



2024-2학기

UNIFIED
MODELING
LANGUAGE™




객체 지향 설계와 분석을 위한

UML
기초와 응용



한빛아카데미

FIGURE 1-1



소프트웨어 공학
이론과 실제

한빛아카데미

게임소프트웨어공학(GSE)

[1주차-3] 이론 1장 소프트웨어 공학(SE) 개요

1. 소프트웨어 고장 사례

2. 소프트웨어 위기

3. 소프트웨어 공학 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

■ 주요 내용

1. 소프트웨어 고장 사례

2. 소프트웨어 위기


3. 소프트웨어공학 기술의 적용


■ 학습 목표

1. 최근 소프트웨어의 오류로 인하여 발생한 대표적인 사고(고장) 사례 이해

2. 소프트웨어 개발에는 어떠한 문제들이 존재하는지, 소프트웨어 개발이 왜 어려운지 이해

3. 소프트웨어공학이 무엇인지 이해



 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

2

UNIFIED
MODELING
LANGUAGE™

이론교재

1장

소프트웨어공학
개요

각에 지향 설계와 분석을 위한
UML
기초와 응용

소프트웨어 공학
이론과 실제

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

- 화성의 대기 탐사를 목적으로 1998년 개발
- 화성 궤도에 진입하는 과정에서 통신 두절

■ MCO 위성 결함 분석 결과

- 발견된 사실
 - 위성 속도 변화 계산 프로그램(SM_FORCE)에 사용된 ΔV 변수가 정상 값보다 4.45배 작게 계산된 것을 발견
- 원인 분석
 - MCO 추력 데이터의 단위: 국제 표준 단위인 1(N-s) 사용 명세
 - SM_FORCE 프로그램: 미국식인 4.45(lb-s)를 사용

■ 유사 사례: 프랑스의 아리안 로켓 발사 실패

- JOVIAL 언어를 사용한 프로그램에서 변수값 오버플로우 문제
- 16비트 자료형 정의를 8비트로 사용, 기존 코드의 copy & paste에서 기인
- 로켓 궤도값의 자료형 오버플로우로 (-)고도로 인식해서 로켓이 자동 폭발함

예상 궤적
실제 궤적
궤적을 낮추기 위해 작용한
화성 중력은 예상 궤적보다
더 크게 작용함

57Km 226Km
화성
To Earth

(b) MCO의 화성 궤도 진입

한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

4

Copyrighted by 석진원(2024-2학기)

2

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

한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

5

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


한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

6

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**

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

7



이론교재

1장

소프트웨어공학
개요



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


 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

8

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


9

소프트웨어 개발 프로젝트 통계

- 소프트웨어 개발 프로젝트의 성공과 실패율

Year	Succeeded (%)	Challenged (%)	Failed (%)
1994	16	53	31
2000	28	49	23
2004	29	53	18
2009	32	44	24
2013	41	40	19
2014	36	47	17
2015	36	45	19

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


 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

10

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

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

11

소프트웨어 고장 그래프

- 하드웨어 고장 그래프

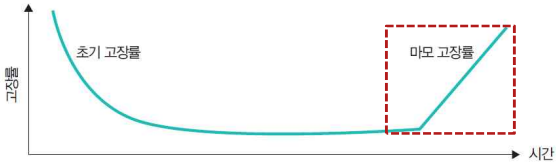


그림 1-11 하드웨어 수명주기에 따른 고장률 그래프

- 소프트웨어 고장 그래프

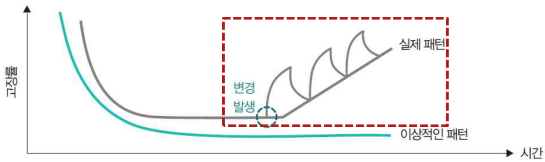



그림 1-12 소프트웨어 수명주기에 따른 고장률 그래프


 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

12

인공지능 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 도적적인 영역에서의 인공지능의 편향성 문제

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

13



이론교재

1장

소프트웨어공학
개요



- 소프트웨어 고장 사례
- 소프트웨어 위기
- 소프트웨어공학 기술의 적용
- 과제/진도

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

14

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


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)

