CS304 软件工程课程大纲

2023 年春季学期起 **2**

2022 年春季学期及之前 **10**

课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	**软件工程 Software Engineering**
2.	**授课院系** **Originating Department**	计算机科学与工程系 Department of Computer Science and Engineering
3.	**课程编号** **Course Code**	CS304
4.	**课程学分 Credit Value**	3
5.	**课程类别** **Course Type**	专业核心课 Major Core Courses
6.	**授课学期** **Semester**	春季 Spring
7.	**授课语言** **Teaching Language**	英文 English
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师）** **Instructor(s), Affiliation& Contact** **（For team teaching, please list all instructors）**	陶伊达，讲师，计算机科学与工程系，taoyd@sustech.edu.cn Yida Tao, Lecturer, Department of Computer Science and Engineering, taoyd@sustech.edu.cn

9.	实验员/助教、所属学系、联系方式 Tutor/TA(s), Contact	王大兴，教学实验师，计算机科学与工程系，wangdx3@mail.sustech.edu.cn Daxing Wang, Teaching Technician, Department of Computer Science and Engineering 田蕾，教学实验师，计算机科学与工程系，tianl3@mail.sustech.edu.cn Lei Tian, Teaching Technician, Department of Computer Science and Engineering
10.	选课人数限额(可不填) Maximum Enrolment（Optional）	

11.	授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify)	总学时 Total
	学时数 Credit Hours	32		32		64

12.	先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	CS109 计算机程序设计基础 Introduction to Computer Programming CS203 数据结构与算法分析 Data Structures and Algorithm Analysis
13.	后续课程、其它学习规划 Courses for which this course is a pre-requisite	CS409 软件测试 Software Testing
14.	其它要求修读本课程的学系 Cross-listing Dept.	

教学大纲及教学日历 SYLLABUS

15. 教学目标 Course Objectives

This course introduces important concepts, methodology, processes, best practices, and tools in Software Engineering. Topics covered include DevOps, software process, requirement analysis, software design, software measurements and metrics, software evolution and maintenance, software testing, CI/CD, deployment pipelines and strategies.

This course also emphasizes on software project management, team collaboration, and process automation, empowered by version control systems, automated tools and platforms that are covered in lab tutorials and practices. Students will also apply the learned knowledge in a practical course project that last for the entire semester.

16. 预达学习成果 **Learning Outcomes**

On completion of this course, the students should be able to:

1. Understand the difference between programming and software engineering
2. Understand the different phases and tasks in a software development process
3. Understand the concepts, methodology, tools, best practices, and challenges in each DevOps phase
4. Create a practical software system in a team, with a disciplined pipeline, and applying proper automation and quality control

17. **课程内容及教学日历 （如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）**

Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

Syllabus of Theory Couse

Chapter 1 Introduction to Software Engineering

- Course logistics
- What is software engineering?
- Time, scale, and tradeoffs

Chapter 2 Software Process

- Software process models
- Agile development
- DevOps

Chapter 3 Version Control Systems

- Types of version control systems
- Git internals & best practices

Chapter 4 Software Requirements

- Stakeholders
- Types of software requirements
- Requirements modeling

Chapter 5 Software Design

- Software design models
- Design concepts

Chapter 6 Software Build

- Build systems
- Dependency management

Chapter 7 Software Quality

- Aspects of software quality
- Code clones
- Code review

Chapter 8 Measurement & Metrics

- Measuring software quality
- Software metrics
- Measurements validity

Chapter 9 Software Evolution & Maintenance

- Software evolution
- Legacy systems & Deprecation
- Software reengineering
- Refactoring

Chapter 10 Software Documentation

- Types of software documentation
- Software documentation best practices
- Case study: g3doc

Chapter 11 Software Testing I

- Types of software testing
- Testing concepts

Chapter 12 Software Testing II

- Blackbox & whitebox testing
- Test doubles

Chapter 13 CI/CD

- CI/CD
- Software deployment strategy
- Deployment pipeline
- Cloud-native applications

Chapter 14 Artificial Intelligence and Software Engineering

- AI overview
- How AI is used in the DevOps process

Chapter 15 Advanced Topics in Software Engineering research

- Individual academic talk

Chapter 16 Project Demonstration

- Project Demonstration
- Course review

Syllabus of Lab Course

Lab 1. Introduction to Course Project

- Introduction to course project
- How to team up
- How to setup GitHub Classroom for project collaboration

Lab 2. Project Management & Demo Project Setup

- Introduction to GitHub project management features
- Installing and executing the Teedy project

Lab 3. Git

- Git installation and command line usage
- Basic git and GitHub workflow
- Practice on using git to update Teedy

Lab 4. UML

- Types of UML diagrams and examples
- Tools for creating UML diagrams

- Practice on requirement analysis

Lab 5. Milestone 1

- Milestone 1 presentation

Lab 6. Maven

- Maven installation
- Using Maven in command line and IDEA
- Using Maven in Teedy
- Practice on using maven for building a simple web application

