는 소프트웨어 개발 활동
  - 소프트웨어 개발 과정의 분할: 요구사항 분석 > 설계 > 구현 > 테스트 등의 단계로 분할
  - 소프트웨어 테스트 과정의 분할: 단위 테스트 > 통합 테스트 > 시스템 테스트 등으로 분할
- **모듈화**
  - 독립적인 기능을 갖는 프로그램의 조작
  - 높은 응집력과 낮은 결합력을 갖는 소프트웨어 구조 설계
  - 이해도를 높이고 변경 영향을 최소화하기 위한 공학적 원리


한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

21

## 소프트웨어공학의 원리

- **추상화**
  - 세부사항은 감추고 대표적인 속성으로 대상물을 정의하는 공학적 원리
  - 대상물에 대한 직관적인 이해를 높이고 관심사를 잘 표현할 수 있음.
  - 예: 함수 정의, 매크로 함수, 객체, 추상 데이터 타입 등
- **변경의 예측**
  - 소프트웨어 개발과정에서 변경 발생은 피할 수 없는 사건
  - 변경을 대처하기 위한 공학적 방법의 적용이 요구됨.
  - 변경이 발생할 것으로 예상되는 부분을 모듈화 구조로 분리
- **일반화**
  - 다양한 플랫폼, 다양한 환경, 다양한 사용자를 지원하기 위한 원리
  - 하드웨어 성능이나 사양에 의존적이지 않는 소프트웨어 개발

한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

22

## 소프트웨어공학의 원리

### ■ 점진성

- 단계적이며 순차적으로 소프트웨어를 개발하고자 하는 공학적 원리
- 작은 단위의 소프트웨어를 반복 개발하면서 전체 시스템을 완성
- 단계적인 개발과 배포를 통한 사용자 피드백의 수집과 반영

### ■ 명세화

- 소프트웨어 개발 과정 및 대상물에 대한 정보를 체계적으로 기술하는 원리
- 팀 기반의 개발 활동을 지원하기 위한 정보의 공유 지원
- 지속적으로 진화하는 소프트웨어에 대한 이력서

## [프로젝트 I] 주제선정/문제기술서(SOP)/프로젝트헌장(PC)-(과제#2~3)

### 프로젝트 I

### 프로젝트 정의서 작성하기

소프트웨어 개발 프로젝트를 준비하기 위한 작업을 진행합니다. 준비 단계에서는 향후 학습을 통해 습득하는 공학적 지식과 기법을 적용할 소프트웨어 프로젝트를 정의하는 것입니다. 프로젝트 정의서에는 다음과 같은 내용을 포함합니다. 부록에 제시한 프로젝트 문서 양식 1번(시스템 정의서)을 활용하여 작성하기 바랍니다.

- (1) 프로젝트 명칭 정하기
- (2) 개발 시스템 명칭 정의하기
- (3) 개발 시스템에 대하여 간략히 설명하기(3~5줄 정도)
- (4) 개발 시스템에서 제공하는 핵심 기능 5개 이상 나열하기
- (5) 기존 유사 소프트웨어에 대하여 조사하고 분석하기

프로젝트: 주제선정/문제기술서(SOP)/프로젝트헌장(PC)-시스템정의서 예

누구나 설치 가능한 Open CCTV

시스템 정의서

팀명	김강진	김강진	김강진	김강진
팀원	팀장	2019039037	팀원	2019041057
		김희우		김후회
				김후회
시스템명	누구나 설치 가능한 open CCTV			
시스템 설명	누구나 설치 가능한 open CCTV			
운영환경	웹, 무선 통신, GPS 위치정보, 라즈베리 파이			
주요핵심기능	기능 1) 설치 가능			
	- 사용자는 자신이 원하는 위치에 CCTV를 설치한다.			
	- 라즈베리 파이와 카메라 모듈을 이용하여 간단하게 화면을 설치한다.			
	기능 2) 스트리밍 가능			
	- 화면을 스트리밍 할 수 있는 서버 프로그램을 제작한다.			
	- 지도의 해당 CCTV 선택 시 화면을 볼 수 있도록 제공한다.			
	- 스트리밍 영상은 모든 사람이 확인할 수 있다.			
	기능 3) 녹화 가능			
	- 설치된 CCTV의 녹화 내용을 DB에 저장한다.			
	- 녹화 내용은 설치한 사람이 확인할 수 있다.			
기타 고려사항	기능 4) GPS 기능			
	- 현재 설치된 CCTV를 구글, 네이버 지도에 표시한다.			
	- 설치자는 GPS모듈을 통해 위치를 측정하거나, 설치자가 직접 위치를 등록한다.			
	기능 5) 도난방지 기능			
	- 설치된 CCTV의 위치가 변경될 시 텔레이지에 알림이 온다.			
	- 설치된 CCTV의 위치가 변경될 시 설치자에게 이메일이 발송된다.			

