

08 Software Development Methodologies

Introduction to OOA OOD and UML

2022 Spring

College of Information Science and Engineering

Ritsumeikan University

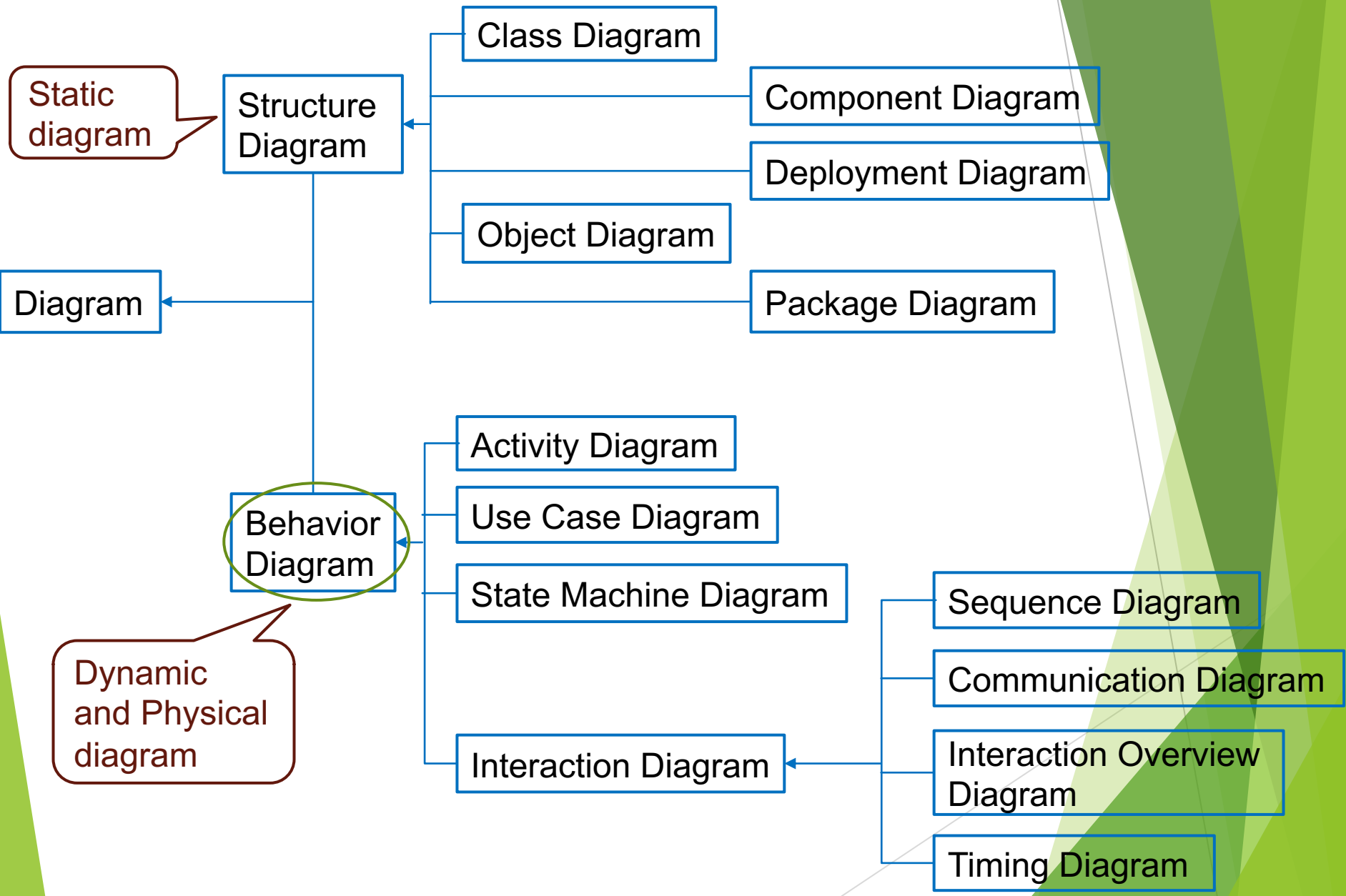
Yu YAN

Outline

- Mid-Term Exam Explanations
- Software Development Methodologies
- Classical Phases in Software Production
- Waterfall Methodology

Outline

- Mid-Term Exam Explanations
- Software Development Methodologies
- Classical Phases in Software Production
- Waterfall Methodology



