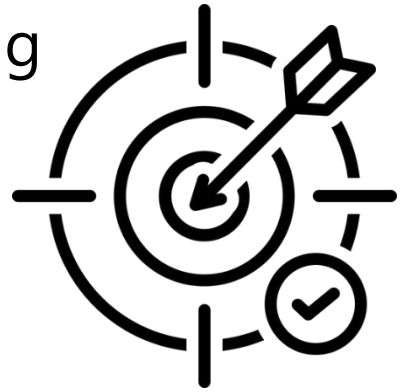
Software Development Lifecycle

Object-Oriented Programming

Mario Simaremare, S.Kom., M.Sc.
Program Studi Sarjana Sistem Informasi
Institut Teknologi Del

Objectives

- The objective of this session is the following:
  - The students are able to describe the general idea of applying the object-oriented approach in the today's software development lifecycle (SDLC).

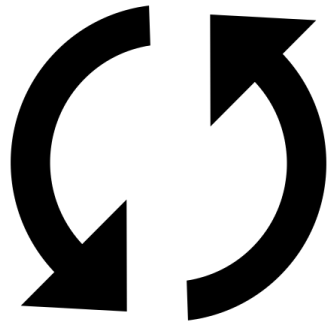


Outlines

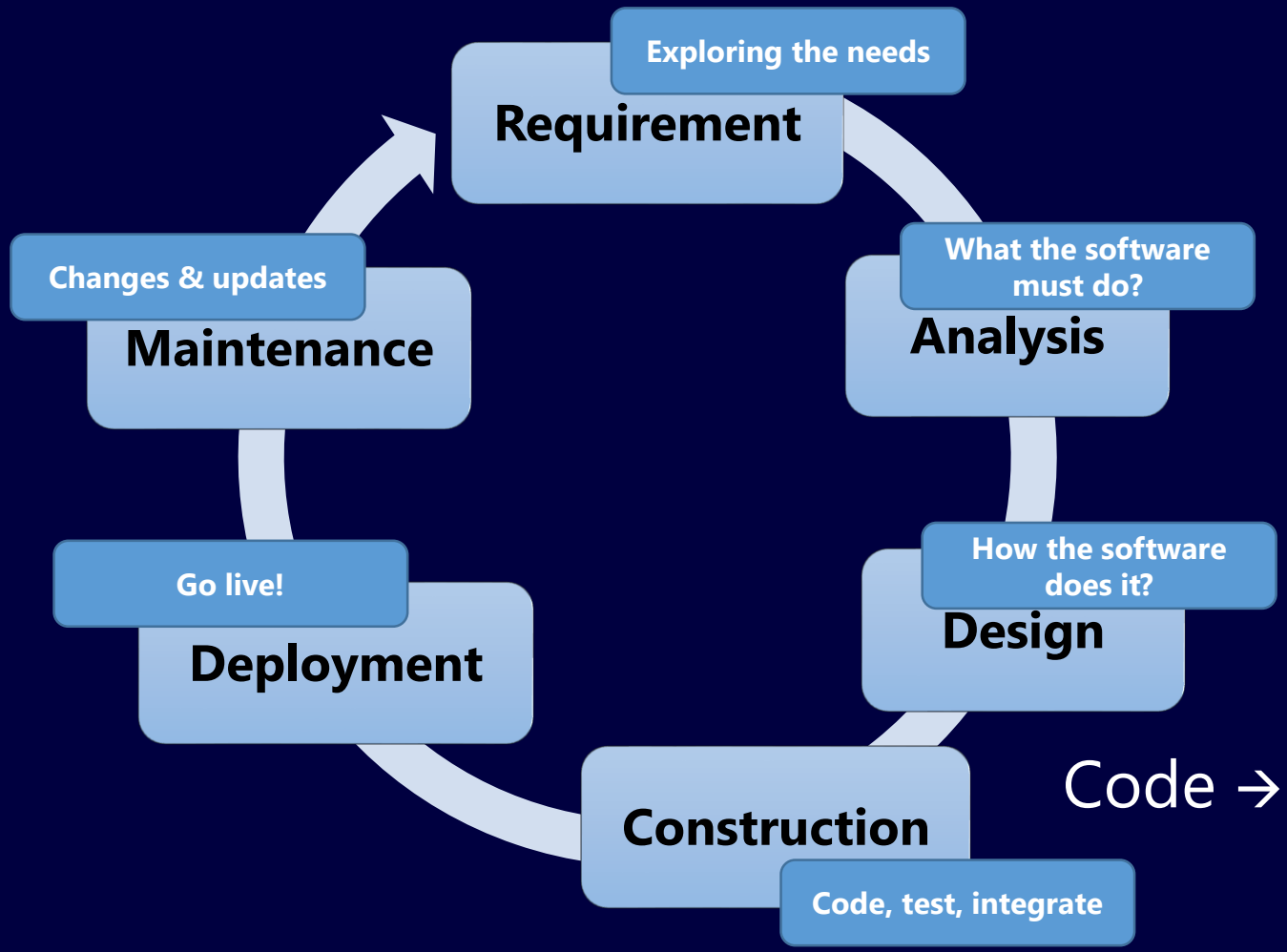
1. Modern Software Development Lifecycle.
2. General phases in Agile methodology.
3. Applying OO.

