

# **SE202**

# **Introduction to Software Engineering**

## **Lecture 9**

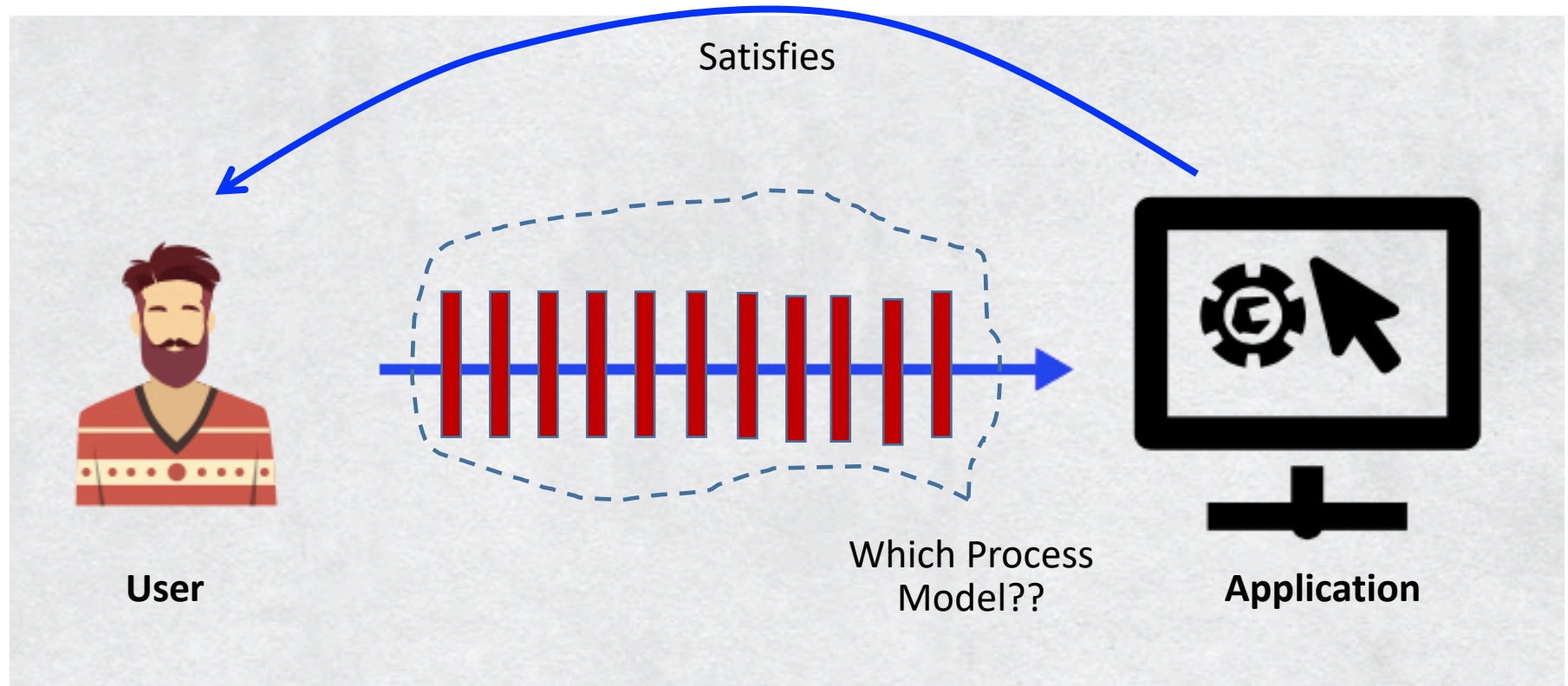
### **Software Process**

Pathathai Na Lumpoon

# Software Development Process

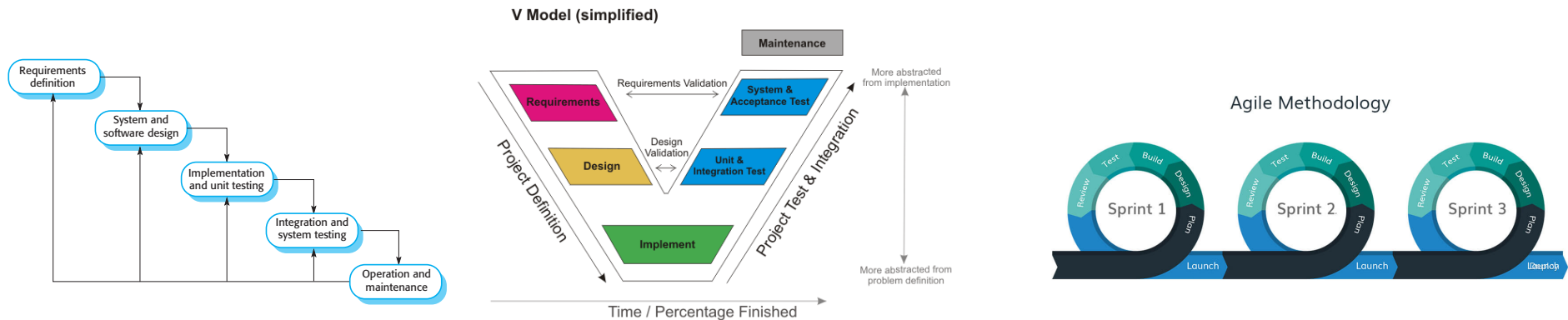
- SDLC consists of:
  - Requirement Analysis
  - Design
  - Implementation
  - Testing
  - Deployment
  - Maintenance
- Each SDLC phase **MUST** have
  - clear process steps, document, reviewable and planning to the next phase

# Basic Problem



# Software Process Model

- A software process model is an abstract representation of a software process. Each model presents a description of a process that can be used to explain a specific approach to the **software development**.
  - E.g. Ad-hoc software development, code-and-fix, waterfall, evolutionary, rapid prototyping, spiral, agile and etc.



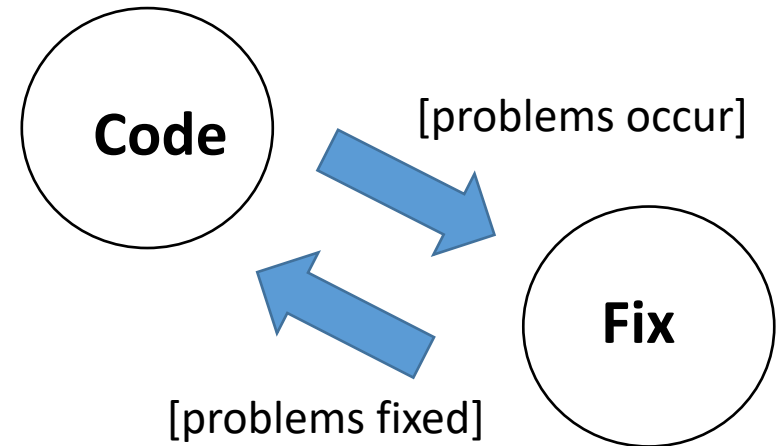
- Different process model is for different SW development purpose. You should know strength and weakness of each for choosing model properly.

# Software development models

- There are two categories:
  - Predictive development model
    - Most requirements, processes, schedule and technologies are anticipated
    - Work well for projects where a clear and complete definition of the whole system is identified
    - Use when the client needs clarity on the target delivery based on the agreed-upon scope
    - E.g. Waterfall model, Incremental waterfall model and V model
  - Adaptive development model
    - Give opportunities to change direction
    - Work well with changing or uncertain requirements
    - Use when the end goals of projects are not clearly defined
    - Use when the project calls for experimental software design
    - E.g. Prototypes, Spiral and Agile

