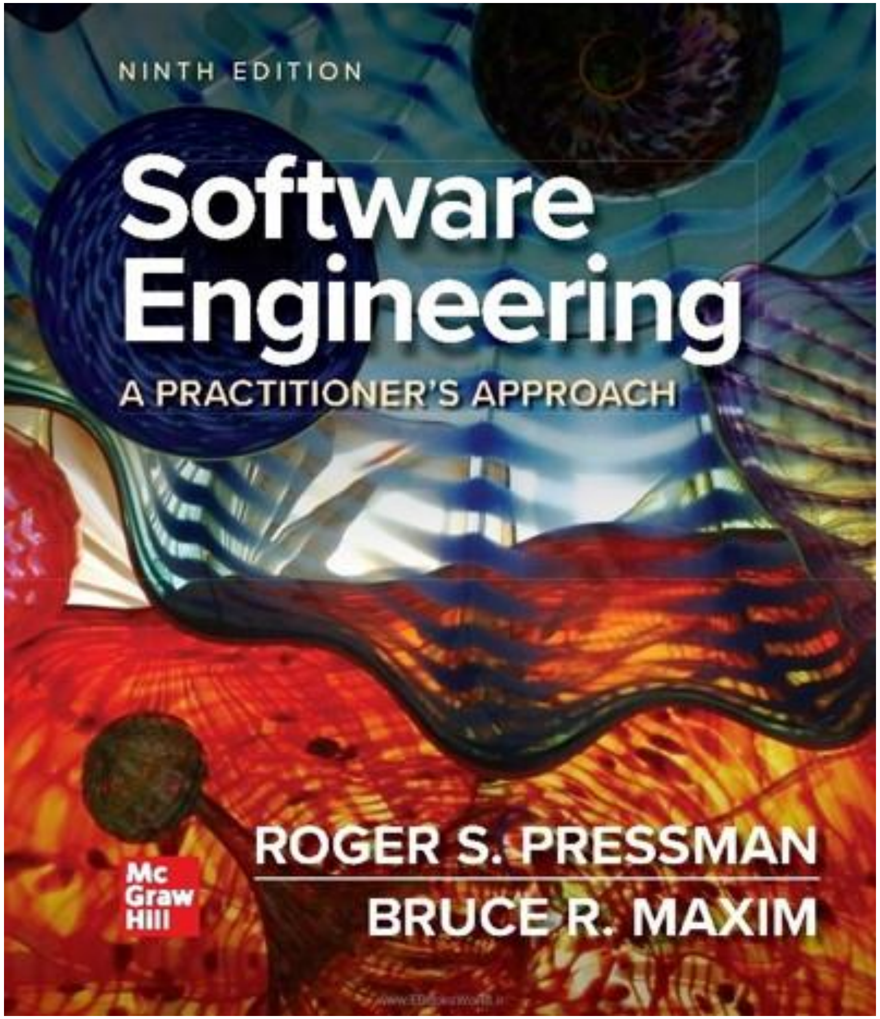
Technical and Vocational University
Minab Branch

Lecturer:

Mohammad Ahmadzadeh

Fall of 2023



SOFTWARE ENGINEERING 9

Chapter 1: Introduction (PPTX),

Chapter 2: Software Processes (PPTX),

- ❑ **RUP, UP, USDP, and FDD Methodologies**

Chapter 3: Agile Software Development (PPTX),

- ❑ **Scrum, XP, Kanban, and Crystal Methodologies, Refactoring, Test**

Chapter 4: Requirements Engineering (PPTX)

- ❑ **Analysis Pattern**
- ❑ **Project(1)**

Chapter 5: System modeling (PPTX)

- ❑ **Unified Modeling Language(UML) and Systems Modeling Language (SysML)**
- ❑ **Object-Oriented Design Pattern**
- ❑ **Business Process Model and Notation (BPMN)**
- ❑ **4+1 View Model**
- ❑ **Petri Net**
- ❑ **Formal Languages, Z Language**
- ❑ **General Responsibility Assignment Software Patterns(GRASP)**

Chapter 6: Architectural design (PPTX)

- ❑ **Architecture overview of GOV.UK applications**
- ❑ **Pattern-Oriented Software Architecture(POSA)**
- ❑ **Architecture description language (ADL)**
- ❑ **Microservice vs Serverless**

Chapter 7: Design and Implementation (PPTX)

- ❑ **JavaScript and Golang**
- ❑ **Object Oriented Programming, Functional**
- ❑ **Clean Code**
- ❑ **Anti-Pattern**
- ❑ **Design Pattern(GoF) and Solid Principles**
- ❑ **Refactoring**

Chapter 8: Software testing (PPTX)

- ❑ **Jestjs, Mochajs and Golang Testify**
- ❑ **Test-driven development(TDD)**
- ❑ **Behavior-driven development(BDD)**

Chapter 9: Software Evolution (PPTX)

Chapter 10: Socio-technical Systems (PPTX)

