Week 1: Deconstructing the Software Development Process

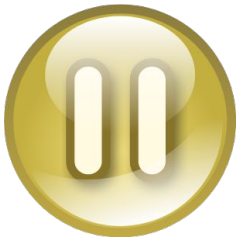
Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

Thinking of Software in terms of Components

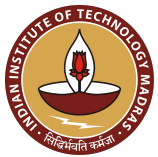
Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

Reflection Spot

We discussed that Amazon has several components. Could you list a few of them ?



Please pause the video and write down your responses

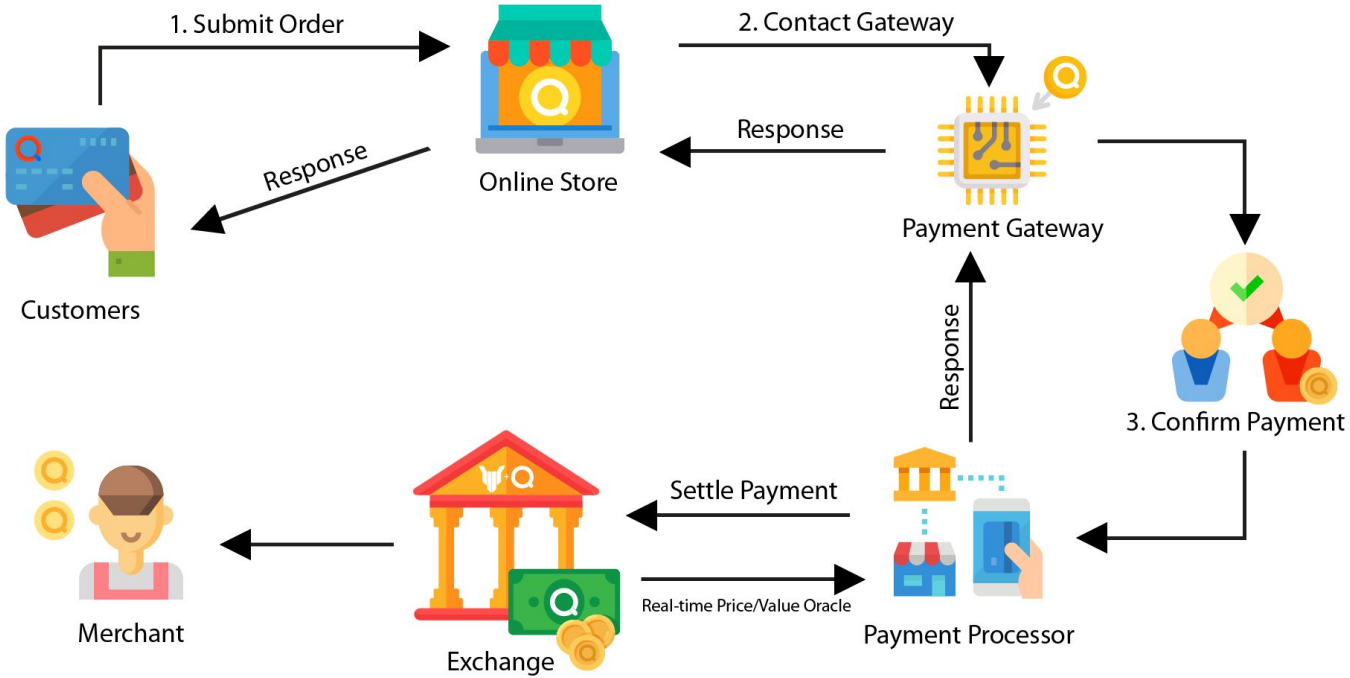


Inventory Management

- Inventory gets updated based on current purchasing and seasonal trends

- My homepage is customised based on my shopping and viewing history

Payment Gateway



https://cdn.truelancer.com/upload-original/1658216-Payment-Gateway-Process.png

Summary

- Discussed various components of a large software system E.g. Amazon
  - Inventory Management System
  - Payment Gateway

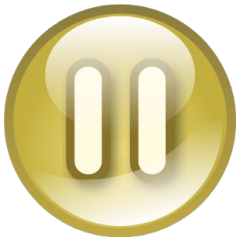
- Software can be divided into separately addressable components called **modules** that are integrated to satisfy requirements

