Study Notes - COMP 4350 Software Engineering II

Course Overview

Project Details

Overview

- Project Composition: Team of 4-5
- Objectives:
  - Hand-on experience with DevOps tools (e.g., GitHub Action, Docker)
  - Quality assurance skills (e.g., load testing, security analysis)
  - Implement a complete web application

Requirements

- At least as many core functional features as team size
- Non-functional requirement: handle 20 users with 200 requests per minute concurrently
- Open choice of programming language, framework, and tech stack
- TA acts as customer

Timeline

- Proposal + Presentation: 5%
- Progress Tracking: 20% (Across 4 sprints, each sprint 5%)
- Technique Sharing Seminar: 5%
- Final Project Presentation: 10%
- Final Project Deliverable: 20%
- Peer Evaluation: Adjusts individual grades based on team contribution

Milestones

- **Proposal**: Includes project summary, core features, frameworks, and user stories. Presentation required.
- **Tracking**: Sprints span 2-4 weeks, with concrete tasks to be accomplished in each.
- **Evaluation**: TAs provide feedback and test the software at each sprint's end.
- **Seminar**: Share techniques and frameworks with the class.
- **Final Presentation**: Features live demo and lessons learned.
- **Final Deliverable**: Includes documentation (user stories, design, manuals, reports) and source code (product code, test cases, CI/CD scripts, Docker image).

GitHub Use

- Central for project management
- Discussions, meeting minutes, and code reviews documented
- Code review before committing
- Track issues using GitHub

Project Management

- Version control with GitHub
- Automate processes with DevOps pipeline
- Team formation within the first week

Course Content

Software Engineering in Data Mining & ML

- Application of AI for software engineering tasks (API recommendation, code completion, debugging)
- Data analytics on software artifacts (Stack Overflow, GitHub, AppStore)
- Software security considerations

Core Topics

- DevOps (Configuration management, CI/CD, AI)
- Software Architecture (Microservices)
- Software Quality Assurance (Advanced Testing, Mutation, Load, Performance Testing)
- Security Analysis
- Refactoring & Research in SE
- Data-Driven SE

Books & Resources

- Recommended reading uploaded on UMLearn

Final Test Details

- **Open book exam**: 40% of final grade
- Will cover all course material
- Date and location announced by Student Records Office

Policies & General Information

- Late submissions: Acceptable up to 2 days with a penalty (10% per late day)
- Re-grading: 7 days available for re-grading requests
- Generative AI: Allowed as a learning aid; require disclosures along with the prompts used

Software Engineering Principles

Software Crisis & Engineering

- The "Software Crisis": Over-budget, over-time, inefficient, low quality, and not meeting requirements
- Software engineering brings systematic, rigorous, and measurable approaches to software development

Software Engineering Activities

- **Specification**: Defining what the software should do

- **Development**: Producing the system
- **Evolution**: Responding to changing requirements or environments
- **Validation**: Ensuring the system meets needs and requirements

Software Process Phases

- Different processes share similar elements but differ in organization
- Examples: Waterfall vs. Agile (Scrum)

Software Engineering Fundamentals

- Managed development process for all software types
- Key aspects: Reliability and Performance

Software Quality Assurance

- Frequent mobile app updates pose risks of new bugs; quality assurance aims to mitigate such risks

Slide: 1

Title: Comp 4350 - Software Engineering II - Lecture 1 Dr. Shaowei Wang

Slide: 2

Body: Research into data mining and software engineering

- Machine learning for software engineering (e.g., API recommendation, code completion)
- Data analytics on software artifacts (e.g., Stack Overflow, GitHub, AppStore)
- Auto debugging (e.g., bug localization, fault localization)
- Software security Contact: shaowei.wang@umanitoba.ca Website: Dr. Shaowei Wang - UM Website

Slide: 3

Title: About me (Shaowei Wang) About the course:

- Course website on UMLearn
- Check regularly for announcements and updates
- Lectures: 60 mins lecture + 15mins Q&A
- All slides will be posted on UMLearn after class

Slide: 4

Contact & Office Hours: Dr. Shaowei Wang

- Tuesday 9am-11am, EITC-E2-408 TAs:
- Xu Yang yangx4@myumanitoba.ca Office hour: TBD
- Shayan Daneshvar daneshvs@myumanitoba.ca Office hour: TBD
- Cody Wallbridge wallbric@myumanitoba.ca Office hour: TBD

Slide: 5

What are we doing here?

Slide: 6

Title: What are we doing here? Cover topics:

- DevOps
- Configuration management
- Continuous integration
- Continuous deployment
- AI for DevOps
- Architecture (microservice)
- Software quality assurance
- Advanced testing (Mutation, Load, Performance)
- Security analysis
- Refactoring
- Research in software engineering
- Machine learning for software engineering
- Data-driven software engineering

Slide: 7

Title: Recommended Books (Uploaded on UMLearn)

Slide: 8

How will the course be evaluated? Title: Overall evaluation scheme Final exam: 40% Project: 60%

Slide: 9

Final exam: 40% Project: 60%

Slide: 10

What is the project about?

Slide: 11

Title: Project topic A team of 4 or 5 students in the course to build a software system:

- Web application with front and back ends
- Core functional features equivalent to team size
- Non-functional requirement: Handle 20 users with 200 requests per minute
- TA acts as the customer
- Any programming language, framework, and tech stack

Slide: 12

Goal of project:

- Obtain hands-on experience/skills of DevOps and software quality assurance

- Create a significant project

Slide: 13

How will the project be tracked and evaluated? Team project (60%):

- Project proposal + presentation (5%)
- Progress tracking (20%, 4 sprints, each 5%)
- Technique sharing seminar (5%)
- Final project presentation (10%)
- Final project deliverable (20%)
- Peer evaluation

Slide: 14

Each team will submit a project proposal, including:

- Project summary
- Core features
- Technique or framework
- User stories for core features
- Present proposed project in class (10 mins)

Slide: 15

Title: Progress tracking

- 4 sprints (2-4 weeks each)
- Each sprint needs concrete tasks finished
- Detailed instructions for TAs to evaluate

Slide: 16-23

Content placeholders

Slide: 24

Title: First thing to do!

- Build the team
- Team leader confirmed in the first week
- Team registration Google Sheet

Slide: 25

Final test (40%):

- Open book
- Time and location announced by Student Records Office

Slide: 26

Title: What do I need to succeed?

- Take the test seriously
- Active class participation
- Software development and delivery process knowledge
- Teamwork and support

Slide: 27

Policy:

- Late submission penalty
- Re-grading window
- Guidelines on using Generative AI

Slide: 28

Introduction to Software Engineering

Slide: 29

Title: Software is everywhere!

Slide: 30-33

Content placeholders

Slide: 34-39

Content related to software complexity and challenges

Slide: 40

Title: What is this?

Slide: 41

Title: “Software Crisis” Challenges faced in software projects leading to the emergence of Software Engineering principles

Slide: 42-43

Explanation of Software Engineering scope and activities

Slide: 44-45

Software process phases and an overview of Scrum

Slide: 46

Title: Software engineering fundamentals (I)

- Principles applicable to all software systems

Slide: 47

Rapid update of Mobile apps Link

Slide: 48

Title: How to reduce the risk of introducing new bugs when making changes to the software?

Slide: 49

Title: Software quality assurance

Slide: 50

Title: Actions Actions to ensure software quality and reliability

Slide: 51

Title: Software Quality Assurance A set of activities designed to ensure that the software under consideration meets all requirements and specifications

Slide: 52

Policy:

- Late submission penalty
- Re-grading window
- Guidelines on using Generative AI