한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

25

[이론 1장] 소프트웨어공학 개요 요약

■ 소프트웨어에 의한 사고의 발생 및 손실의 증가, 거대화

• MCO 위성/아리안 로켓, 보잉 737 MAX8 항공기, 자율주행 자동차 등 소프트웨어 결함으로 사고 발생

• 사고로 인한 경제적, 인명적 손실의 발생

• 체계적이고 안전한 소프트웨어 개발의 중요성이 높아짐

■ 소프트웨어의 위기 의식 확산 및 해결 노력

• 복잡도 증가, 규모의 대형화, 사용자 니즈의 급변에 대응하는 소프트웨어 개발 필요

• 인공지능 혹은 기계 학습 기반의 소프트웨어 개발의 범람

• 다양한 공학적 원리 및 기법의 적용을 통한 품질 좋은 소프트웨어 개발

■ 소프트웨어공학으로 발전

• 정의: 소프트웨어의 개발, 운용, 유지보수와 같은 수명주기 전반을 체계적·서술적·정량적으로 다루는 활동의 총체적인 모임

• 목표: 품질 좋은 소프트웨어의 개발

한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

26

Copyrighted by JinweonSuk(2025-2학기)

13



이론교재

1장

소프트웨어공학  
개요



1. 소프트웨어 고장 사례

2. 소프트웨어 위기

3. 소프트웨어공학 기술의 적용

4. 과제/진도/Q&A

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

27

4. 과제/진도: [과제#1] 이론1장+실습1장 연습문제 풀이

[과제#1] 이론1장+실습1장 연습문제 풀이(2점)


작성 방법: 이론1장/실습1장의 연습문제(전체)를 풀이하여 작성방법 적용하여 작성

제출 방법: 제출파일명 부여방법 적용하여 PDF 파일로 변환하여 LMS의 과제란에 제출기한 이내에 제출

내용/분량: 제한 없음

제출 파일: [과제#1]-이론1장+실습1장연습문제-"게임공학과"-**"학번"**-**"이름"**-20250918  
예시) [과제#1]-이론1장+실습1장연습문제-게임공학과-2023123456-한게임-20250918

제출 기한: 2025.9.18. (목요일), 24시 이전까지

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

28


4. 과제/진도: 2주차/전체			
주차	강의 내용	수업 유형	학습 활동
1	공통8장 강의안내+이론1장 SE개요+GSE 과목 사전 설문지 작성	대면수업(이론/실습)	대면수업/실습, 과제 해결
2	이론2장 SW의 품질+실습1장 UML 이해+실습12장(+@) starUML 모델링도구 사용법/실지/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
3	이론3장 SW 개발프로세스+실습2장 UML 구성요소/뷰+프로젝트 팀편성/주제선정/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
4	이론4장 SW 개발방법론(DevOps+UP)+실습3장 유스케이스 다이어그램+문제기술서(SOP) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
5	국경일(개천절) 휴강(15주차 보강)	국경일 휴강	국경일 휴강
6	이론5장 프로젝트 관리+실습4장 클래스 다이어그램+프로젝트정의서(PC) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
7	이론6장 SW 비용산정+실습5장 순차 다이어그램+프로젝트관리계획서(PMP) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
8	이론7장 요구사항 도출+실습6장 통신 다이어그램+요구사항정의서(SRD)/중간발표(PT+PMR) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
9	중간고사(필기+개인)+프로젝트 중간발표(PT+PMR+팀별)/피드백	대면수업(시험/발표)	서술형 필기시험/구두발표
10	이론8장 객체지향 분석+실습7장 활동 다이어그램+요구사항추적표(RTM) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
11	이론9장 모듈화 설계+실습 8장 상태 다이어그램+1. 요구사항명세서(SRS) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
12	이론10장 설계 패턴+이론11장 객체지향 설계+실습9장 컴포넌트 다이어그램+설계기술서(SDD) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
13	이론12장 인스펙션+이론13장 코딩+실습10장 배치 다이어그램+구현계획서(SIP) 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
14	이론14장 화이트박스 테스트+이론15장 블랙박스 테스트+실습 11장 패키지 다이어그램+시험 계획서(STP)/시험설계서(STD 작성/피드백	대면수업(이론/실습)	대면수업/실습, 과제 해결
15	이론16장 SW 개발 적용 기술+실습12장 것과 깃허브 활용 방법+구현결과서(SIR)/시험결과서(STR)/최종발표(PT+PCR) 작성/피드백	대면수업(이론/실습) (5주차 보강)	대면수업/실습, 과제 해결 (5주차 보강)
16	기말고사(L&L+개인중간고사(필기+개인)+프로젝트 중간발표(PT+PCR+팀별)/피드백 +최종보고서(PCR) 제출	대면수업(시험/발표)	서술형 필기시험/구두발표