Software Development Process - Requirement Specification

Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

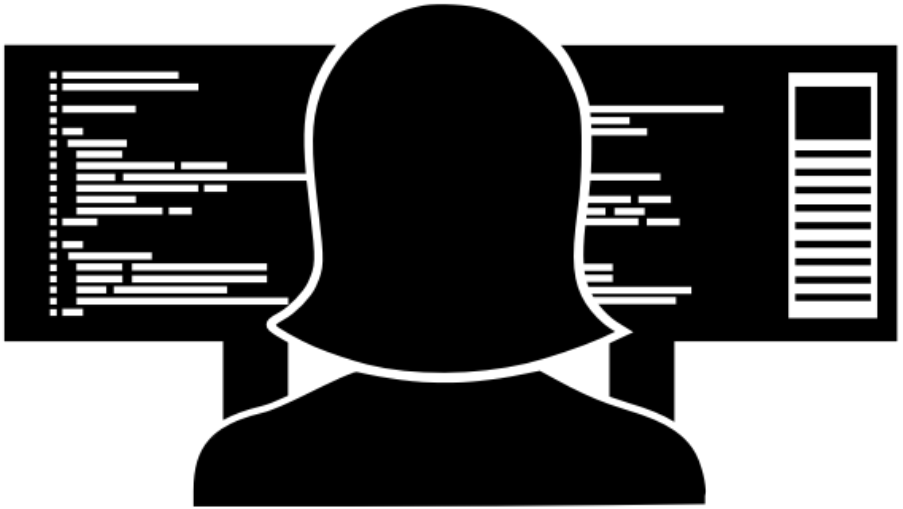
Reflection Spot

What do you think is the first step in creating a new software component?



Please pause the video and write down your responses

First Step in Creating Software



First Step in Creating Software

- Study existing components of the system -
  to understand how the new component will interact with existing components

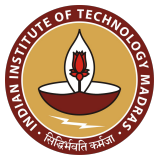
First Step in Creating Software

- Look at similar systems to understand features
  e.g. PayTM, PhonePe

Requirement Specification

Requirements

- Goals the implemented system should have
- Should cater to the need of clients



Client - External User

Example - Mobile banking software serves bank customers

Client - Internal to your Company

Example -
Building an internal employee resource portal

- Internal products team → To build this portal

Client - Another Software

Example -
A payment gateway interfaces with another ecommerce system

Clients

- Think about **who** is going to use your software, for **what purpose**, and in **what way**

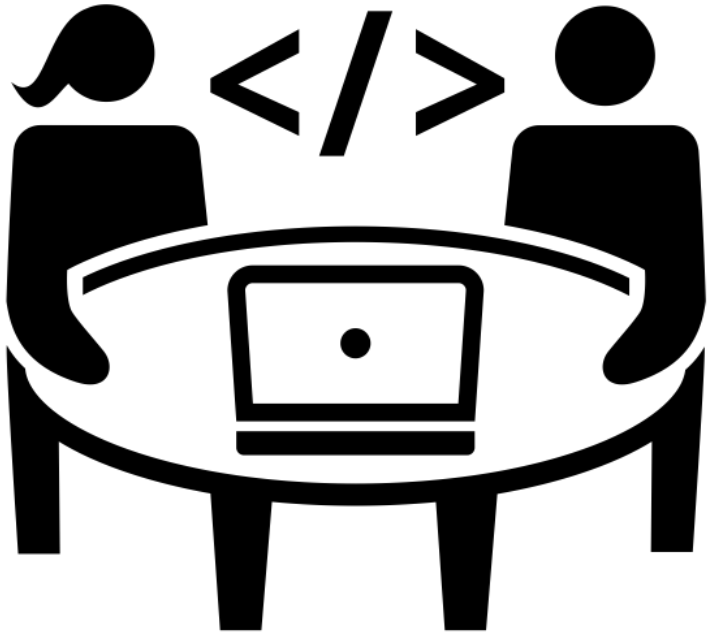
Summary

- **Requirement specification** - First step in the software development process
- **Clients** - end users of the software
- Need to ensure that the requirements capture clients' needs

Software Design and Development

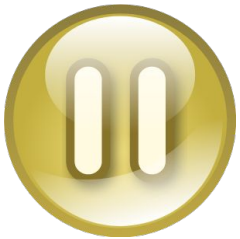
Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

