

# ← 1. What is this cour...



Master of SDLC

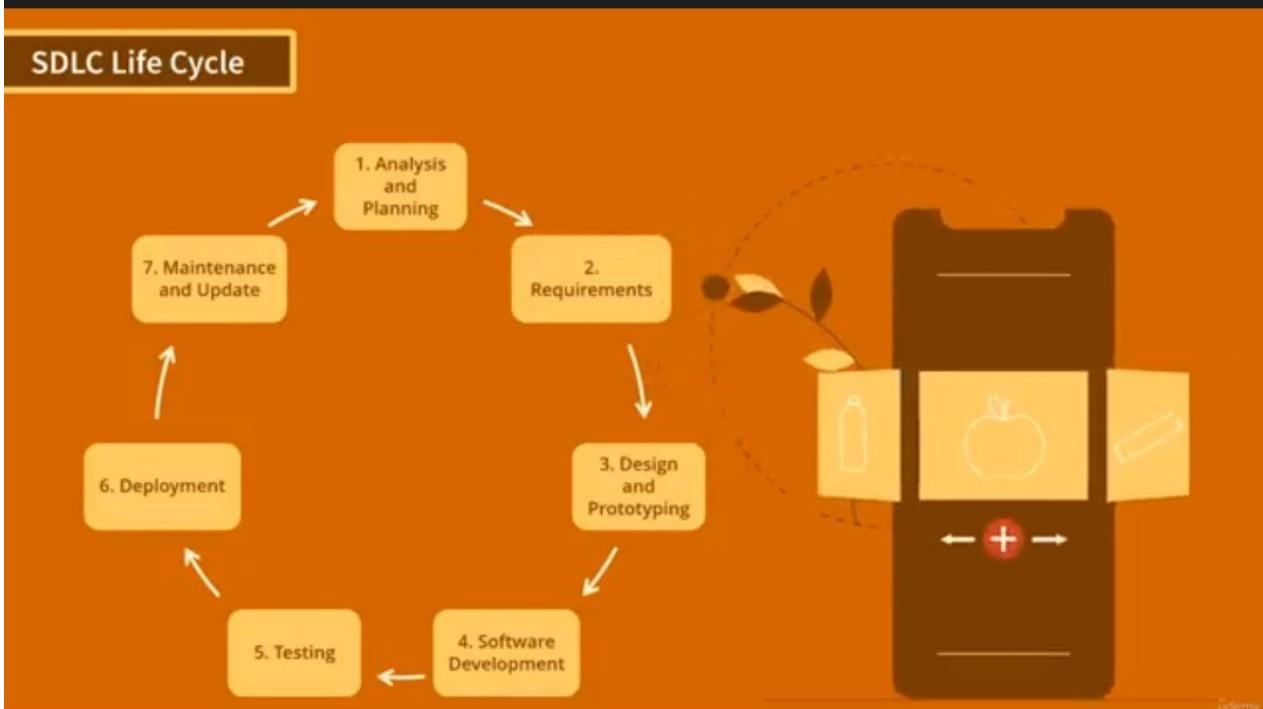
END To END Process Knowledge

Most Used and Popular Methodologies

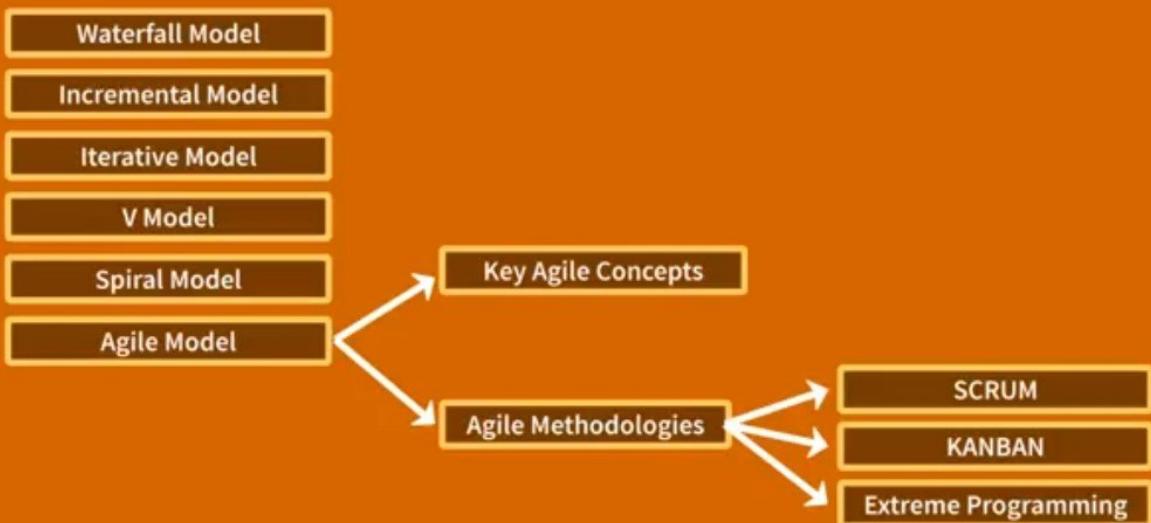
Confidence to Implement Best Practices

Setting Proper Mindset

### ← 3. How to take up t...

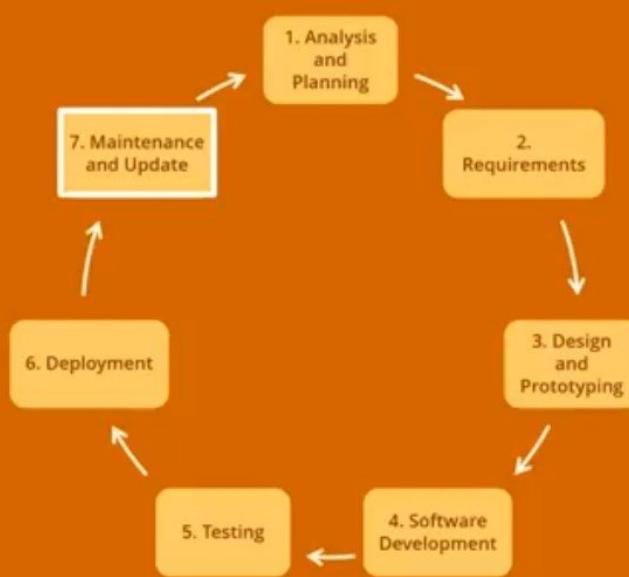


## SDLC Models



## What is SDLC?





## Why SDLC ?

**“If you fail to plan, you are planning to fail!”**  
— Benjamin Franklin

## ← 2. Why SDLC.mp4



### Benefits of SDLC

1. Clear Goals and Responsibilities

2. Common Vocabulary for Each Step

3. Project Tracking and Visibility

4. Improved Product Quality

5. Increase Deployment Speed

6. Decrease Project Risk

7. Decrease Project mgmt Overhead

8. Reduce Unnecessary Cost



HR Team



EmployeePortal

- IT Team
- Finance Team
- Project Team
- Security Team

## Analysis and Planning

- Alignment
- Resource Allocation
- Scheduling
- Cost Estimation

## All Impacted Teams



## ← 4. First Phase Analysis...



### Scope of Work (SOW)



## ← 5. Sample SDLC Pr...



### Sample Project Schedule and Timelines

DailyNeeds EmployeePortal High Level Timelines	FY-22												
	Q4		Q1			Q2			Q3			Q4	
	Nov	Dec	Jan	Feb	Mar	April	May	June	July	Aug	Sept	Oct	Nov
Analysis and Planning													
Requirement Gathering													
Design and Prototyping													
Software Development													
QA Testing													
UAT Testing													
Deployment													
Maintainance and Updates													

Hypercare Period for 1 or 2 week → HOTFIX Deployment

Issues Reported

## Requirements

How requirements are gathered?



## ← 6. Second Phase...



### Requirements

Who will be part of this meetings/workshops?