Why Is a Methodology Needed? (1)

- All software should be produced using some kind of **methodology**
 - In general, a **methodology** is a systematic way of doing things
 - In **software development**, a **methodology** is a repeatable process that we can follow from the earliest stages of **software development** through to the maintenance of an installed system
 - In **software development**, a **methodology** will also include advice or techniques for **resource management**, **planning**, **scheduling** and **other management tasks**

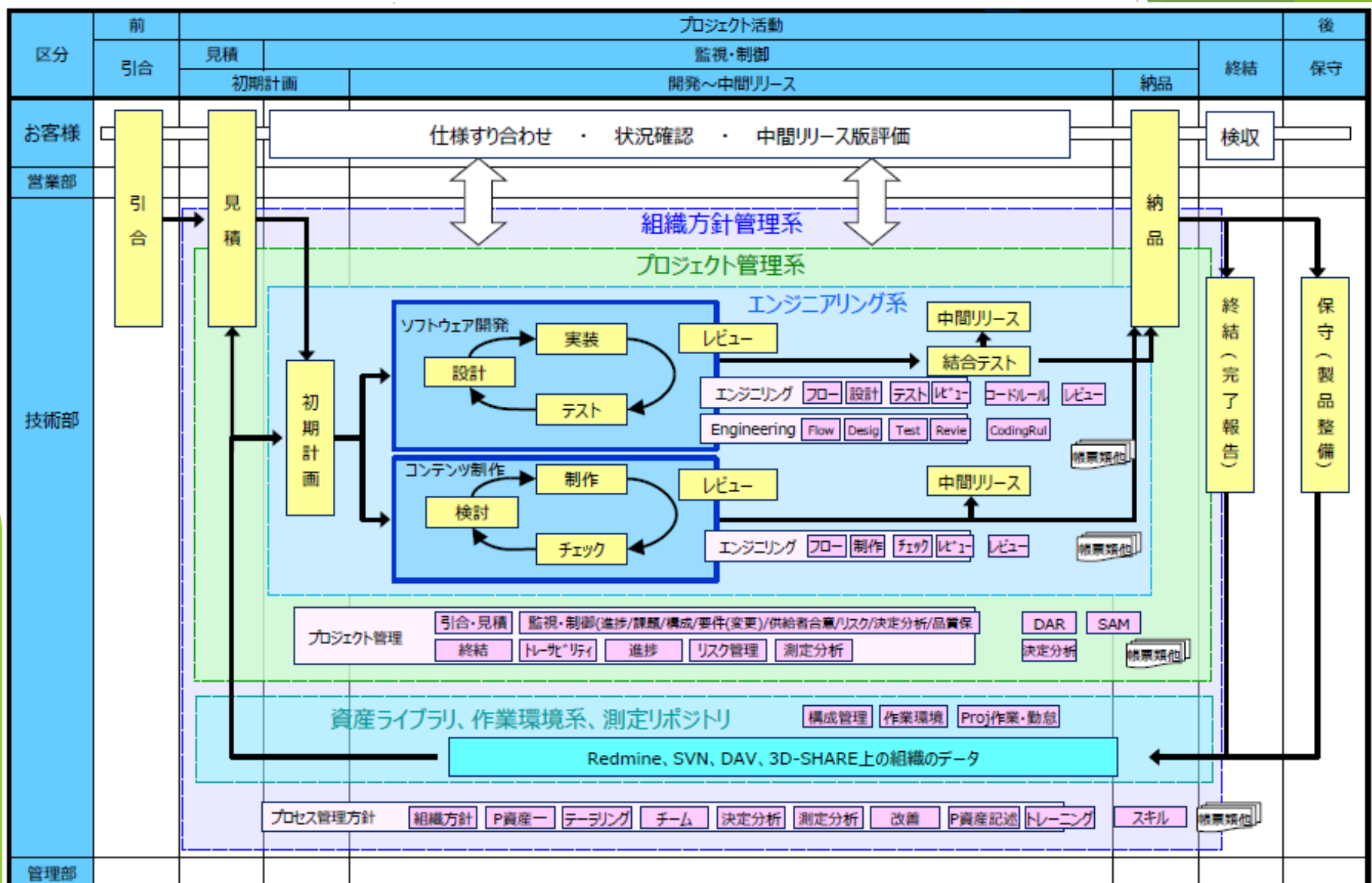
Why Is a Methodology Needed? (2)

