College of Computing and Information Technology

Introduction to Software Engineering

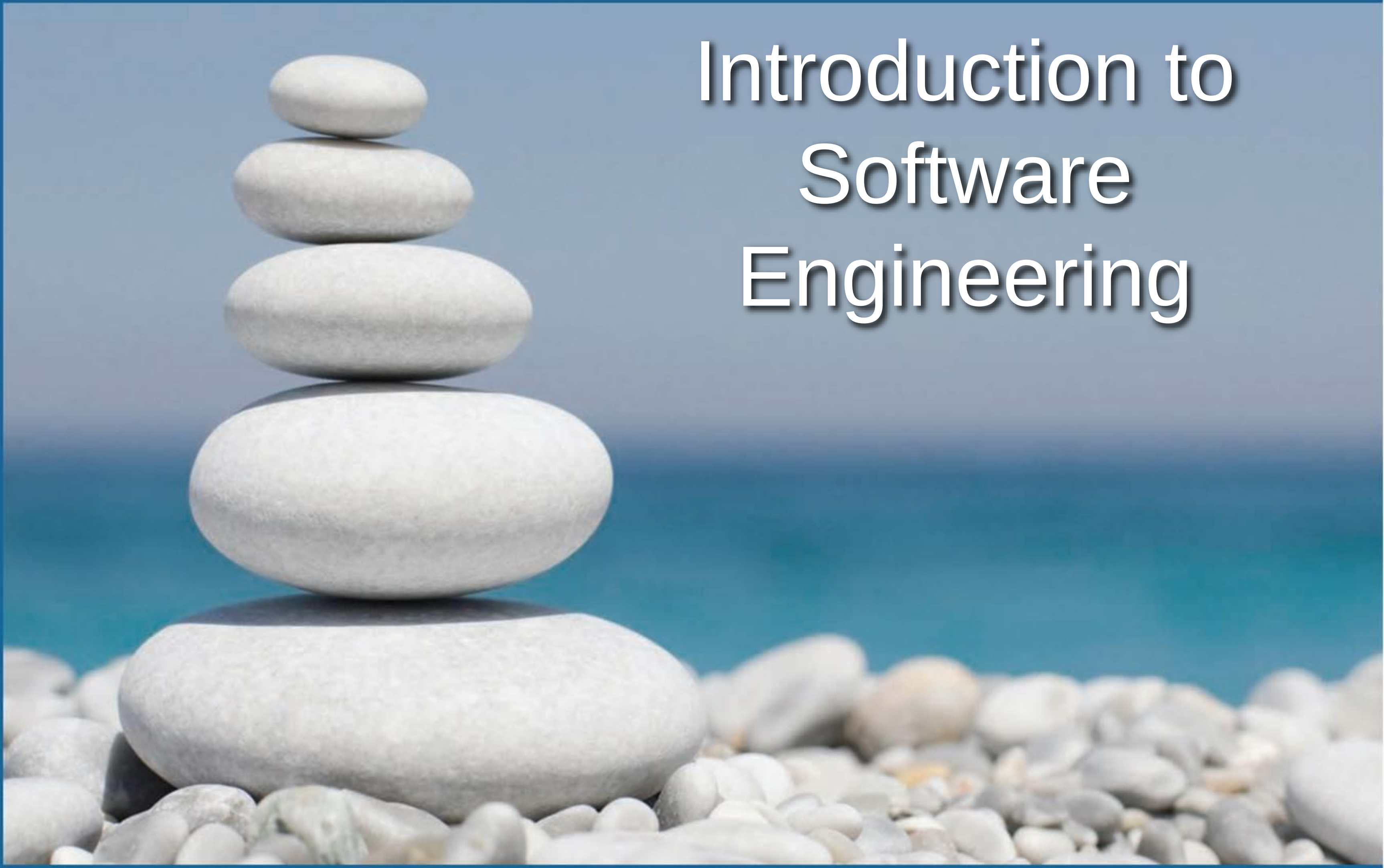
Lecture # 01

Assoc.Prof/ Mohammed Safy

Spring '24

CHAPTER 1

Introduction to Software Engineering



Course Objectives

Understanding the software engineering process and different process models and how to choose between them.

How to elicit requirements from a client and specify them.

Design in the large, including principled choice of a software architecture.

Develop a system design using UML notation.

Understanding good coding practices, including documentation, contracts, regression tests and daily builds.

Various quality assurance techniques, including unit testing, functional testing, and automated analysis tools.

Understanding of the role of project management including planning, scheduling, risk management, etc.