Software Development Team



Reflection Spot

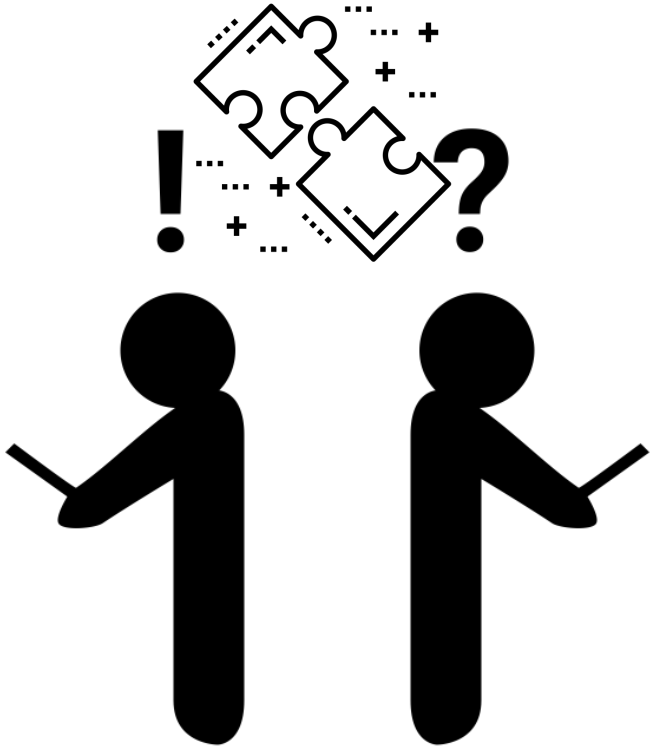
What difficulties are you likely to encounter if you directly start coding?



Please pause the video and write down your responses

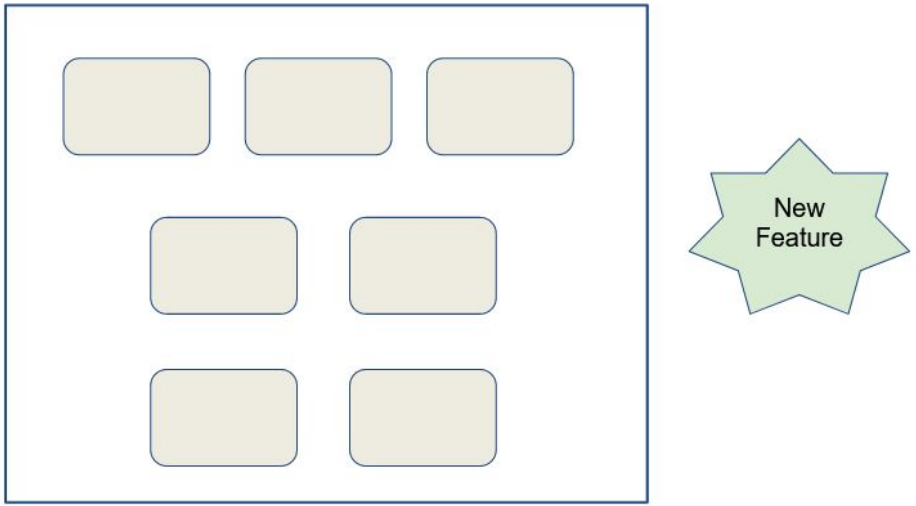
Issues during integration

- Different developers may have different ideas about how the functionality should be implemented



Difficulties while adding new features

- Adding new features - big picture view of the system is necessary



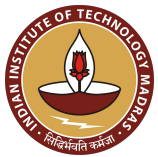
Software Design

Requirements

- Goals the implemented system should have
- Should cater to the need of clients

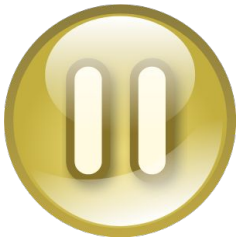
Design

- Big picture view of the software system
- Provides a structure to the software system

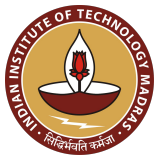


Reflection Spot

How do you think people work in the development phase?