 한국공학대학교


이론 1장 소프트웨어공학(SE) 개요

17

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 적용되는 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)

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

18

소프트웨어공학의 목표

```
graph TD; A([소프트웨어 개발을 위한  
다양한 공학적 기법 탄생]) --> B[개발 대상의 명확화]; A --> C[개발 과정의 체계화]; A --> D[개발 수명주기 지원]; B --> E[사용자의 요구사항을 충족시키는  
품질 좋은 소프트웨어 개발]; C --> E; D --> E;
```

한국공학대학교
이론 1장 소프트웨어공학(SE) 개요


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

소프트웨어공학의 원리

- 추상화
 - 세부사항은 감추고 대표적인 속성으로 대상을 정의하는 공학적 원리
 - 대상에 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

21

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

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

22

[이론 1장] 소프트웨어공학 개요 요약

■ 소프트웨어에 의한 사고의 발생 및 손실의 증가, 거대화

• MCO 위성, 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

23

[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

 한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

24

Copyrighted by 석진원(2024-2학기)

12

25

26

4. 과제/진도: [과제#1] GSE 과목 사전 설문지 작성

- [과제#1] GSE 과목의 사전 설문지를 개인별로 작성해서 제출(2점)
 - 작성/제출 방법: 반드시 업로드한 양식 사용해서 작성(HWP), 작성 후 pdf 파일로 변환해서 LMS(e-class)에서 제출기한 내에 업로드
 - 내용/분량: 과목 사전 설문지를 자세하게 작성 제출(1페이지 이상 2페이지 이내)
 - 제출 파일명: [과제#1]-게임공학과-"학번"-이름"-GSE사전설문지-20240911
 - 제출기한: 2024.9.11.(수요일), 24시 이전까지

[과제#1] (o o o)의 객체지향소프트웨어공학(OOSE) 사전 설문지 (2024년 8월 29일, 1주차)

학 번		이름	
학 과		학 계	
학과제 (비교/비교)		학과제 (비교/비교)	
자기 소개 (필수 사항)	1. 성격과 특징 2. 장우 및/또는 3. 학업 수준 및 현황 4. 자신에 대하여 명확하고, 구체적으로, 솔직하게 소개를 해주십시오. 자기 소개는 솔직하고 구체적으로...		
이 과목에 수강 목적	1. 이 과목에 대한 사전 이해 2. 수강 목적 3. 알고 싶은 것 4. 강의 평가(이론, 실습, 프로젝트의 방법) 5. 기타 의견 이 과목에 대한 이해와 수강 목적을 잘 정리/구명해서 알려주세요. 수강 목적 및 요구사항은 구체적으로...		

전공/수업/영역	1. "여러분의 전공" 주제를 설명하는 "개념"을 2. 프로그래밍 가능한 언어 종류 3. 프로그래밍에서 데이터(변수/구조) 작성방법 학습/경험 4. UI/프로그래밍 설계내용 문서화 수준/경험 5. 구현된 프로그래밍, 시험/실습/실습 경험 등 6. 프로그래밍, 개발 및 프로젝트 등에서 소스 코드의 버전 관리 및 개발시스템의 활용 /활용 관리 학습과 수행한 경험 및 도구 7. 프로젝트 관리(방법/도구) 및 개발/개발의 프로세스, 방법, 생명주기 등에 대한 수준/학습 및 적용 경험 8. 선진화된 학습/프로그래밍, 프로젝트 등에서 소스 코드와 설계, 학습방법, 활용수준, 프로젝트 경험(주제/목적/방법/결과/성공/실패 등) 9. 여러분과 주제에 대해서도 학습/경험을 정리해서 알려주세요. 학습이나 경험이 잘 안되었거나 어려웠던 점을 구체적으로 기술해주세요
프로그래밍 (선택) 능력	- Python 언어: 앞 1, 중 1, 뒤 1 - C 언어: 앞 1, 중 1, 뒤 1 - C++ 언어: 앞 1, 중 1, 뒤 1 - Java 언어: 앞 1, 중 1, 뒤 1 - HTML/Javascript/PHP/ASP/SP 언어 등: 앞 1, 중 1, 뒤 1 - Database 언어(관계형): 앞 1, 중 1, 뒤 1 - 기타 언어(언어명): 앞 1, 중 1, 뒤 1 프로그래밍 학습과 관련된 기타사항: - 현재까지 학습한 프로그래밍 언어와 수준을 간단해서 표시/기술해주세요. - 위의 내용 이외에 배웠던 것이 더 있다면 기술해주세요.
기타사항 (추가사항, 기타사항 등)	



