

# Software Process Frameworks

*Sections 2.1 & 2.3 (Sommerville)*  
*Chapter 3 (Sommerville)*

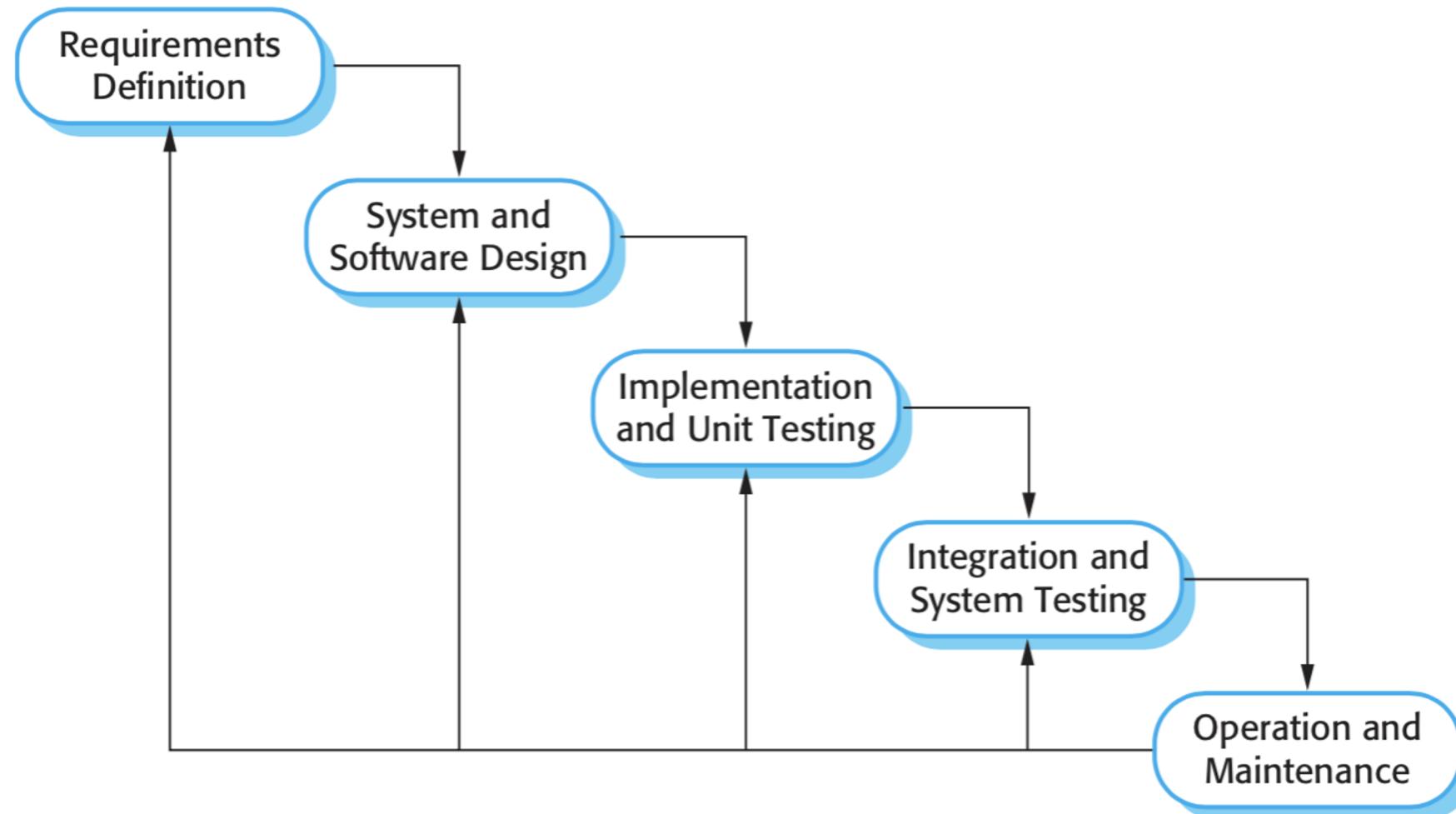
Fall Semester 2021  
1st Semester 1443 H