Contribution to Program Student Outcomes

(SO1) - Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

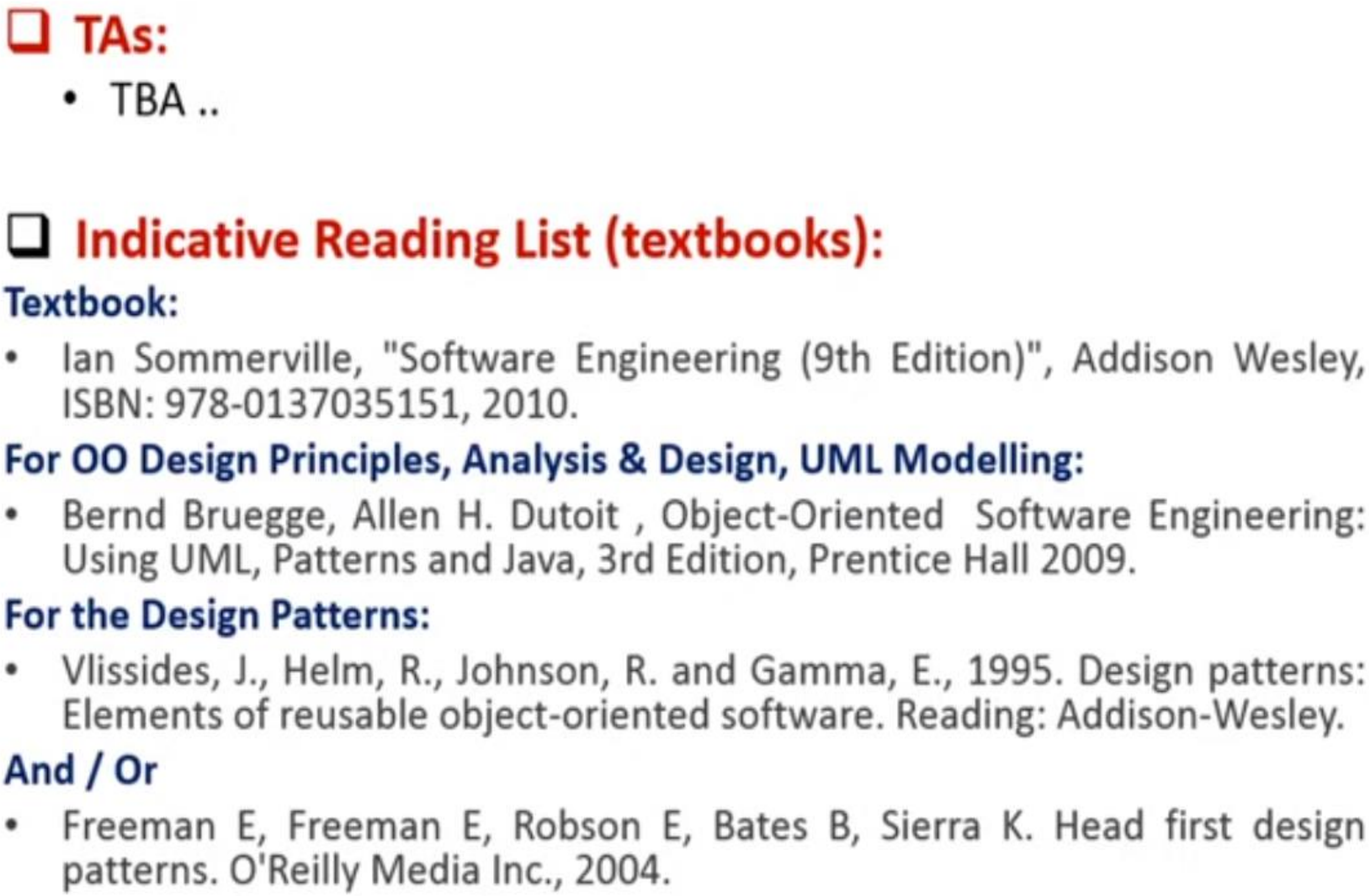
(SO2) - Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

(SO3) - Communicate effectively in a variety of professional contexts.

(SO4) - Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

(SO5) - Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline

Course Outline



❑ TAs:

- TBA ..

❑ Indicative Reading List (textbooks):

Textbook:

- Ian Sommerville, "Software Engineering (9th Edition)", Addison Wesley, ISBN: 978-0137035151, 2010.

For OO Design Principles, Analysis & Design, UML Modelling:

- Bernd Bruegge, Allen H. Dutoit , Object-Oriented Software Engineering: Using UML, Patterns and Java, 3rd Edition, Prentice Hall 2009.