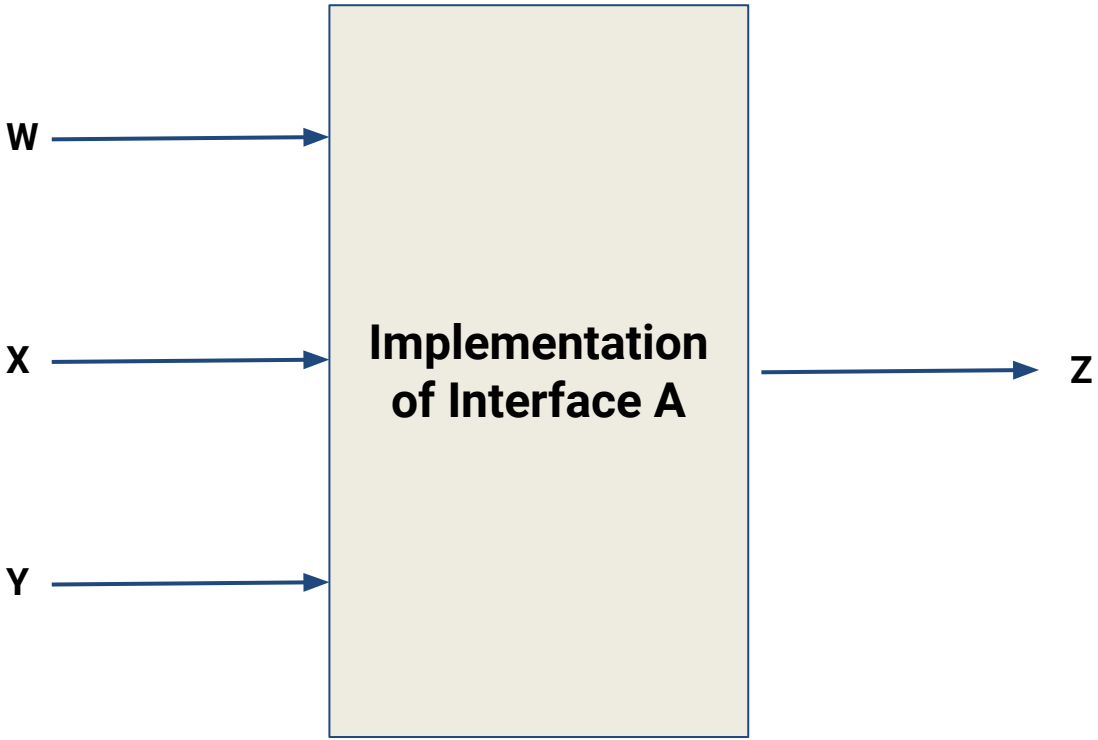
Please pause the video and write down your responses



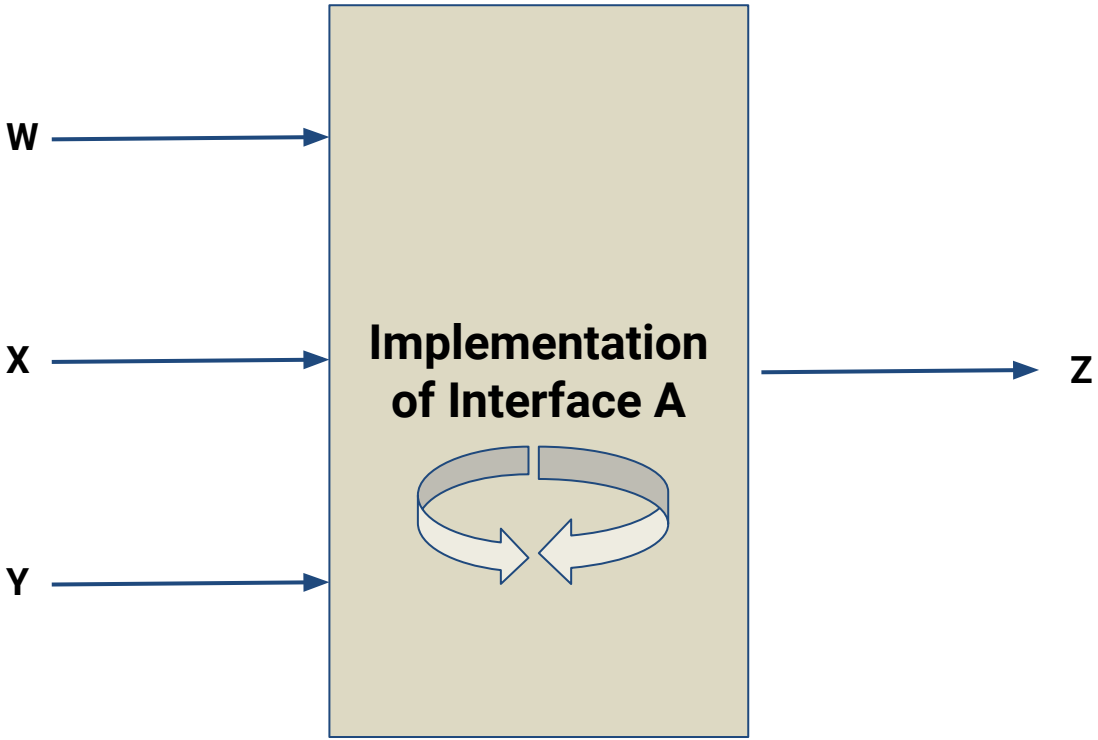
Development Phase

Development Phase

Development Phase



Development Phase



Software Development

Requirements

- Goals the implemented system should have
- Should cater to the need of clients

Design

- Big picture view of the software system
- Provides a structure to the software system