# Software Process Frameworks

Waterfall

Incremental

# The Waterfall Model



Video: [https://www.youtube.com/watch?v=Y\\_A0E1ToC\\_I](https://www.youtube.com/watch?v=Y_A0E1ToC_I)

# The Waterfall Model

## Advantages

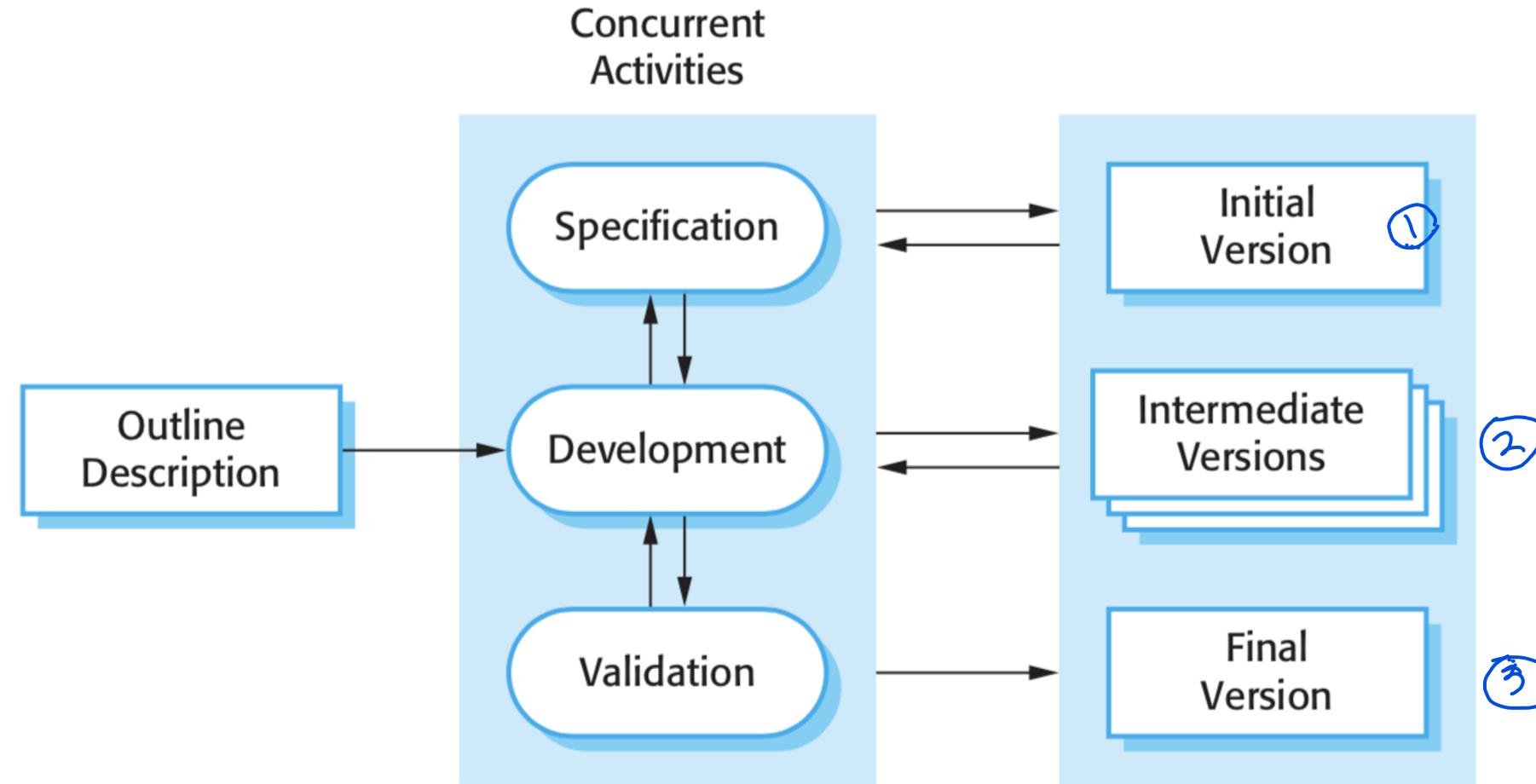
- Documentation is produced at each phase.
- This makes the process visible so managers can monitor progress against the development plan.

