

Software Engineering

Course Introduction

소프트웨어 공학(Software Engineering)

과목-강좌#: 109130-31001

- Pre-requisite: OOP에 대한 충분한 이해와 능숙한 프로그래밍 능력.
- 소프트웨어 개발에 관련된 내용을 배우게 되므로, 그 기반이 되는 프로그래밍은 잘 할 수 있다고 가정하고 강의가 진행됩니다.
- 수업시간: 목 1~4교시 (09:00AM~12:50PM)
- 수업형태
 - 강의는 온라인으로 진행되며 대부분의 경우 온라인 동영상 강의를 시청하게 됩니다.
 - 일부 강의와 시험의 경우 실시간 Zoom강의가 진행되거나 대면 시험이 치러질 수 있습니다.
 - 실습은 팀 프로젝트 형태로 진행되고 필요한 경우 강의 자료가 제공됩니다.

강의 정보

강의 자료 및 연락방법

- Textbook
 - Ian Sommerville, Software Engineering, 10th Edition, Pearson
 - Ian Sommerville 지음, 권기태 외 옮김, 소프트웨어 공학, 제10판, 한티 미디어
- 강의자료는 기본적으로 교재 내용을 바탕으로 일부 내용이 변경/추가되어 제공됩니다.
- 번역본을 구매하여 공부해도 상관없으나, 기술 용어 등은 모두 원서를 기준으로 사용합니다.
- 그 외 실습 자료로 다양한 온라인 문서 및 튜토리얼 등이 사용될 수 있습니다.

강의 정보

강의 자료 및 연락방법

- Communications
 - 김진대, 미래관 331호, 방문 전 미리 연락주세요.
 - 강의관련 질문 및 기타 문의는 모두 e-class를 통해 받습니다 - 이메일 사용 x
 - 평일 오후 9시 이후, 주말에는 가급적 질문을 하지 말아주세요.
 - 이 경우 답변은 다음 날 또는 월요일에 드릴 생각입니다.
 - 저녁이 있는 삶, 주말이 있는 삶을 적극 지지합니다.

평가 방식

배점 비율

- 중간고사: 30%
 - 문제 풀이 형식의 시험
 - 일부 단답형 + 서술형 문제 위주
 - 중간고사 이전까지 배운 내용.
- 기말고사: 30%
 - 중간고사와 같은 방식
 - 전 범위 - 중간고사 이후 배운 내용을 중점적으로 다룸.

평가 방식

배점 비율

- 팀 프로젝트: 40%
 - 프로젝트 결과물 20%
 - 클라우드 서비스를 이용, 온라인 쇼핑몰 개발
 - 공통 요구사항 구현 + 팀별 추가 구현
 - 프로젝트 진행과정 20%
 - Scrum을 이용한 Agile Development로 개발 진행.
 - 강의 시간에 배우는 다양한 이론 내용을 개발과정에서 적절히 적용하였는지 평가함.
 - 팀 내에서도 개인별 프로젝트 기여도에 따라 점수가 달라질 수 있음.

등급별 최저 점수제

적용 기준

- 배점 비율에 따라 최종 점수를 100점으로 환산
- 환산된 점수를 기준으로 A~F의 등급을 부여
 - 주의: 출석 점수는 따로 없지만, e-class 기록상 3분의 1이상 출석이 미달하는 경우 F가 부여됩니다.
- 각각의 등급을 받기 위해 요구되는 최저 점수가 있습니다.
 - A+: 80 / A0: 60
 - B+: 50 / B0: 40
 - C+: 30 / C0: 20

등급별 최저 점수제

최저 점수제 적용 예시

- 상대평가로 A0는 수강생 30%까지, B+는 수강생 50% 부여가능한 경우
- 최종 점수가 65점이라면,
 - 65점의 점수가 수강생의 30%내에 속하면 A0
 - 30%초과, 50%내에 속하면 B+
- 최종 점수가 55점이라면,
 - 수강생의 10%이내라도 B+
 - 50% 초과 70% 이내라면 B0

등급별 최저 점수제

적용 이유

- **열심히 한 수강생이 충분한 보상을 받을 수 있도록 하기 위함.**
 - 코로나로 인해 확대된 상대평가 정책이 적용되면 수강생의 50%가량에게 A가 부여.
 - 시간을 들여 과목에서 학습해야 하는 내용을 익힐 동기가 충분하지 않음.
- **강의 계획서에 공식적으로 등급평가 기준을 제시하여 부여된 성적의 가치를 높이기 위함.**
 - 유래없는 코로나 사태로 학점 인플레가 커진 상황에서 성적 자체가 근거자료로 가치를 잃을 가능성이 있음.
 - 최저한의 학습 내용을 익혔다는 것을 보장하여 얻은 성적에 최소한의 가치를 보장.

등급별 최저 점수제

적용 이유

- 지그 지글러(Zig Ziglar)의 벼룩 훈련 이야기
 - 뚜껑이 있는 유리병에 벼룩을 가둬 놓았다가 나중에 유리병을 없애면,
 - 벼룩이 실제 더 뛸 수 있음에도 유리병보다 높이는 뛰어오르지 않음.
 - 실제 과학적인 실험을 통해 증명된 사실은 아니지만, 자주 인용되는 이야기입니다.
- 등급별 최저 점수제는 수강생 점수 평균이 높은 경우 아무런 영향이 없음.
- 여러분이 충분히 B0 등급 최저 점수는 받을 수 있다고 생각합니다.