4. 과제/진도: 1주차/전체

주차	강의 내용	수업 유형	학습 활동
1	공통0장 강의안내+이론1장 개요+GSE 과목 사전 설문지 작성	대면수업(이론/실습)	대면수업/실습, 과제 해결
2	이론2장 SW의 품질+실습1장 UML 이해+실습12장(+@) starUML 모델링도구 사용법/설치/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 기타 기타 활용 방법+구현결과서(SIR)/시험결과서(STP)/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





4. 과제/진도: 1주차(결과)-2주차(계획)					
1주차 강의 진행 결과					
주차	주요학습내용	학습성과 학습목표	수업운영방법	학습준비사항	교재, 참고도서 (page)
1주차	<ul style="list-style-type: none">0장 과목 오리엔테이션: 과목/교수 및 강의계획 설명이론 1장: 소프트웨어공학 개요	<ol style="list-style-type: none">게임소프트웨어 공학에 대한과목 개요 및 강의계획 이해게임개발과 소프트웨어공학의 이해	<ul style="list-style-type: none">대면강의+실습[과제#1] 자기소개서 작성 제출	교재 준비(이론, 실습) 및 교재의 목차와 이론 1장 읽어 보기	강의계획서+이론/실습 교재/참고도서+강의자료
2주차	<ul style="list-style-type: none">이론 2장 SW의 품질실습 1장 UML의 이해실습 12장 StarUML을 이용한 SW 개발방법	<ol style="list-style-type: none">SW 품질의 이해UML의 이해StarUML 모델링 도구사용법 이해	<ul style="list-style-type: none">대면강의+실습	교재 준비(이론, 실습) 및 이론 2장/실습 1장, 12장 읽어 보기	강의계획서+이론/실습 교재/참고도서+강의자료

한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

29

7. 과제/진도: 2주차 안내	
<ul style="list-style-type: none">강의계획서는 잘(정확히) 숙지하고, 매주 강의 진도 확인하기[과제#1] GSE 과목의 사전 설문지 작성 제출(개인별, 2점)<ul style="list-style-type: none">앞에 제시된 과목 사전 설문지 작성양식(A4용지, 세로)은 반드시 작성작성된 과목 사전 설문지는 9월 11일(수요일) 24시까지 LMS(e-class) 시스템에서 제출(업로드) (작성/제출 방법 준수!)제출 파일명: [과제#1]-게임공학과-"학번"- "이름"-GSE사전설문지-20240911강의교재(이론/실습) 준비 및 교재 1~2장 읽어보기 <p> 2주차: 이론 2장/실습 1, 12장+@ 강의 및 실습</p>	


한국공학대학교

이론 1장 소프트웨어공학(SE) 개요

30


□ 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

 한국공학대학교

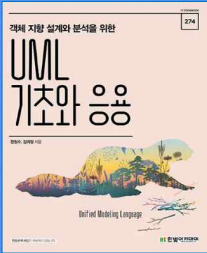
이론 1장 소프트웨어공학(SE) 개요

31

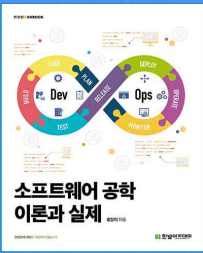
UNIFIED MODELING LANGUAGE™



각종 지향 설계와 분석을 위한 UML 기초와 응용




소프트웨어 공학 이론과 실제



[GSE 1주차-3] 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

32

Copyrighted by 석진원(2024-2학기)

16