- ❑ **Project (2): Meta, Google, Banking**

Chapter 11: Dependability and Security (PPTX)

- ❑ **The Open Worldwide Application Security Project(OWASP)**
- ❑ **Software Cracking, Decompile and …**
- ❑ **CEH**

Chapter 12: Dependability and Security Specification (PPTX)

Chapter 13: Dependability Engineering (PPTX)

Chapter 14: Security Engineering (PPTX)

Chapter 15: Dependability and Security Assurance (PPTX)

Chapter 16: Software Reuse (PPTX)

- ❑ **Service-Oriented Architecture**
- ❑ **ERP Architecture**
- ❑ **MS Office Product Line**

Chapter 17: Component-based Software Engineering (PPTX)

- ❑ **CASE Tools**
- ❑ **MDE**

Chapter 18: Distributed Software Engineering (PPTX)

- ❑ **Golang Micro**

Chapter 19: Service-oriented Architecture (PPTX)

- ❑ **Microservice by Golang and Nodejs**
- ❑ **Golang GoKit and RPCX**

Chapter 20: Embedded Systems (PPTX)

- ❑ **Embedded Linux and Robotic OS**

Chapter 21: Aspect-oriented software engineering (PPTX)

Chapter 22: Project management (PPTX)

- ❑ **PMBOK**
- ❑ **Ms Project**
- ❑ **Jira**

Chapter 23: Project planning (PPTX)

Chapter 24: Quality management (PPTX)

Chapter 25: Configuration management (PPTX)

- ❑ **DevOps**

Chapter 26: Process improvement (PPTX)

Chapter 27:Reengineering and Patterns

Chapter 28: Blockchain

Appendix:

- ❑ **CDN and Cache Server**
- ❑ **Load Balancer**

Students Report

N	Date	Issue
1		Analysis of Scrum with XP, Kanban, and Crystal
2		Analysis of RUP with UP, USDP, and FDD
3		Review of Analysis Patterns
4		Petri Net with small case study
5		Analysis of TDD and BDD
6		Study of a Language Library, Framework and Tools
7		Study of DevOps Tools and Techniques
8		Study of a Public Blockchain Architecture
9		Review of Reengineering and Patterns

Students Presentation

N	Date	Issue	Names
1	02/07/9	Version Control System(VCS), Git	
2		Container Concepts, Docker	
3		Kubernetes, Docker Swarm	
4		Cloud Space	
5		Authentication and Authorization	
6		UI-UX	
7		Framework Concept, Reactjs	
8		Microsoft C# and Java Ecosystems	
9		Blockchain Architecture	
10		GoKit Microservice	
11		RabbitMQ Message Broker	
12		ActiveMQ Broker	
		DevOps	

Scores of 100

N	Title	Score	Delivery Date
1	Project 1	15	02/08/10
2	Project 2	10	02/08/30
3	Project 3	15	02/09/10
4	Present	15	02/09/30
5	Mid-Term(Part 1 and 2)	15	02/08/29
6	Final-Exam(Part 3,4 and 5)	25	02/10/18
7	Report	10	Two Weeks
8	Additional Scores for Codding	----	
9	Additional Scores for Research	10	02/09/20

Hints

- ❑ **Project 1:**

  Software Requirement Specification(SRS), Requirement Analysis and UML

- ❑ **Project 2 :**

  Enterprise Application Architecture such as Meta, Google and Banking Software Systems and any system you know

- ❑ **Project 3 :**

  Application or Service with whatever language you specialize in, Implement a real program in any language you like. This program cannot include phone book, agency, hospital, accounting and repetitive and cliché topics. Also, the project includes codes and documentation.

- ❑ **Present, Report , Research:**

  Explain a topic coherently with concept, structure, advantages, disadvantages and application. Mention some real examples of it and analyze them.

Refs:

- https://sourcemaking.com
- https://en.wikipedia.org/wiki/Pattern-Oriented_Software_Architecture
- https://www.dre.vanderbilt.edu/~schmidt/POSA/
- https://github.com/ppizarro/coursera/tree/master/POSA
- a SET OF 9 Patterns Introduced as a learning aid by Craig Larman
- Applying UML and Pattern: An introduction to object-oriented analysis and design and iterative development
- https://ecomputernotes.com/software-engineering/softwarerequirementsspecification
- https://en.wikipedia.org/wiki/Software_analysis_pattern
- https://martinfowler.com/books/ap.html
- Golang Design Patterns:

https://github.com/tmrts/go-patterns

https://golangbyexample.com/all-design-patterns-golang/

- Golang :

https://virgool.io/golangpub/go-developer-roadmap-part-3-yme5wi3jpsfe

Refs:

https://www.umsl.edu/~sauterv/analysis/F08papers/View.html

https://www.educba.com/case-tools/

https://www.javatpoint.com/software-engineering-function-oriented-design

https://www.javatpoint.com/software-testing-tutorial