## Disadvantages

- Inflexible partitioning of the project into distinct stages.
- Commitments must be made at an early stage in the process, which makes it difficult to respond to changing customer requirements.

The waterfall model should only be used when **the requirements are well understood** and **unlikely to change** radically during system development.

# The Incremental Model



# The Incremental Model

## Advantages

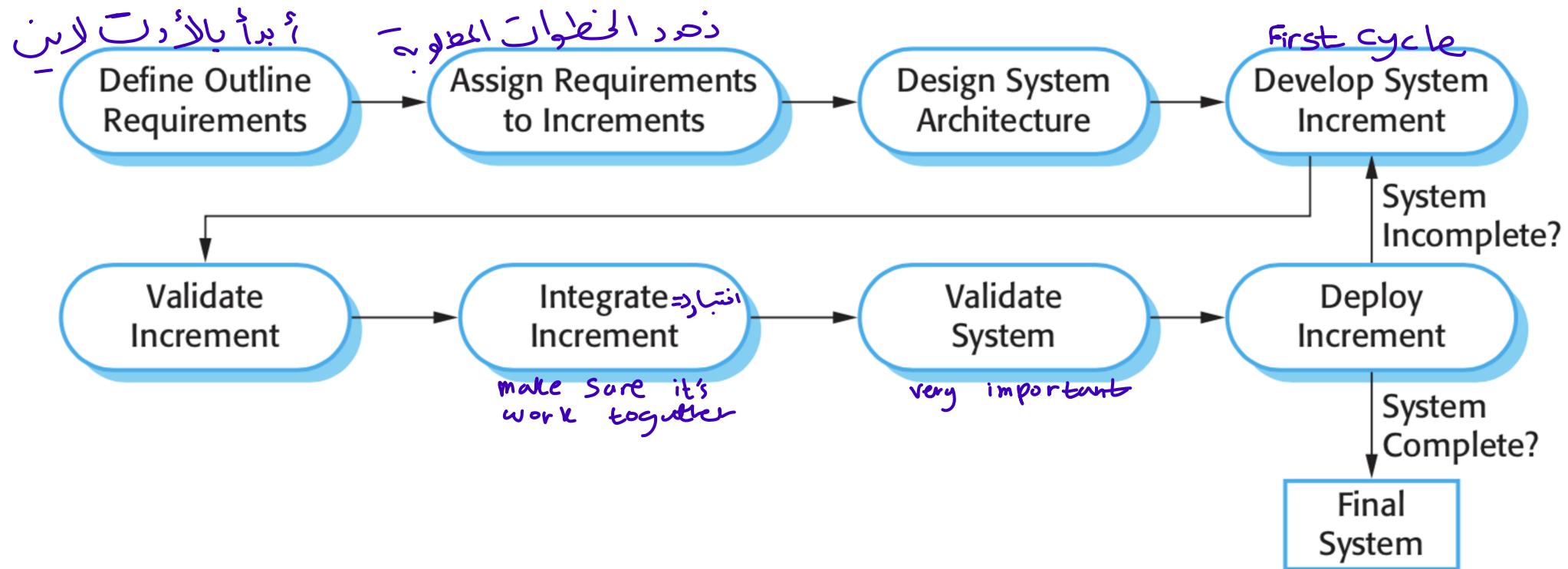
- The cost of accommodating changing customer requirements is reduced. *⇒ The change very easy*
- It is easier to get customer feedback on the development work that has been done.
- More rapid delivery and deployment of useful software to the customer *after each cycle*