# code-and-fix model

Without spec or design



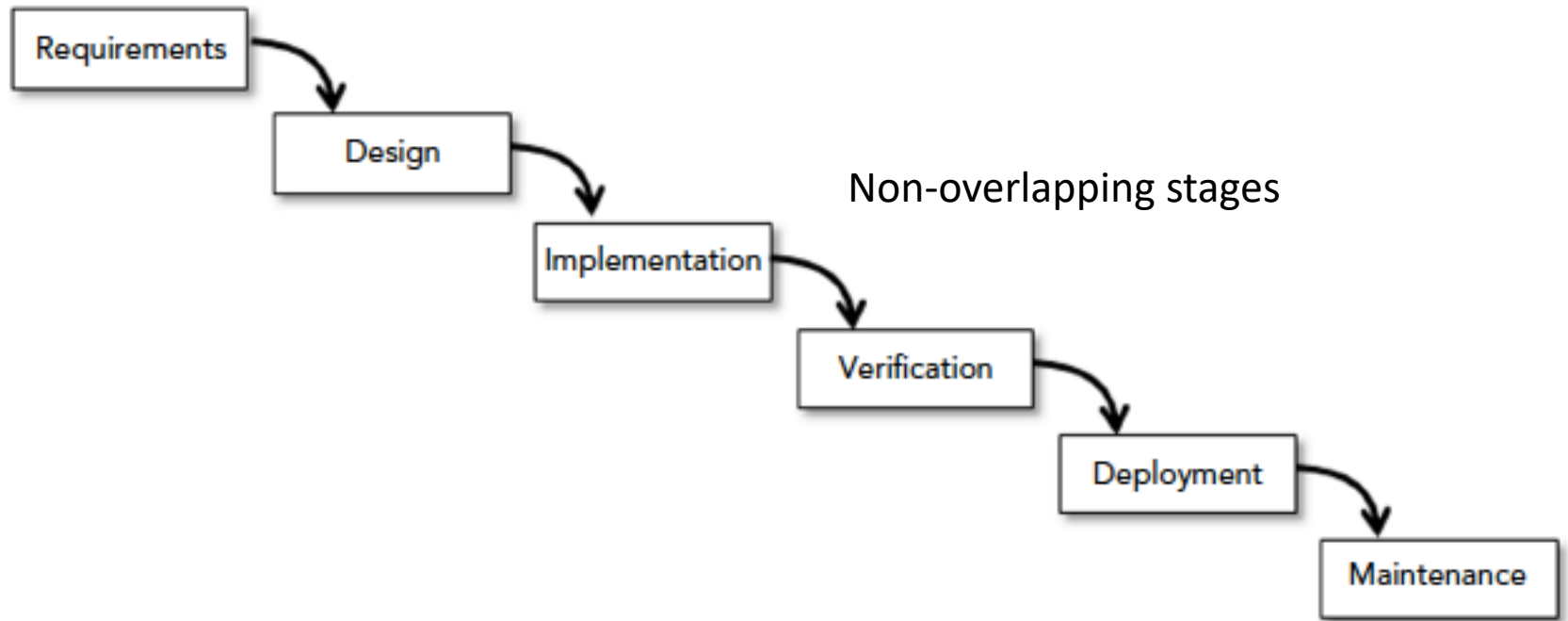
- Most frequently used development methodology in SE
  - Get the immediate feedback
- No initial plan → Ad-hoc approach
- Simple two phases model (Code and fix)
- Start coding without specifications or any attempt at design, fixing problems when occurs, until the project is complete.
- Increase poorly structured code → maintenance becomes expensive
- Suitable for small programming or small project that does not have plan for future development

**not recommend!!**

# Predictive development model

- Waterfall model
- Waterfall with overlapping phases
- Incremental Waterfall model
- V model