Lab 7. Code Review

- Introduction to code review features in GitHub
- Practice on code review with Teedy

Lab 8. Code analysis

- Using JArchitect to compute code metrics for Teedy
- Using PMD to analyze coding problems of Teedy

Lab 9. Milestone 2

- Milestone 2 presentation

Lab 10. Software Documentation

- Using JavaDoc in command line and IDEA to create documentation
- Using Swagger to create API documentation
- Practice on generating documentation for a web application

Lab 11. Software Testing I

- Unit testing with JUnit
- Generate test report
- Generate code coverage report with Jacoco

Lab 12. Software Testing II

- Automated UI testing with Selenium

Lab 13. Containers

- Introduction to Docker
- Docker installation & usage
- Practice on using Docker

Lab 14. CI/CD

- Using Jenkins for CI/CD
- Using GitHub Actions for CI/CD
- Practice on using Jenkins

Lab 15. Project Demonstration I

- Project demonstration and presentation
- Project evaluation

Lab 16. Project Demonstration II

- Project demonstration and presentation
- Project evaluation

18. 教材及其它参考资料 Textbook and Supplementary Readings

Software Engineering A Practitioner’s Approach by Roger Pressman, 8th edition

Software Engineering by Ian Sommerville. 10th edition

Software Engineering at Google by Titus Winters, et al.

现代软件工程基础 by 彭鑫 et al.

课程评估 ASSESSMENT

19.

评估形式 Type of Assessment	评估时间 Time	占考试总成绩百分比 % of final score	违纪处罚 Penalty	备注 Notes
出勤 Attendance				
课堂表现 Class Performance		10%		
小测验 Quiz				
课程项目 Projects		40%		
平时作业 Assignments		20%		
期中考试 Mid-Term Test				
期末考试 Final Exam		30%		
期末报告 Final Presentation				
其它（可根据需要改写以上评估方式） Others (The above may be modified as necessary)				

20. 记分方式 GRADING SYSTEM

☑ A. 十三级等级制 Letter Grading

□ B. 二级记分制（通过/不通过） Pass/Fail Grading

课程审批 REVIEW AND APPROVAL

21. 本课程设置已经过以下责任人/委员会审议通过

This Course has been approved by the following person or committee of authority

课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	**软件工程 Software Engineering**
2.	**授课院系** **Originating Department**	计算机科学与工程系 Department of Computer Science and Engineering
3.	**课程编号** **Course Code**	CS304
4.	**课程学分 Credit Value**	3
5.	**课程类别** **Course Type**	专业核心课 Major Core Courses
6.	**授课学期** **Semester**	春季 Spring
7.	**授课语言** **Teaching Language**	英文 English
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师）** **Instructor(s), Affiliation& Contact （For team teaching, please list all instructors）**	陈馨慧，助理教授，计算机科学与工程系，tansh3@sustech.edu.cn 陈馨慧，Assistant Professor, Department of Computer Science and Engineering, tansh3@sustech.edu.cn

9.	实验员/助教、所属学系、联系方式 Tutor/TA(s), Contact	胡春风，教学实验师，计算机科学与工程系，hucf@sustech.edu.cn Chunfeng Hu, Teaching Technician, Department of Computer Science and Engineering, hucf@sustech.edu.cn
10.	选课人数限额(可不填) Maximum Enrolment（Optional）	

11.	授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
	学时数 Credit Hours	32		32		64

12.	先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	CS102A 计算机程序设计基础 A Introduction to Computer Programming A CS203 数据结构与算法分析 Data Structures and Algorithm Analysis
13.	后续课程、其它学习规划 Courses for which this course is a pre-requisite	CS409 软件测试 Software Testing
14.	其它要求修读本课程的学系 Cross-listing Dept.	无

教学大纲及教学日历 SYLLABUS

15. **教学目标 Course Objectives**

This course introduces some basic concepts in Software Engineering. Topics covered include unit testing, software configuration management, software process and reverse engineering. Some advanced topics like test generation and static analysis will also be covered. This course emphasizes on software project management, team collaboration and usage of version control system. Students will apply the knowledge they learned in various lab individual assignments. Students will also apply their knowledge on project management and collaboration through version control system in a course project that last for 8 weeks.

16. **预达学习成果 Learning Outcomes**

Upon completion of this course, the students are expected to have a good understanding of writing unit tests, using software configuration management system, applying software process and reverse engineering in real-world applications.

17. **课程内容及教学日历 （如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）**

Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

Week 1: Introduction

Introduction to course

Introduction to Software Engineering

[Lab] Introduction to different roles in software projects, including project manager, software engineering and software tester.

Week 2: Software Configuration Management

Introduction to concepts in software configuration management

Example of coftware configuration management systems (CVS, SVN, Git)

Explain the differences between different systems and how to use these systems

[Lab] Learn about the commands in Git and the basics of using GitHub for managing a software project.