- A **methodology** is essential for **software development**, because:
 - It can help to impose discipline on the coding effort
 - Going through even the basic steps of a **methodology** increases our understanding of the problem, improving the quality of our solution
 - Writing lines of code is only one of the many activities in **software development**
 - At every stage, a **methodology** specifies what we should do next
 - A **methodology** helps us to produce code that is more extensible (easier to change), more reusable (applicable to other problems) and easier to debug (because it has more documentation)

Why Is a Methodology Needed? (3)

- A good **methodology** will address at least the following issues:
 - Planning (what next)
 - Scheduling (timing of development)
 - Resourcing (software, hardware, human resources)
 - Workflows (sub-process planning of design, coding, testing, etc)
 - Roles (personnel of development, for example, developer, tester or salesperson)
 - Artifacts (design documents, test tables, pieces of software, manuals)
 - Education (training personnel)

Example of Industrial Methodology



Outline

- Mid-Term Exam Explanations
- Software Development Methodologies
- Classical Phases in Software Production
- Waterfall Methodology

Requirements

- **Requirement capture** is about discovering what we're going to achieve with our new piece of software
- There are two aspects of **Requirements**:
 - Business modeling: it involves understanding the context in which our software will operate (from the standpoint of customers)
 - System requirements modeling (or functional specification): it means deciding what capabilities the new software will have and writing down those capabilities. All relevant functions must be implemented, and all irrelevant functions must be avoided

Analysis

- **Analysis** means understanding of a problem:
 - Relevant entities
 - Entity properties
 - Interrelation among the entities
- We also need to be able to verify our understanding:
 - Include customer as problem domain experts

Design

- In the **design** phase, we work out how to solve the problem based on our experience, intuition and educational background
- **System design** breaks the system down into **logical subsystems (processes)** and **physical subsystems (computers and networks)**, decides how machines will communicate, chooses the right technologies for the job, and so on.
 - UML is a good tool for system design
 - In **subsystem design** we decide how to cut each logical subsystem into effective, efficient and feasible code.

Specification

- A **specification (or class specification)** means “describing the expected behavior of our programming components”
Planning (what next)
 - A **class specification** is a clear, unambiguous description of the way the components of our software should be used and how they will behave if used properly
- **Specification** can be used in the following ways:
 - As a basis for designing test software to exercise the system.
 - To demonstrate that our software is correct
 - To document our software components to the extent that they could be implemented by third parties.
 - To describe how our code can be reused safely by other applications.

Implementation

- **Implementation** is coding (Class interfaces and method interfaces already done at design stage):
 - Write bodies of classes
 - Implement all methods
 - Commenting code
- Coding must be in accordance with design and specifications

Testing

- When our software is complete, it must be **tested** against the **system requirements** to see if it fits the original goals:
 - Test on platform most close to customer PC
 - Build test tables
 - Test using cases
 - Using critical values during testing (min/max possible values, different baud rates)
 - Find bugs (Abnormal, unusual behavior)

Deployment

- In the **deployment** phase, we're concerned with getting the hardware and software to the end users, along with manuals and training materials:
 - Deployment is done at the end-user platform
 - Prepare user manual and reference manual
 - Fix bugs reported by the end user

Maintenance

- When our system is deployed, **maintenance** starts.
- The reason we, as software developers, are interested in **maintenance** is that the **faults (bugs)** that are found in our software:
 - We must find the **faults** and remove them as quickly as possible, rolling out fixed versions of the software to keep the end users happy
 - As well as **faults**, our users may discover deficiencies (things that the system should do but doesn't) and extra requirements (things that would improve the system). From the business point of view, we would hope to fix and improve our software over time to maintain competitive advantage.