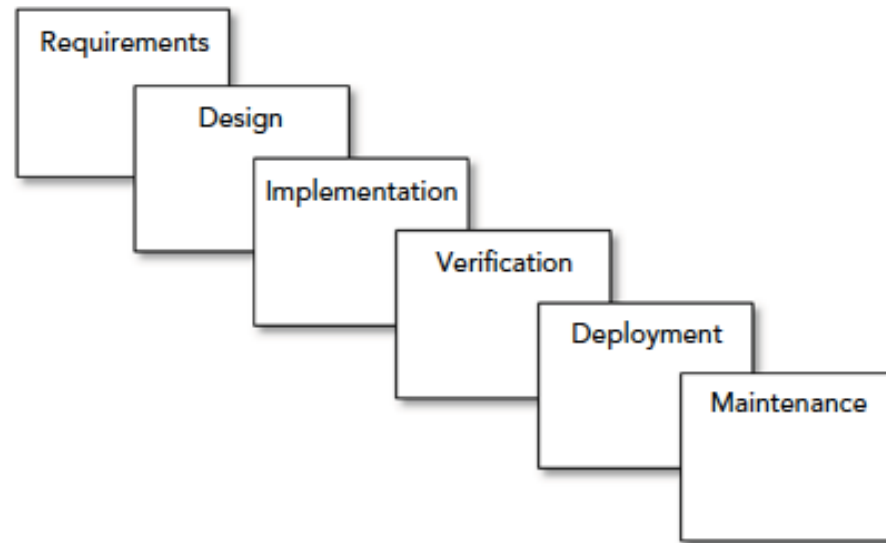
# Waterfall model



- The classic and oldest model of SE
- Widely used in government projects and quality control projects.
- Each phase requires formal review and intensive documentation
- The pure waterfall model, the requirements must be done before the design, and design must be complete before coding. → There is no overlap between stages.
  - In practice, errors of requirements often found at design or implementation or testing
- The model allows returning back to an earlier phase → costly rework

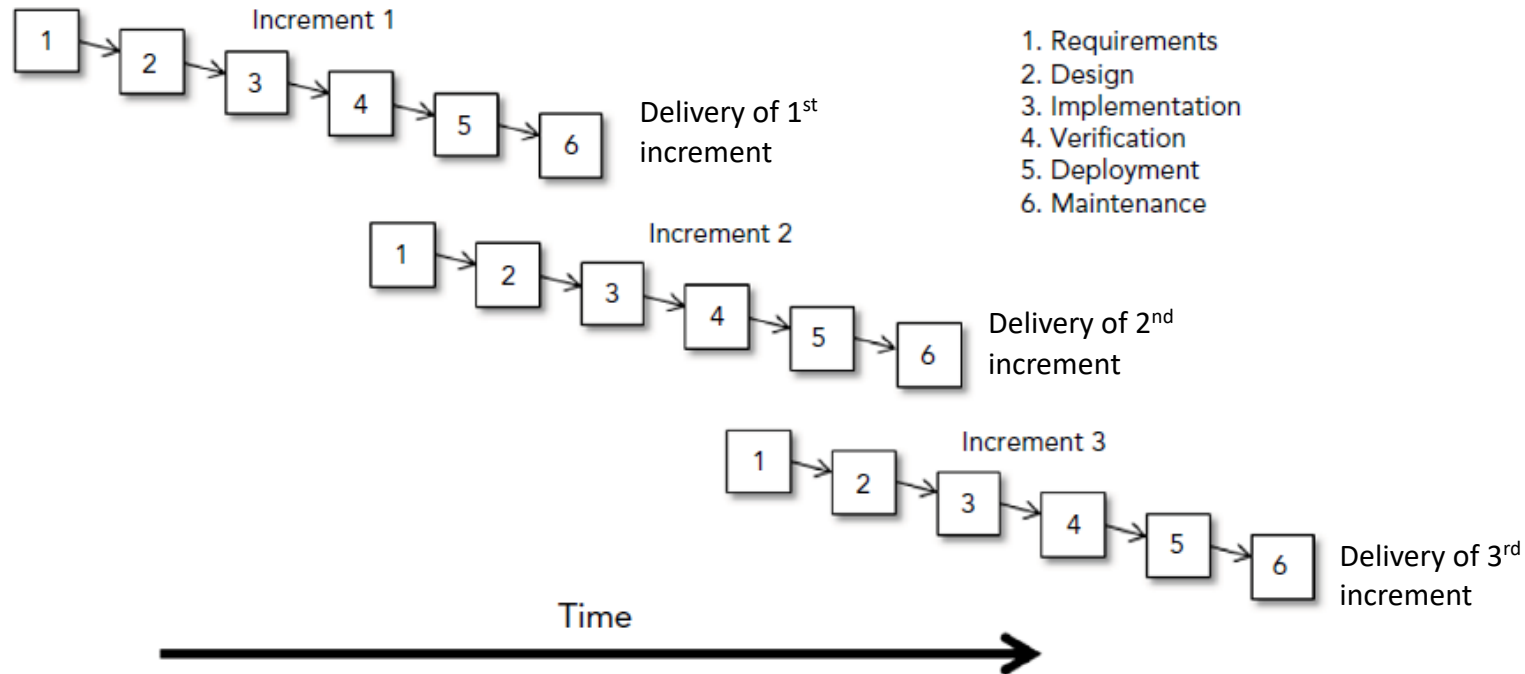


# Waterfall with overlapping phases



- Similar to the waterfall model except the steps are allowed to overlap.
- Advantages:
  - Team members can work without waiting for others
  - Allow later phases modify earlier phases
    - Integrate feedback from the design phase into the requirements.
- Disadvantage:
  - Progress is more difficult to track