Modern SDLC

- Today's SDLC practices are based on the Agile methodology.
  - Developer + Client are working as a team.
  - Short time-boxed sprint delivery (continuous delivery).
  - New features are added and integrated continuously.
  - Compared to the traditional approach, it is much flexible where changes are embraced.



General Phases in Agile Methodology



Every requirement (user story) will go through the cycle.

The cycle loops until all the requirements are fulfilled.

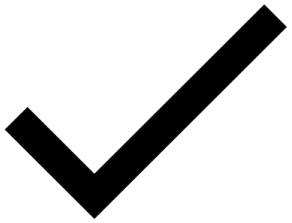
A set of user stories are planed to be delivered in a particular time-boxed (sprint).

Code → test → refactor → retest → done.

Applying OO

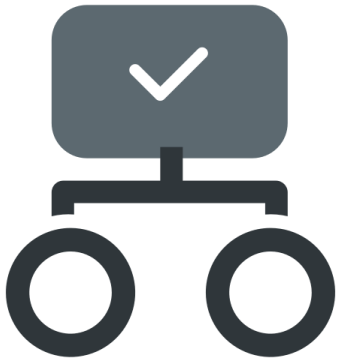
- For each phase, we will apply OO approach and produce artifacts in OO-related models.

- In this course, we are focused on three phases:
  - Requirements analysis.
    - Artifacts: use case, scenario, and domain model.
  - Solution design.
    - Artifacts: interaction and sequence diagrams.
  - Solution construction.
    - Artifacts: source code and working program.



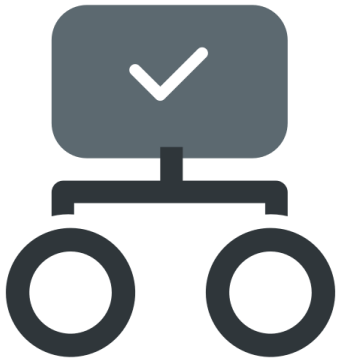
Requirement Gathering and Analysis

- In the requirement gathering and analysis:
  - Working with client to understand their problem (challenge).
  - We are trying to understand the use cases.
  - A use case is then detailed in a scenario.
  - The scenario is used to identify the domain entities.
  - Entity Relationship Diagram is also produced.



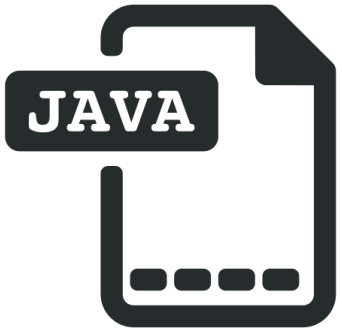
Design phase

- In the design phase:
  - We are trying to define the relationship between entities and their interactions.
  - Class diagram, interaction and sequence diagram, conceptual and physical data model are produced.



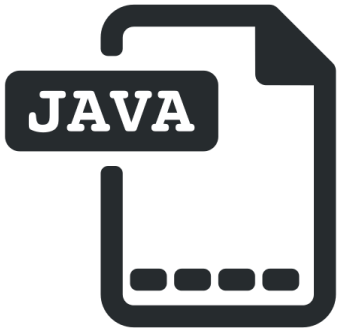
Construction phase

- Construct the designed solution into a working solution.
  - Prepare some test scenarios based on the user stories.
  - Writing the codes, just to make it work.
  - Testing the solution.
  - Once the testing has been passed, refactor the code for a better quality (clean code, performance improvements).
  - Re-testing to ensure everything is still working as expected.



Other phases

- Deployment, maintenance, and other phases are not covered in this course.



References

- Cay Horstman. Core Java.
- Matt Weisfeld. The Object-Oriented Thought Process.

Thank you