*we get feedback from Costmer*

## Disadvantages

- The process is not visible for managers.
  - System structure may degrade as new increments are added.
- 
- The incremental approach can be used for application systems. *⇒ easier to get feedback*
  - Not suitable for large complex systems, or safety critical systems *good for high user interaction*

# Incremental Delivery



# Agile Software Process Model

# In Which Framework does Agile Fit?

Waterfall



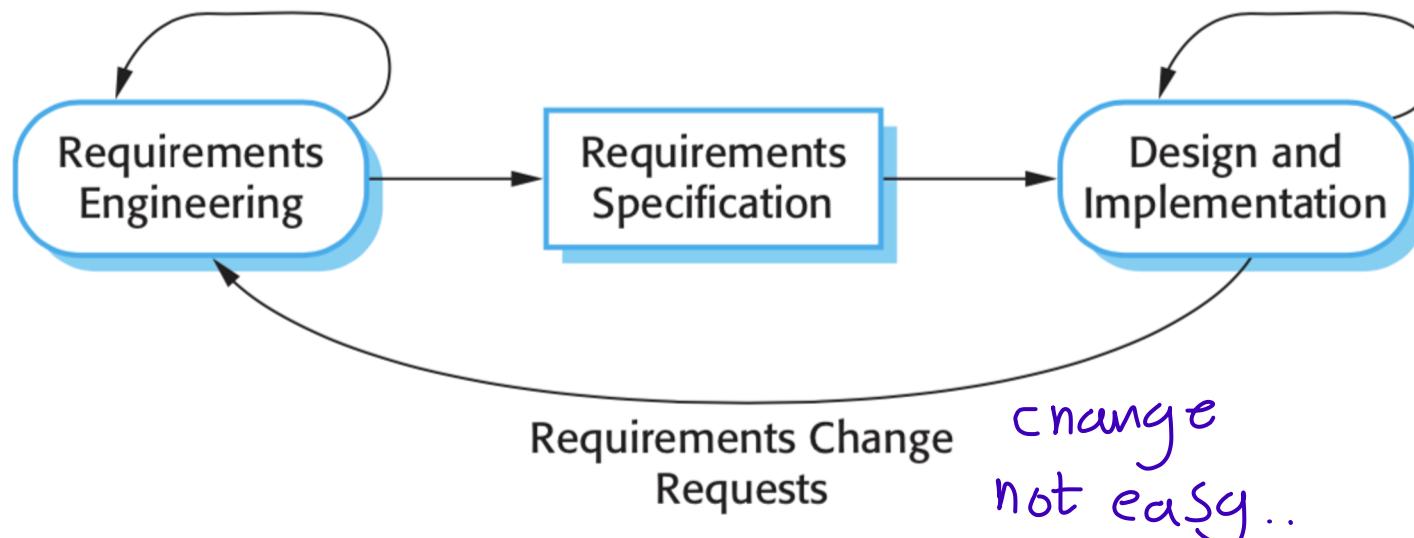
Incremental

Agile

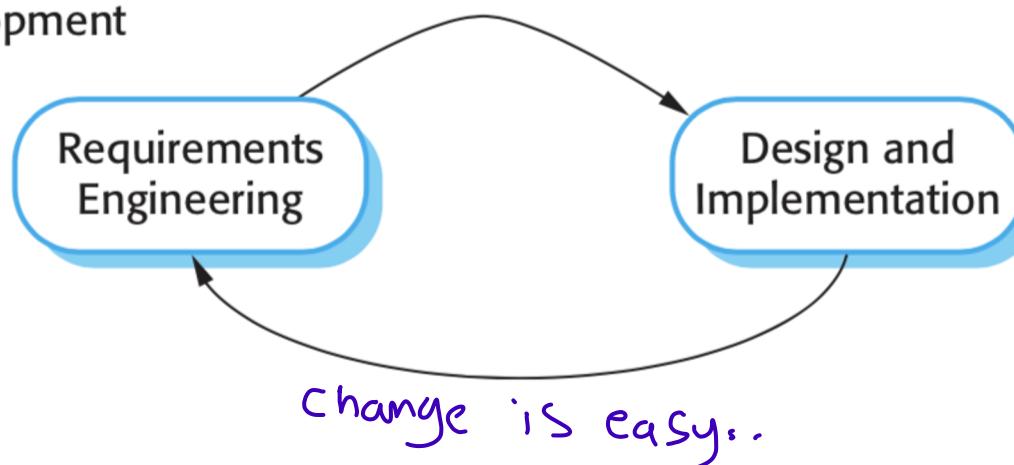
# Agile and Plan-Driven Development

- Agile approaches to software development consider design and implementation to be the central activities in the software process.
- They incorporate other activities, such as requirements elicitation and testing, into design and implementation.  