# Incremental Waterfall model

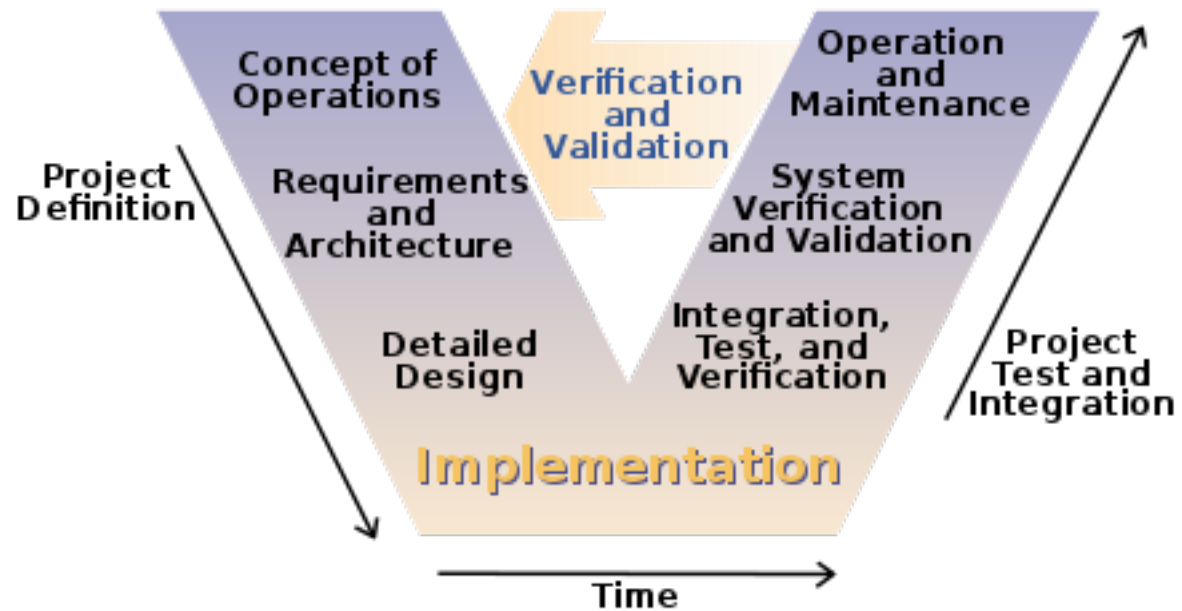


- Model is designed, implemented and tested **incrementally** (a little more is added each time) until the product is finished
- A series of separate waterfall iterations.
- Each iteration ends with the delivery of a usable application called **an increment**.
- Each increment includes more features than the previous one

# Incremental model analysis

Advantages	Disadvantages
Working software is quickly produced early during the software life cycle.	Needs good planning and design.
Model is more flexible as changes to scope may be made to every increment.	Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
It is easier to test and debug during a smaller iteration.	Total cost is higher than waterfall.
A customer can respond to each increment.	
Errors are easy to be identified.	

# V model



- An extension of the waterfall model
- Based on the association of a testing phase for each corresponding development stage.
- For every single phase in the development cycle, there is a directly associated testing phase.

# Predictive models analysis

Advantages	Disadvantages
<p>Give structure of the project</p> <p>Plan everything ahead</p> <p>Development plan and documentation</p> <p>Easier to add new people</p>	<p>Cannot handle well with the changes</p> <p>Requirements, technologies, team experiences, economics, management, time, budget must be identified.</p>
<p>Less cost if everything goes according to the plan</p>	<p>Not flexible</p> <ul style="list-style-type: none"><li>- It is difficult to fully specify requirement at the beginning of the project</li><li>- The amount of documentation is excessive and inflexible</li></ul>
	<p>Less customer involvement</p>

# Success Indicators

- User involvement
- Clear vision
- Limited size
- Realistic
- Established technology

# Failure Indicators

- Incomplete requirements
- Unclear requirements
- Changing requirements
- No resources

# Quiz time

- Go to [www.menti.com](https://www.menti.com) and use this code