SE300, Software Engineering Practices

Course Description: This course introduces students to the fundamental principles and methodologies of large-scale software development. Students learn about theory and practice of software engineering and work as part of a team on a full life-cycle software that includes planning, software specification, software design, coding, inspections, and testing.

Goals: The primary course objective is for students to acquire knowledge and experience about the processes, methodologies, tools and techniques used in developing large software systems. This course is designed to prepare students to work on a larger and more complex project.


Weekly Syllabus [Please Consult Blackboard as well]

Week-1 [1/8]:

Read: SEPA (Software Engineering – A Practitioner’s Approach), Chapter 1


Activity-1: Transformation Architecture – Speed-up? Sequential and Concurrent Transformation, Issue with clock scaling pushing computer architecture to threading and vector processing

Start Assignment #1 (Software Process Fundamentals and Architecture Patterns)

Week-2 [1/13, 15]:

Read: SEPA Chapters 2, 3 & 4

Lecture & Discussion: Software Process Concepts

Activity Follow-up: Transformation Speed-up, Observations, Analysis, Misconceptions, Q&A

Lecture: Software Process Models (Waterfall vs. Evolutionary), Architecture Patterns (Transformation, Client/Server, Interactive, Graphics-Visualization, Services, Transaction)

1/15: Pop-Quiz Given on Reading, Solutions Provided

Week-3 [1/20, 22]:
Read: SEPA Chapter 5, 6, 7 & 8

Lecture: Agile Development and Teams, Group Discussion on Process and Architecture Patterns / Anti-Patterns

Lecture: Principles of Requirements Analysis

Pop-Quiz Returned with Feedback

Activity-2: Client/Server Architecture – Who’s the Client and who’s the Server? - File RAID Archive

Start Assignment #2 (Requirements Practice – Requirements for a Cloud Storage-as-a-Service Application)

Week-4 [1/27, 29]:

Read: SEPA Chapter 8 & 9

Lecture: Requirements Modeling with Scenarios

Activity Follow-Up: Client/Server Architecture – Observations, Analysis, Misconceptions, Q&A


1/29: QUIZ #1 - Part-1 on Foundations and Architectures (Chapters 1-8), Discuss Quiz Solutions

Week-5 [2/3, 5]:

Read: SEPA Chapter 10 & 12

Lecture: Class-Based Methods and Introduction to Modelio, OOA/OOD and Comparison to SA/SD

Lecture: Software Architecture Level Design Patterns
1. Transformation – E.g. Data Processing
2. Client-Server – E.g. Cloud Storage
3. Interactive – GUI, Augmented Reality, Command-Line
5. Services – E.g. Navigation System
6. Transaction – Web, DBMS, File systems
7. Other? Hybrid?

Activity-3: Interactive Architecture
When, How and Why Does an Application become Interactive?
Does Update Rate Matter? (Canny Example with Interactive Threshold)

Minute Paper-3: Choose an Application (Secure Messaging, Cloud Storage, Computer Vision Threat Detector, On-line Music Store, OCR, System Status Monitor, System Loading Analysis), Does it Fit an Common Architecture
Pattern? Write Top 5 Questions you Have on the Domain as an exercise to Discover Requirements

Start Assignment #3 (More Analysis and Design Practice with Modelio, use of CMVC)

---

**Week-6 [2/10, 12]:**

Read: SEPA Chapter 12 & 13

Lecture: Practical Prototype, PoC Software Configuration Management and Version Control Methods

Activity Follow-Up: Observations on Interactive Applications

Lecture: Software Design Principals, Transition from Architecture Level to Design

**Week-7 [2/17, 19]:**

Read: Review all SEPA Chapters 1 to 13, excluding Chapter 11

Lecture: REVIEW, Chapters 1 to 7, Process and Patterns for Process and Architecture

Lecture: REVIEW, Chapters 8-10, 12-13, UML, Requirements, Analysis and Architecture, Transition to Design

EXAM-1 Practice Quiz

---

**Week 8 - MID-TERM EXAM (Feb 24 & 26) – Chapters 1-10 & 12,13**

EXAM-1, Day-1, Knowledge & Concepts: SEPA 1-10, 12, 13 and Notes from Class

EXAM-1, Day-2, Analysis & Architecture: SEPA 1-10, 12, 13 and Notes from Class

---

**Week-9 [3/3, 5]:**

Return Mid-term and Review Solutions in Class

Read: SEPA Chapter 14 & 15

Activity-4: Crash of Flight 447 – PBS Video on Contributing Factors (Sensor failure, Pilot error, Side-stick controller and Avionics User Interfaces)

Minute Paper-4: Beyond Command Line, GUIs, to what?
Lecture: User Interface Design Principals

Discussion: Team Formation for Application Design & Development

Start Assignment #4 (Team Formation for Application Analysis and Design)

SPRING BREAK - 3/9 to 3/13 - NO CLASS

Week-10 [3/17, 19]:

Read: SEPA Chapter 16

Lecture: Component (Module) Level Design, SA/SD and Legacy Systems, Refactoring, Goals for Re-use and Maintenance

Minute Paper-5: Service Architectures: Periodic, Peer-to-Peer, Different than Client/Server?