*output for one cycle is input for another cycle.*
- By contrast, a plan-driven approach to software engineering identifies separate stages in the software process with outputs associated with each stage.
- The outputs from one stage are used as a basis for planning the following process activity.

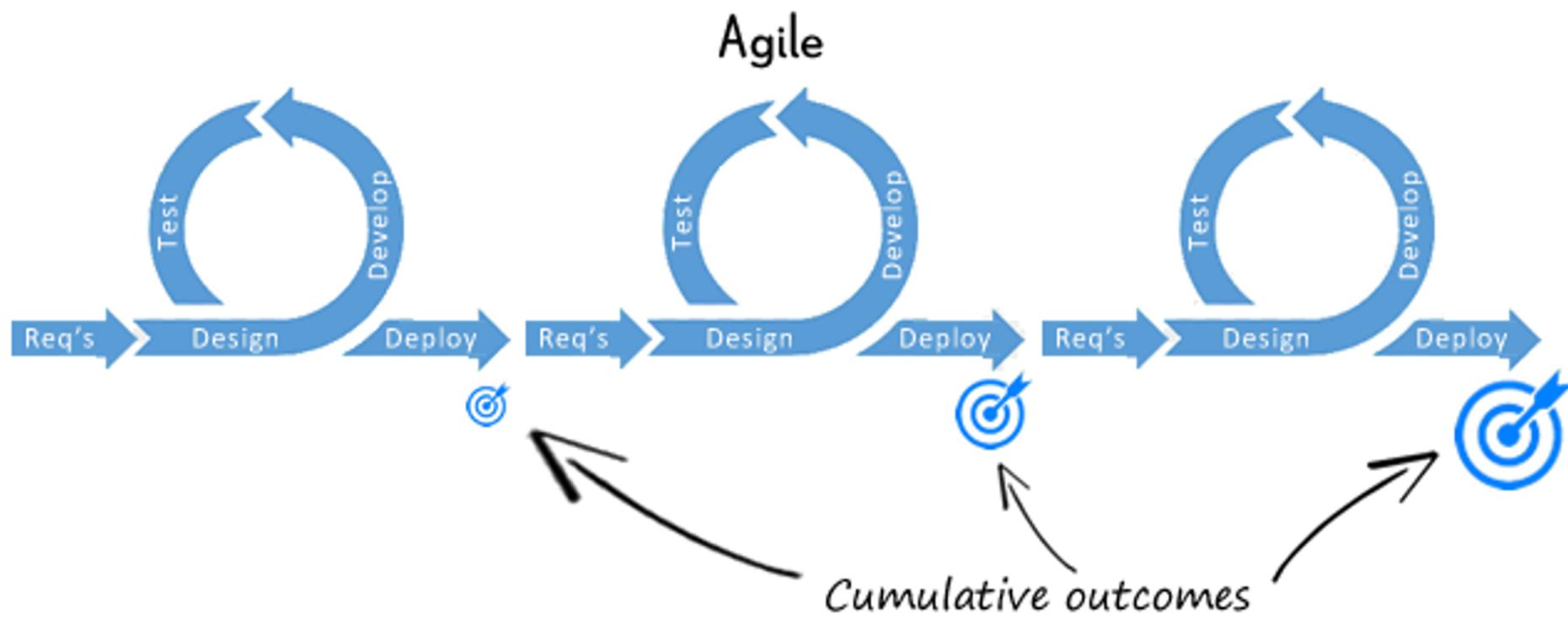
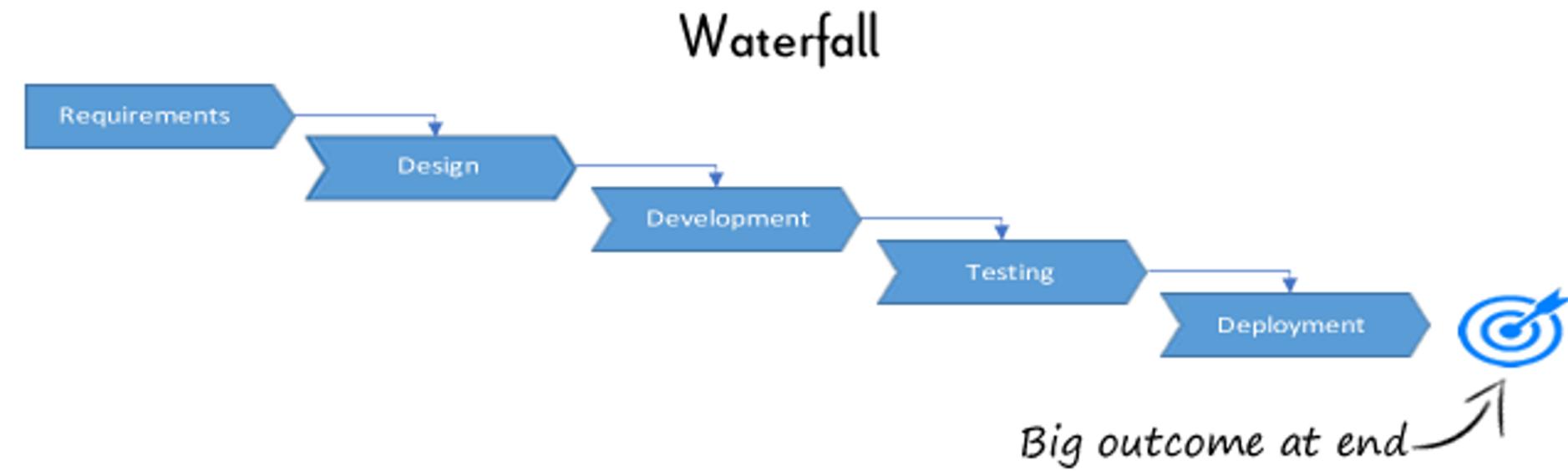
## Plan-Based Development