For the Design Patterns:

- Vlissides, J., Helm, R., Johnson, R. and Gamma, E., 1995. Design patterns: Elements of reusable object-oriented software. Reading: Addison-Wesley.

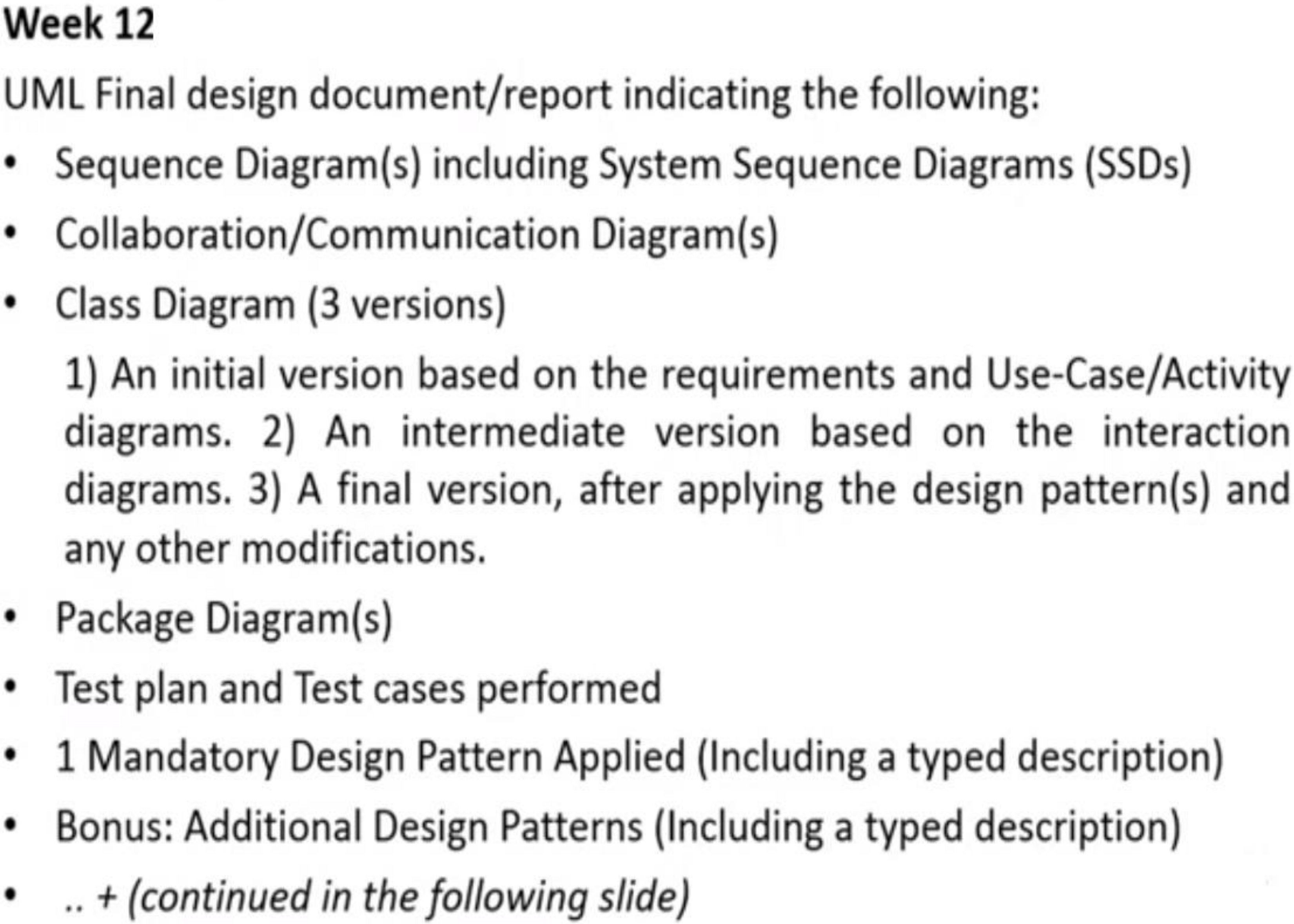
And / Or

- Freeman E, Freeman E, Robson E, Bates B, Sierra K. Head first design patterns. O'Reilly Media Inc., 2004.