Week 3: Extreme Programming

History of XP

Introduce some general practices of XP (Planning Game, pair programming and user stories)

Provide examples and benefits of planning poker and pair programming.

[Lab] Learn about how to use Github classroom for submitting programming assignments and the syntax of markdown in GitHub,

Week 4: Concepts in Extreme Programming and Introduction to Testing

Explain about the role of testing in XP

Introduce the motivations behind testing

Provide examples and benefits of testing.

[Lab] Learn about how to write JUnit test and practice the concept of “pair testing”

Week 5: Test-driven Development

Introduce the steps in Test-driven development

Provide examples and benefits of test-driven development

Introduce the concept of code coverage and explain how to compute different coverage metrics

[Lab] Learn about how to use learn about how to use automated test generation tool for generating tests and how to measure test coverage using Jacoco plugin

Week 6: Mutation testing and Test generation

Introduce the concepts of mutation testing

Introduce the concepts of test generation

Introduce the test generation algorithm of Evosuite.

[Lab] Learn about the basics of tensorflow API and build a simple mobile app that uses Tensorflow-Lite.

Week 7: Software metrics

Introduce the concepts of software metrics

Introduce technical and non-technical software metrics

Explain coupling and cohesion

[Lab] Learn about the basics of tensorflow API and build a simple mobile app that uses Tensorflow-Lite.

Week 8: Reverse Engineering

Introduce the concepts of mutation testing.

Introduce some re-engineering patterns.

Explain some reverse engineering activities.

[Lab] Learn about the basic of static analysis tools and integrate these tools to the course project, .

Week 9: Static Analysis

Introduce the concepts of static analysis.

Introduce coding standard and common bug patterns.

Explain the benefits of static analysis compared to dynamic analysis.

[Lab] Learn about how to reverse engineer an Android app using apktool and Java Decompiler.

Week 10: Component and Reuse

Introduce the concepts of component and framework.

Explain the benefits of software reuse

Explain the techniques in improve software reuse

[Lab] Learn about some popular frameworks and examples of software reuse. Group discussion on course project.

Week 11: UI design

Introduce the concepts of good UI designs.

Explain good UI design through several examples.

Explain how the UI design influence the design of code and tests.

[Lab] Compare some UI designs and choose a good UI design for the course project

Week 12: Continuous Integration & Regression Testing

Introduce the concepts of continuous integration.

Introduce the concepts of regression testing.

Explain the tool supports and techniques for continuous integration and regression testing.

[Lab] Learn about how to use configure software projects for continuous integration server and write good regression test cases.

Week 13: Documentation

Introduce the concepts of good software documentation.

Explain formal and informal documentation.

Introduce research on software documentation

[Lab] Learn about how to write good Javadoc comments and prepare user manual and documentation for course project.

Week 14: Debugging and Repair

Introduce several debugging techniques.

Introduce program slicing

Introduce automated program repair techniques

[Lab] Learn about how to use a debugger and automated program repair technique.

Week 15: Review for Final exams

Review all concepts thought in the classes

Provide some class exercises as a preparation for final exam

Provide guidelines on course project report and presentation.

[Lab] Group discussion on course project and prepare report for the course project.

Week 16: Course Project Presentation

Students will give a presentation on course project.

Discuss the pros and cons of each project.

Provide feedback for improving the writing and future presentation skills.

[Lab] Prepare the source code and user manuals for the final release. Prepare the final report and future works for the course project.

18. 教材及其它参考资料 Textbook and Supplementary Readings

Code Complete: A Practical Handbook of Software Construction (Second Edition) by Steve McConnell.

Software Engineering (10th Edition) by Ian Sommerville.

Object-Oriented Reengineering Patterns by Oscar Nierstrasz, Stéphane Ducasse, Serge Demeyer

Various articles in journals and conference proceedings given during the lectures.

课程评估 ASSESSMENT

19.	评估形式	评估时间	占考试总成绩百分比	违纪处罚	备注

Type of Assessment	Time	% of final score	Penalty	Notes
出勤 Attendance		10%		In-Class Exercises/Attendance
课堂表现 Class Performance		20%		Weekly lab tutorial
小测验 Quiz				
课程项目 Projects		35%		
平时作业 Assignments				
期中考试 Mid-Term Test				
期末考试 Final Exam		35%		
期末报告 Final Presentation				
其它（可根据需要改写以上评估方式）Others (The above may be modified as necessary)				

20. 记分方式 **GRADING SYSTEM**

☑ A. 十三级等级制 **Letter Grading**

☐ B. 二级记分制（通过/不通过） **Pass/Fail Grading**

课程审批 REVIEW AND APPROVAL

21. 本课程设置已经过以下责任人/委员会审议通过

This Course has been approved by the following person or committee of authority