## Agile Development



**Plan-driven and agile approaches to system specification**



## ACTIVITY:

How to Decide on the Balance Between a Plan-Based and an Agile Approach?

# Software Systems

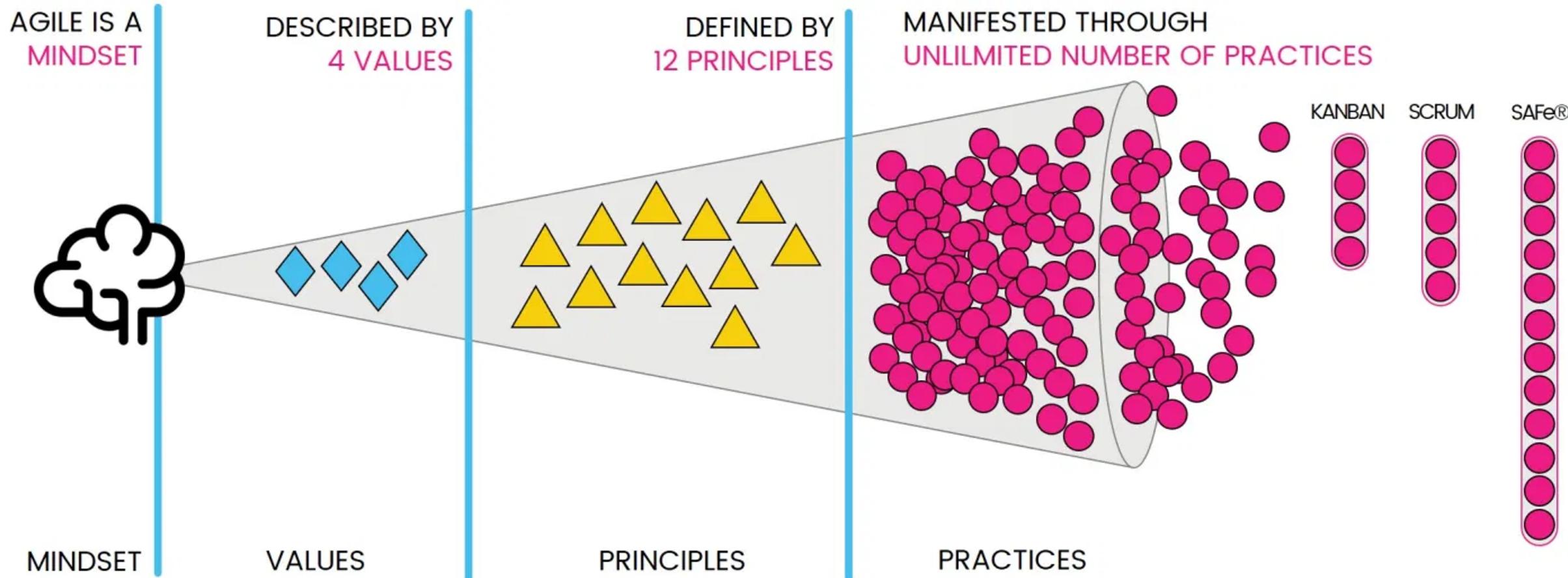
1. A system to perform robotically assisted heart surgery. *waterfall*
2. A system for a digital smart camera to capture images, store image data, and to represent this data.
3. A virtual reality system to support learning physics for high school students.
4. A university accounting system that replaces an existing system.
5. An interactive travel planning system that helps users plan journeys with the lowest environmental impact. The client would like to offer the services for *feedback* customers before the other competitors. - *Agile*
6. A mobile App that allows users to track their workout & fitness training. - *Agile*  
*if the System expecting Long time || high skills people in Agile*  
*waterfall is the best*

# When to Use Agile Methods?

- For application development where the system **requirements usually change rapidly** during the development process.
- To **deliver** working software **quickly to customers**, who can then propose new and changed requirements to be included in later iterations of the system.
- To cut down on process bureaucracy by **eliminating documentation** that will probably never be used.

غایباً" ادAgile  
اذاً نعم نعم الْهَوَّد

# What is an Agile?



Adapted from Ahmed Sidky's Agile Mindset

Image source: <https://zenexmachina.com/agile-as-a-mindset-agile-as-behaviour/>

Thought leaders in the software industry formalized the agile movement in 2001 with the publication of the Manifesto for Agile Software Development.

*“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*

- 
- 1. Individuals and interactions over processes and tools**
  - 2. Working software over comprehensive documentation**
  - 3. Customer collaboration over contract negotiation**
  - 4. Responding to change over following a plan**

*That is, while there is value in the items on the right, we value the items on the left more. “*

# 12 AGILE PRINCIPLES BEHIND THE AGILE MANIFESTO

- 1 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- 2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- 3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- 4 Business people and developers must work together daily throughout the project.
- 5 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 6 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 7 Working software is the primary measure of progress.
- 8 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 9 Continuous attention to technical excellence and good design enhances agility.
- 10 Simplicity – the art of maximizing the amount of work not done – is essential.
- 11 The best architectures, requirements, and designs emerge from self-organizing teams.
- 12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

# Key Points

- Software Process Models
  - Waterfall
  - Incremental
- The software crisis and agile to the rescue
- Agile and plan-driven development
- When to use agile
- The agile manifesto
- Principles of agile methods

# Thank you!