“

Outline

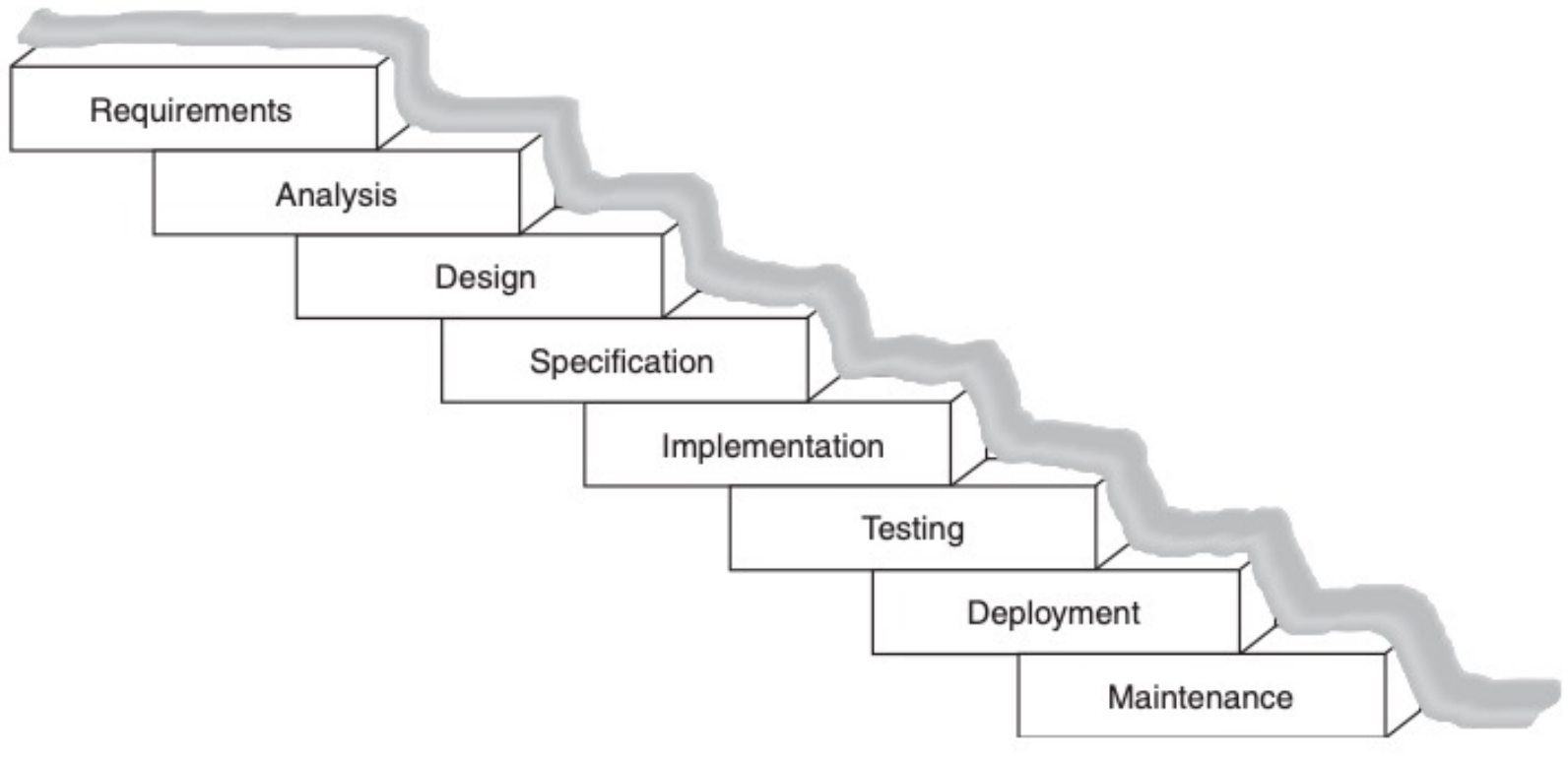
- Mid-Term Exam Explanations
- Software Development Methodologies
- Classical Phases in Software Production
- Waterfall Methodology

What Is Waterfall Methodology? (1)

- **Software engineering:** “software production could be like building a real-world structure”
 - **Engineering** is systematic: if we follow the rules, we will deliver a working product, complete with safety margins to protect against abnormal conditions
 - **Software engineering methodologies** help software production process to be systematic and predictable

What Is Waterfall Methodology? (2)

- **Waterfall Methodology**: development flows smoothly over the classical phases (*requirements, analysis, system design, etc.*), with each phase being completed satisfactorily before the next phase is attempted.



Class Vocabularies

Methodologies (software development), Waterfall methodology

Summary

- The classical phases of software production – requirements, analysis, system design, subsystem design, specification, implementation, testing, deployment and maintenance - were considered.
- Waterfall methodology was mentioned as the simplest software methodology
- Usage of OO tools for software development were considered