Development

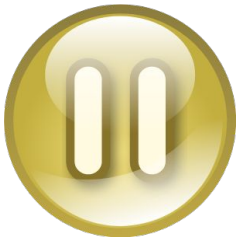
- Write code based on the requirements and the design
- Usually distributed
- Developer documentation and precise interface definitions

Software Testing and Maintenance

Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

Reflection Spot

Why do you think testing is necessary? What can go wrong if we release the software directly?



Please pause the video and write down your responses

Importance of Testing

- Testing is done to ensure that the software behaves according to the requirements

- Many bugs might still exist in the system

Importance of Testing

- In 2002 - $59.5 billion in losses [1]
  In 2016 - $1.1 trillion [2]

- A failure to address bugs can even cause severe catastrophes

[1] Newman, M., 2002. Software errors cost us economy 59.5 billion annually. NIST Assesses Technical Needs of Industry to Improve Software-Testing
[2] https://medium.com/@ryancohane/financial-cost-of-software-bugs-51b4d193f107

Testing

- Unit Testing
- Integration Testing
- Acceptance Testing

Testing Methodologies

Alpha Testing

Conducted by: internal employees in a lab/staging environment

Goal: catch as many issues as possible before the product has been released to the public

Testing Methodologies

Alpha Testing

Conducted by: internal employees in a lab/staging environment

Goal: catch as many issues as possible before the product has been released to the public

Beta Testing

Conducted by actual users in a real-live setting

Maintenance Phase

Maintenance - After the feature is rolled out, monitor how users are using the feature

Purposes of maintenance

- Monitor what users are doing, and how they are using the software.
- Change the code for upgrades/updates
- Add features

Software Development Process

Requirements

- Goals the implemented system should have
- Should cater to the need of clients

Design

- Big picture view of the software system
- Provides a structure to the software system

Development

- Write code based on the requirements and the design
- Usually distributed

Testing

- Ensures that the software behaves according to the requirements

Maintenance

- Monitor what users are doing
- Change code for updates

Software Development Models - Plan and Document Perspectives

Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

Beginning of Software Engineering Discipline

- Experience in previous projects
  - “Build and Fix”

- Good Principles and practices + Research Innovations ➜ Software Engineering

- Well-defined set of activities -
  - Software life cycle model
  - Software Development Life cycle (SDLC)
  - Software development process model

Software Lifecycle

Different stages/phases/activities over which a software evolves from the initial customer request to a fully developed software

Software Development Lifecycle

Requirements

- Goals the implemented system should have
- Should cater to the need of clients

Design

- Big picture view of the software system
- Provides a structure to the software syste

Development

- Write code based on the requirement
- Usually distributed

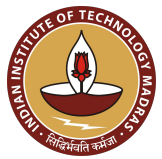
Testing

- Ensures that the software behaves according to the requirements

Maintenance

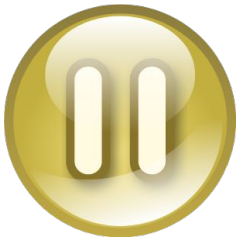
- Monitor what users are doing
- Change code for updates