학습 내용

소프트웨어 공학 이론

- 소프트웨어 공학은 높은 품질의 소프트웨어를 효율적으로 개발하는 방법을 연구하는 학문
 - 어떤 것이 높은 품질의 소프트웨어인가?
 - 그런 소프트웨어를 어떻게 개발할 것인가?
- 이론 내용 자체가 어느 정도 실무적인 고려사항을 반영할 수 밖에 없음.
- 소프트웨어 개발 전반을 다루므로 그 범위가 매우 넓습니다.
- 전반적으로 세세하고 구체적인 내용보다는 상당히 상위의 개념들을 다루게 됨.

학습 내용

실습 및 팀 프로젝트

- 보다 실무적인 내용은 실습을 대체하는 팀 프로젝트와 관련하여 배우게 됩니다.
- 수없이 많고 다양한 개발 방법론, 프로그래밍 언어, 개발 도구 및 기법들이 존재함.
- 이를 세세하게 하나씩 배우는 것은 중요하지 않습니다.
 - 소프트웨어 분야는 매우 빠르게 발전하는 산업 분야
 - 현재 대세로 생각되는 것들도 5년 뒤에는 비주류가 될 수 있음

학습 내용

실습 및 팀 프로젝트

- 몇 가지를 선택하여 실습을 진행하나, 그 기반이 되는 이론적인 내용을 이해하는 것이 더 중요합니다.
 - 이를 기반으로 새로운 기술들을 보다 빠르고 쉽게 익힐 수 있는 토대를 닦아야 함.
- 새로운 소프트웨어를 개발할 때는 또 새로운 방법을 적용해야할 수 있음.
 - 자신의 지식을 바탕으로 적절한 개발 방법 및 도구를 선택할 수 있는 능력을 갖추는 것이 필요함.
- 단순한 개발자(Software Developer)가 아닌 소프트웨어 엔지니어(Software Engineer)를 목표로 합시다.

팀 프로젝트

온라인 쇼핑몰 개발

- 클라우드 서비스를 이용하여 온라인 쇼핑몰을 개발합니다.
- 구체적으로는 Amazon Web Services (AWS)에서 EC2 (Elastic Compute Cloud)를 이용합니다.
 - 간단한 설정을 통해 Linux 서버를 이용할 수 있습니다.
- 공통 요구사항으로 여러분의 온라인 쇼핑몰에서 반드시 구현되어야 하는 기능들이 제시됩니다.
 - e.g.) 메인 화면, 상품 리스트 화면, 결제화면, 쇼핑몰 관리자 화면 등
- 그 외 팀별로 자신들만의 아이디어를 더해 쇼핑몰을 구현합니다.
 - e.g.) 온라인 서점을 구현하려는 경우 → 서점에 특화된 기능에 따라 요구사항 추가

팀 프로젝트

개발 방식

- 팀원들의 핵심 역량에 따라 적절한 아키텍처 선택
- LAMP Stack
 - Linux + Apache + MySQL + PHP를 합친 형태
 - 가장 기초적이고 기본적인 형태이며 쉽게 설정 가능
- 그 외 Node.js, jQuery, React, Bootstrap 등 원하는 라이브러리/프레임워크를 사용하여도 무방함 → 사용하는 것을 적극 권장.
- 실습 자료로는 기본적인 온라인 쇼핑몰의 구조를 설명하고, LAMP Stack을 EC2에서 구축하는 방법을 예시로 설명합니다.
- 웹 서버나 데이터 베이스로 다른 것을 이용하고 싶은 경우에는 문서를 보고 직접 설치하시면 됩니다.

팀 프로젝트

프로젝트 관리

- 본격적인 개발은 3주차에 배우게 될 Scrum을 통해 프로젝트를 관리하며 진행하게 됩니다.
- 이 과정에서 Version Control, Issue Management 등이 적절하게 이루어지는지 개발 과정 자체도 평가 대상이 됩니다.
- 이를 위해 필요한 Git, GitHub 및 Trello 등의 사용법은 실습 자료로 제공되고, 그 외 다른 도구를 이용하고 싶은 경우에도 얼마든지 사용할 수 있습니다.
- 단, 기본적으로 개발과정에서 생성되는 산출물의 경우에는 평가를 위해 제시할 수 있어야 합니다.
 - e.g.) 개발 과정에서 발생한 Commits, 이슈 처리 과정 등

팀 프로젝트

평가 방식

- 팀 프로젝트 평가는 두 팀이 짝을 이루어 서로의 개발 진행 상황과 결과를 평가하는 방식.
- 중간고사 후 중간 보고, 기말고사 후 최종 발표를 통해 팀 프로젝트 진행 상황과 최종 결과물을 제출합니다.
- 짝을 이룬 팀은 서로의 보고서와 결과물을 보고 적절히 개발되었는지 평가합니다.
 - 요구사항이 적절히 구현되었는지, 개발은 적절한 방법을 사용해 이루어졌는지
- 개발자만이 아닌 프로젝트 관리자 및 검수자 등 개발에 참여하는 다양한 역할을 경험하기 위함.
- 평가가 적절하지 않다 생각하면 반박이 가능하며 이 경우 평가를 한 팀이 검수자로서의 점수에서 감점을 당할 수 있음.

팀 프로젝트

팀 구성 방식

- 각 팀은 **최소 3명 이상의 팀원**으로 이루어져야 함.
- 프로젝트 진행과정에서 번갈아가며 프로젝트 매니저 역할을 수행해야 하므로 팀 결성시 이 부분을 고려하세요.
- 원하는 팀원으로 팀을 구성할 수 있지만, 팀 결성 상황에 따라 임의로 팀원이 변경될 수 있습니다.
- e-class 팀 생성 기능을 이용하여 등록 → 2주차까지 완료.
- 상호 평가를 위한 팀 매칭은 제가 임의로 결정합니다.
- 팀별 최대 인원은 5명 이하를 권장하지만 특수한 경우 그 이상이 될 수 있습니다.

Q & A