한국공학대학교  
Korea University of Science and Technology

이론 1장 소프트웨어공학(SE) 개요

29

4. 과제/진도: 2주차(결과)-3주차(계획)					
2주차 결과/3주차 계획					
주차	주요학습내용	학습성과 학습목표	수업운영방법	학습준비사항	교재, 참고도서 (page)
2주차	<ul style="list-style-type: none"><li>이론 2장 SW의 품질</li><li>실습 1장 UML의 이해</li><li>실습 12장 StarUML을 이용한 SW 개발 방법</li></ul>	<ol style="list-style-type: none"><li>SW 품질의 이해</li><li>UML의 이해</li><li>StarUML 모델링 도구</li><li>사용법 이해</li></ol>	<ul style="list-style-type: none"><li>대면강의+실습</li><li>[과제#1] 이론(1장)+실습(1장) 연습문제 풀이</li></ul>	교재 준비(이론, 실습) 및 이론 2장/실습 1장, 12장 읽어 보기	강의계획서+이론/실습 교재/참고도서+강의자료
3주차	<ul style="list-style-type: none"><li>이론 2장 SW의 품질</li><li>실습 2장 UML 구성요소와 뷰</li><li>실습 12장 StarUML을 이용한 SW 개발방법2</li></ul>	<ol style="list-style-type: none"><li>SW 품질의 이해</li><li>UML 구성요소와 뷰의 이해</li><li>StarUML 모델링 도구의 사용법 이해2</li></ol>	<ul style="list-style-type: none"><li>대면강의+실습</li><li>[과제#2] 이론(2장)+실습(2장) 연습문제 풀이</li></ul>	교재 준비(이론, 실습) 및 이론 2장/실습 2+12장 읽어 보기	강의계획서+이론/실습 교재+참고도서+보충자료

한국공학대학교  
Korea University of Science and Technology

이론 1장 소프트웨어공학(SE) 개요

30

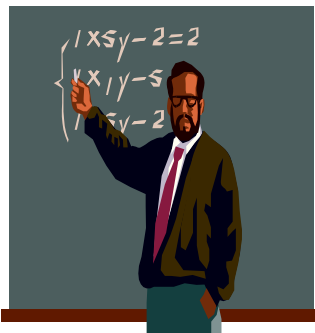
## 7. 과제/진도: 3주차 안내

- 강의계획서는 **잘(정확히) 숙지**하고, 매주 강의 **진도 확인**하기
- [과제#1] 이론1장+실습1장 연습문제 풀이(개인별, 2점)**
  - 제출 파일: **[과제#1]-이론1장+실습1장연습문제-"게임공학과"- "학번"- "이름"-20250918**  
예시) [과제#1]-이론1장+실습1장연습문제-게임공학과-2023123456-한게임-20250918
  - 제출 기한: **2025.9.18. (목요일), 24시 이전까지**
- 강의교재(이론/실습) 준비 및 교재 1~2장 읽어보기

👉 3주차: **이론 2장/실습 2, 12장+@ 강의 및 실습**

## □ 궁금하면 질문하자!


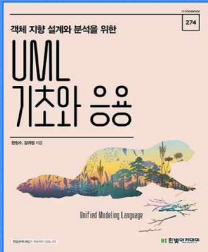

- LMS의 질의응답, 댓글, 쪽지 및 메일 기능을 적극 활용하기 바랍니다!**



학습(學習)은 배우고 익히라는 것이다.  
배우기만 힘쓰면 스스로 할 수 없는 사람이 된다!  
항상 배우고 익혀야 한다!

# Q & A






# [GSE 2주차-1] 수고했습니다!

## 다음 시간에 만납시다~~~

이 과목에서 사용되는 일부 자료, 영상물 등은 강의 내용을 보충하기 위해 교육 목적으로 활용하였습니다. 이 과목의 강의 자료 및 영상물의 불법적 이용, 무단 전재·배포는 법적으로 금지되어 있으니, 학생 여러분은 학습 외 용도의 사용을 주의하시기 바랍니다.

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

33