Plan and Document Perspective - Waterfall model



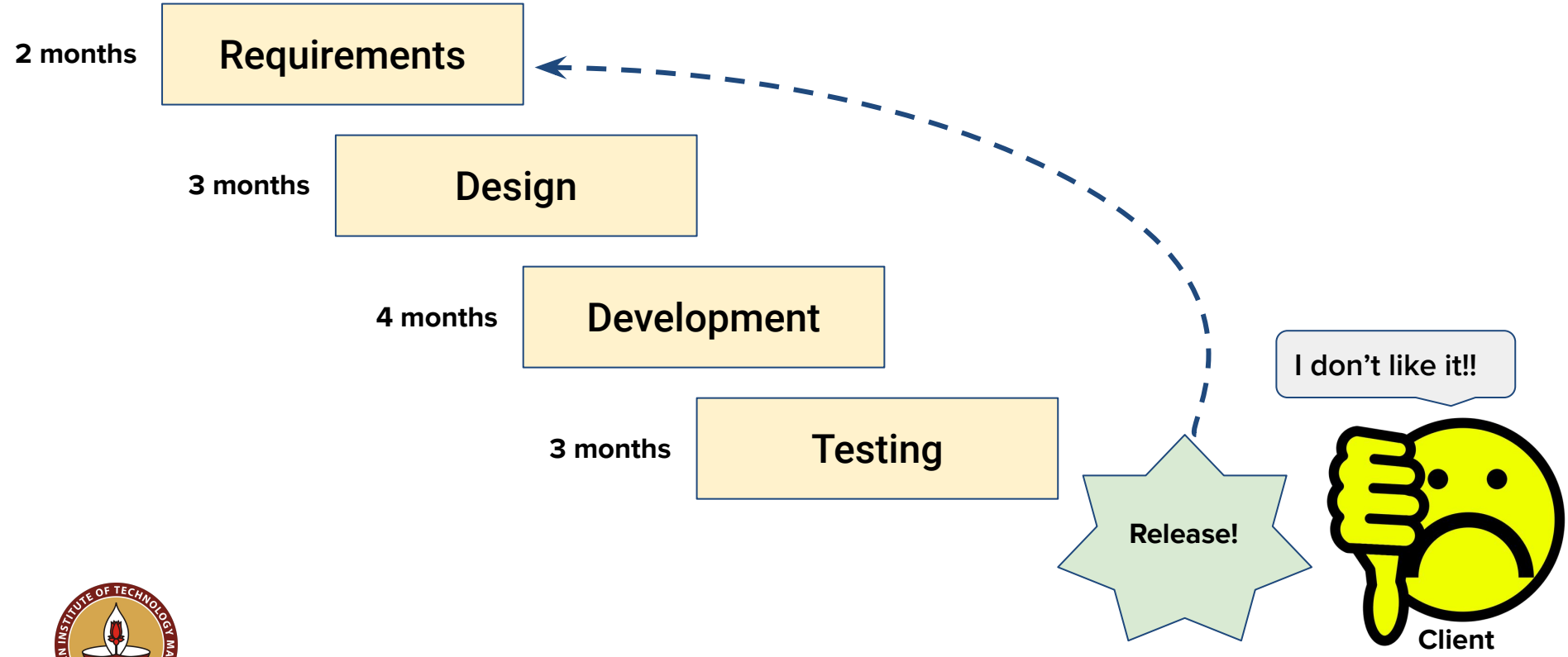
Reflection Spot

What could go wrong if we follow these phases sequentially?



Please pause the video and written down your responses

Drawbacks of the Waterfall Model

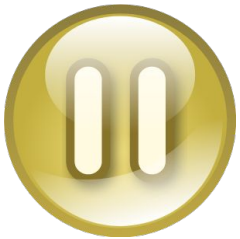


Drawbacks of the Waterfall Model

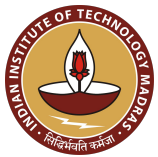
- Increase in cost, time if changes are required later on
- Clients may not know what they need!
- Designers may not know which design might be the most feasible/usable by clients
- Quite long - usually takes 6-18 months for 1 cycle

Reflection Spot

How can we address this issue of the waterfall model?