Lecture: Pattern-Based Design and Advantages of OOA/OOD with OOP

Activity-5: Develop 10 minute Presentation for Next Week

Week-11 [3/24, 26]:

Read: SEPA Chapters 19, 20 & 21

Lecture: SQA - Software Configuration and Testing Fundamentals (Introduction to GitHub)

Minute Paper-6: Biggest Threats to Team Success?

**** Thursday Team Activity Follow-up: Scrum Sprint Presentations - Requirements & Architecture Analysis Walk-throughs

Start Assignment #5 (Team Specification and Analysis, Design, Implementation & Test Plans)

Week-12 [3/31, 4/2]:

Read: SEPA Chapter 22, 23, 24 & 29

Lecture: Software Configuration Management and Testing Strategies

Minute Paper-7: Are You Ready to Start Assignment #6? Would a Prototype or PoC Help?

Lecture: Testing Conventional SA/SD vs. OO Applications

Activity: Transaction Architecture: List Types? Does it Require a Database?
Week-13 [4/7, 9]:

Read: SEPA Chapter 28, 33 & 34

Lecture: Methods of Validation and Verification, Project Estimation, Planning and Re-planning

Minute Paper-8: Can you Test a Design? Does Testing Have to Wait for Coding?

**** Thursday Team Activity Follow-up: Scrum Sprint Presentations – Design Walk-throughs

Start Assignment #6 (Team Development Lifecycle for Prototype Application)

Week-14 [4/14, 16]:

QUIZ #2 - 4/14 to Warm-up for Exam #2
REVIEW for EXAM #2

EXAM-2, Day-1, Knowledge & Concepts: Chapters 16, 19, 20, 21, 22, 23, 24, 29, 28, 33, 34

Week-15 [4/21, 23]:

EXAM-2, Day-2, Design & Construction: Chapters 16, 19, 20, 21, 22, 23, 24, 29, 28, 33, 34

**** Thursday Team Activity Follow-up: Scrum Sprint Presentations – PoC and/or Mock-Up Walk-throughs

Week 16 – FINAL EXAM, Thursday April 30th, 8:00am, King 132

Final Software System Design Presentation by Team (30 minutes Each, 20 minutes + 10 minutes Q&A)

Learning Outcomes:

1. Describe the major problems in large software system development. (d, h, j)
   o Week-#1, Assignment #1

2. Explain the major phases common to all standard software engineering life cycle modes. (e, k)
3. Exhibit knowledge of and explain each phase; the major product(s) of that phase, activities involved in the production of each product and quality factors for that product. (e, k)
   - Week-#2, Assignment #2

4. Describe issues, principles, methods and technology associated with software engineering theory and practices (e.g., planning, requirements analysis, design, coding, testing, quality assurance and configuration management.) (c, e)
   - Week-#4, Assignment #3

5. Explain the difference among the various major software engineering life cycle models and the advantages and disadvantages of each. (e)
   - [Spiral and Agile Evolutionary Process, SA/SD (Structured Analysis / Structured Design), OOA/OOD (Object Oriented Analysis and Design), PRC Assignments and general C/C++ programming]
   - Week-#5, #6, Assignment #3

6. Explain the difference between blackbox and white box requirements. (e)
   - Week-#12, Assignment #6

7. Explain the difference between explicit and implicit requirements/design methodologies. (e)
   - Week-#3, #4, Assignment #4

8. Explain the roles of coupling and cohesion in the evaluation of software designs. (e)
   - Week-#8, Assignment #5

9. Explain the difference between transformationally intensive and transactionally intensive systems. (e)
   - Week-#6, Assignment #6

10. Explain the difference between black box and white box testing. (e, k)
    - Week-#11, Assignment #5

11. Explain the relationship between testing and integration. (c, f)
    - Week-#10, Assignment #5

**ABET Outcome Statements** (Expectations of student knowledge and skills attained by graduation)

- a. An ability to apply knowledge of mathematics, science, and engineering
- b. An ability to design and conduct experiments, as well as analyze and interpret data
- c. An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- d. An ability to function on multi-disciplinary teams
- e. An ability to identify, formulate, and solve engineering problems
- f. An understanding of professional and ethical responsibility
- g. An ability to communicate effectively
- h. An understanding of the impact of engineering solutions in a global and societal context
- i. A recognition of, and an ability to engage in, life-long learning
j. An understanding of contemporary issues in electrical engineering
k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

**ABET Objectives Statements** (Expectations of our alumni working within industry)

The objectives of the College of Engineering are to produce engineers who:
- Demonstrate achievements in their chosen profession
- Contribute to the development of the profession
- Engage in professional growth
- Contribute to the welfare of society through service

Examples of activities which demonstrate each of these objectives include:

- **Demonstrate achievements in their chosen profession**
  - Project Leadership
  - Technical Leadership
  - Received professional recognition
  - Strong early-career performance

- **Contribute to the development of the profession**
  - Educational improvement within profession
  - Product development or improvement
  - Presentation, publications, patents

- **Engage in professional growth**
  - Membership in technical societies
  - Obtaining additional education
  - Achieved career advancement

- **Contribute to the welfare of society through service**
  - Participation in activities outside the professional area