Grading Policy

Assessment	Method
7th Week (30%)	Exam (30)
12th Week (20%)	Group Project: Presentation (10) + Document (10)
Coursework (10%)	Section Quiz (10)
Final (40%)	Examination (40)

Group Project



Week 12

UML Final design document/report indicating the following:

- Sequence Diagram(s) including System Sequence Diagrams (SSDs)
- Collaboration/Communication Diagram(s)
- Class Diagram (3 versions)

  1) An initial version based on the requirements and Use-Case/Activity diagrams. 2) An intermediate version based on the interaction diagrams. 3) A final version, after applying the design pattern(s) and any other modifications.
- Package Diagram(s)
- Test plan and Test cases performed
- 1 Mandatory Design Pattern Applied (Including a typed description)
- Bonus: Additional Design Patterns (Including a typed description)
- .. + *(continued in the following slide)*

Course Outline

Week 1: Introduction to Software Engineering Course
Week 2: Software Process
Week 3: Software Process Models
Week 4: Software Requirements Gathering
Week 5: Software Requirements Analysis and Specification
Week 6: Structured Analysis
Week 7: 7th Week Examination
Week 8: Object Oriented Analysis and Design
Week 9: Design Concepts and Principles
Week 10: Architectural Design
Week 11: Implementation Coding Practices and Principles
Week 12: PRESENTATIONS
Week 13: **Software Quality Assurance (**Verification and Validation**)**
Week 14: Project Management
Week 15: Revision
Week 16: Final Examination

SECTION 1.1

Software Engineering

What is Software Engineering?

Software engineering is the branch of computer science that deals with the **design, development, testing, and maintenance of software applications**. Software engineers apply engineering principles and knowledge of programming languages to build software solutions for end users.

Software engineers design and develop computer games, business applications, operating systems, network control systems, and middleware—to name just a few of the many career paths available.

What Careers Are There in Software Engineering?

Career opportunities in software engineering are driven by new technologies in automobiles, aviation, data management, telecommunications, factory control, robotics, defense, and security. Software engineers may develop computer games, business applications, operating systems, network control systems, and more. A bachelor's degree or higher is often required to work as a software engineer.

Two common jobs within software engineering are applications developers and systems developers.

What Careers Are There in Software Engineering?

Applications developers design computer applications, such as games, for consumers. They may create custom software for a specific customer or commercial software to be sold to the general public. Some applications developers create databases or programs for use internally or online.

Systems developers create operating systems, either for the public or for an organization. These operating systems keep computers functioning and control most of the consumer electronics in use today, including those in cell phones and cars. Often, systems developers also build the interface that allows users to interact with the computer.

What Careers Are There in Software Engineering?

Some common job titles for software engineers include:

- Applications Architect
- Applications Developer
- Computer Applications Engineer
- Database Administrator
- Mobile Developer
- Program Manager
- Programmer
- Project Leader
- Quality Assurance Analyst
- Software Engineer
- Software Quality Engineer
- Systems Analyst
- Systems Developer

What Tasks do Software Engineers do?

Day-to-day tasks for a software engineer might include:

- Designing and maintaining software systems or applications
- Evaluating and testing new software programs
- Optimizing software for speed and scalability
- Writing and testing code
- Ensuring platform compatibility
- Creating and enforcing IT standards within an infrastructure
- Maintaining documentation of IT systems
- Releasing software updates
- Collaborating with development teams, senior systems architects, and data science professionals
- Consulting with clients, engineers, security specialists, and other stakeholders
- Recommending software upgrades for existing programs and systems
- Presenting new features to stakeholders and internal customers

How Much do Software Engineers Make?

Thanks to strong industry demand and their own technical expertise, skilled software engineers are well compensated for the value they deliver. The mean annual salary for software developers was **$132,930** in May 2022.

The hiring outlook for software engineers is great, as well. The Bureau of Labor Statistics projects that **software engineering jobs will increase 25 percent through 2031**.

Source: U.S. Bureau of Labor Statistics

2023 Computing Salary Statistics:

https://www.mtu.edu/computing/salaries/

What Skills do Software Engineers Need?

Generally speaking, most software engineers will need to have the following technical knowledge and skills:

- Coding languages like Python, Java, C, C++, or Scala
- Object-oriented programming
- Database architecture
- Agile and Scrum project management
- Operating systems
- Cloud computing
- Version control
- Design testing and debugging

What Skills do Software Engineers Need?

Other beneficial soft skills for a software engineer may include:

- Collaboration and communication
- Multitasking
- Continuous learning
- Attention to detail
- Problem-solving
- Agile project management

The Future of Software Engineering

With our ever-increasing dependency on technology and the growing internet of things, the future of software engineering is bright. Software engineers are employable in nearly every industry, in both large and small organizations.

According to the US Bureau of Labor Statistics, employment of software developers, quality assurance analysts, and testers is projected to **grow 22 percent through 2030**—much faster than the average for all occupations.

Source: US Bureau of Labor Statistics