Please pause the video and written down your responses

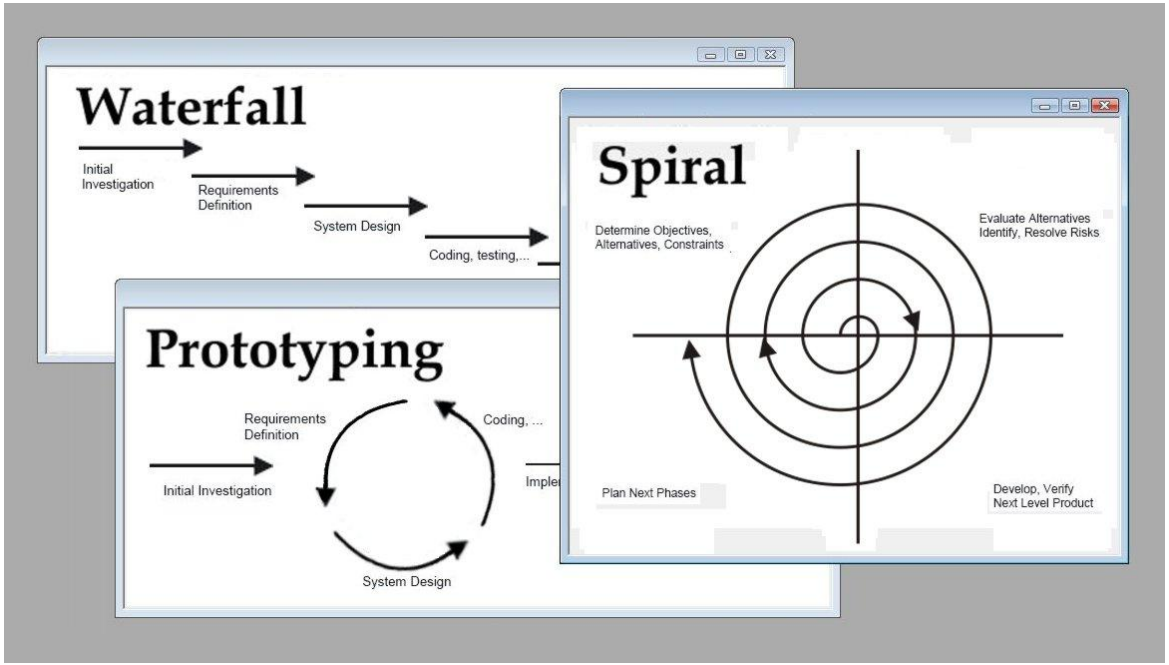


Prototype Model

- Build a working prototype before development of the actual software
- Prototype - usually not used later
- Advantages -
  - Exact form of solution and technical issues are unclear
  - Useful to get feedback from customers
- Disadvantages -
  - Increased development costs
  - Bugs can appear later in the development cycle

Spiral Model

- Incrementally build the software and get feedback, refine



Summary

- Software development lifecycle

- Different models in the plan and document perspective -
  - Waterfall
  - Prototype
  - Spiral

Software Development Models - Agile Perspective

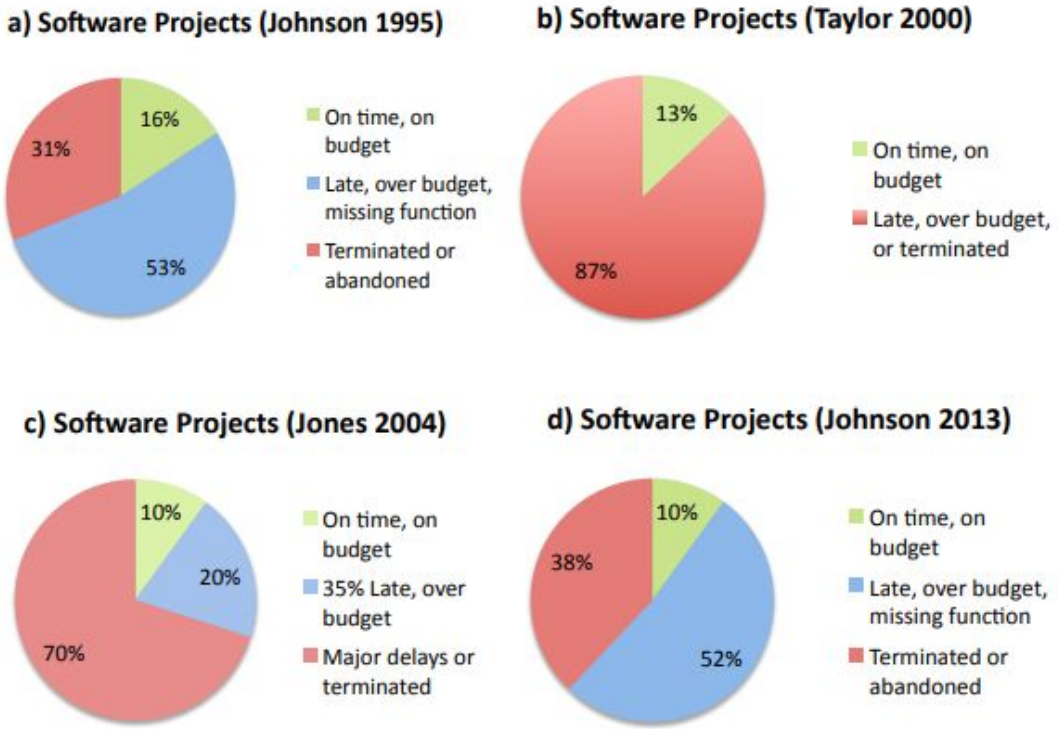
Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

Recap

- Software development lifecycle

- Different models in the plan and document perspective -
  - Waterfall
  - Prototype
  - Spiral

Study of Software Projects

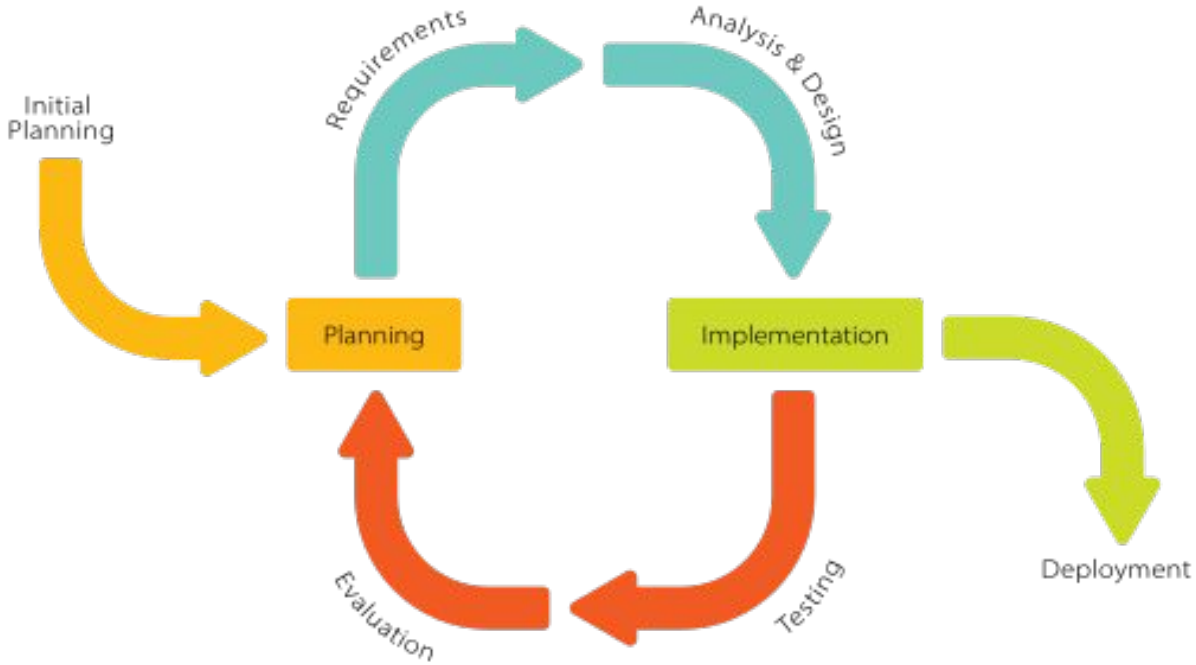


Engineering Software as a Service: An Agile Approach Using Cloud Computing (saasbook.info) - Ch1 - pg 12

Agile Manifesto

- 4 key principles -
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan

Incremental Development



Agile Approaches

- Extreme Programming (XP)
- Scrum
- Kanban

Agile Practices

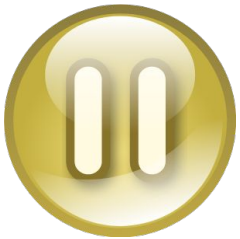
- User stories
- Sprints
- Scrum Stand-up meetings
- Test-driven development

Agile Philosophy

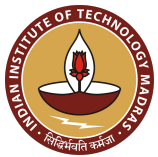
- Rather than just following approaches and practices - more of adhering to the broad philosophy
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan

Reflection Spot

Plan and Document vs Agile - When to use?



Please pause the video and written down your responses



When to use Agile/Plan and Document

	Question: A no answer suggests Agile; a yes suggests Plan and Document
1	Is specification required?
2	Are customers unavailable?
3	Is the system to be built large?
4	Is the system to be built complex (e.g., real time)?
5	Will it have a long product lifetime?
6	Are you using poor software tools?
7	Is the project team geographically distributed?
8	Is team part of a documentation-oriented culture?
9	Does the team have poor programming skills?
10	Is the system to be built subject to regulation?

I. Sommerville. Software Engineering, Ninth Edition. Addison-Wesley, 2010. ISBN 0137035152.