Business  
Analyst

Project  
manager

Technical  
Architect



Business  
Team



Lead Developer  
Project Team

## ← 6. Second Phase...



### Identification of UseCases

1. How existing system is working?
2. Who is going to use this system?
3. What is input and output needed?
4. Do we need to integrate with third party APIS or Tools?
5. How to handle Security and privacy ?
6. What are limitations?



Software Requirement  
Specification Document  
(SRS Document)

## ← 6. Second Phase...



### Functional Requirements :

What the software application or product should do  
Requirements stated and demanded by users



DailyNeeds

1. Login form for Employee

2. Form to fill new Joinee details

### Non-Functional Requirements :

Quality Constraints.  
How the software system should behave.



DailyNeeds

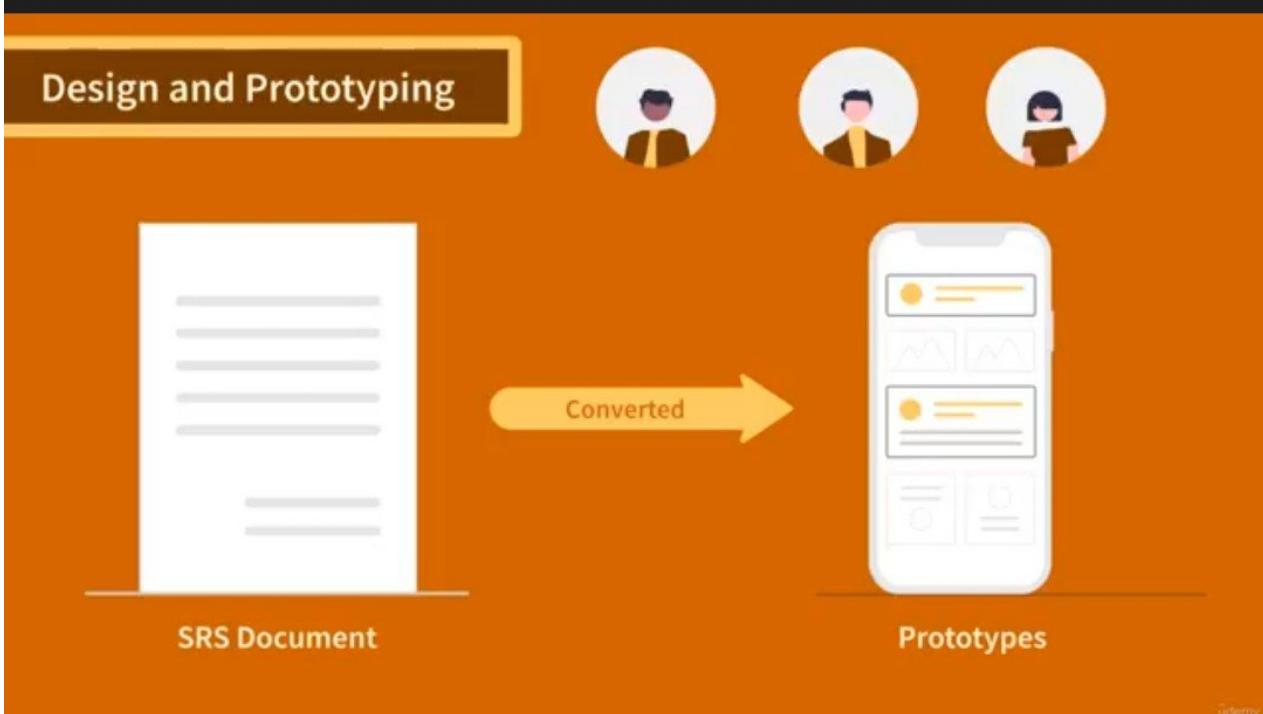
Performance/Reponse time  
Accessibility  
Scalability  
Availability  
Maintainability  
Extensibility

1. Every page should load in 3 sec
2. Should be accessible over all broswers in web and mobile
3. Should be available 99% of the time

### Technical Requirements :

Technical stack ,Database

## ← 8. Third Phase Des...



## ← 8. Third Phase Des...



### How to Design Application End to End

#### High Level Design

Entire System

Architecture

Database Design

Data Flow

Services, Platforms, Modules

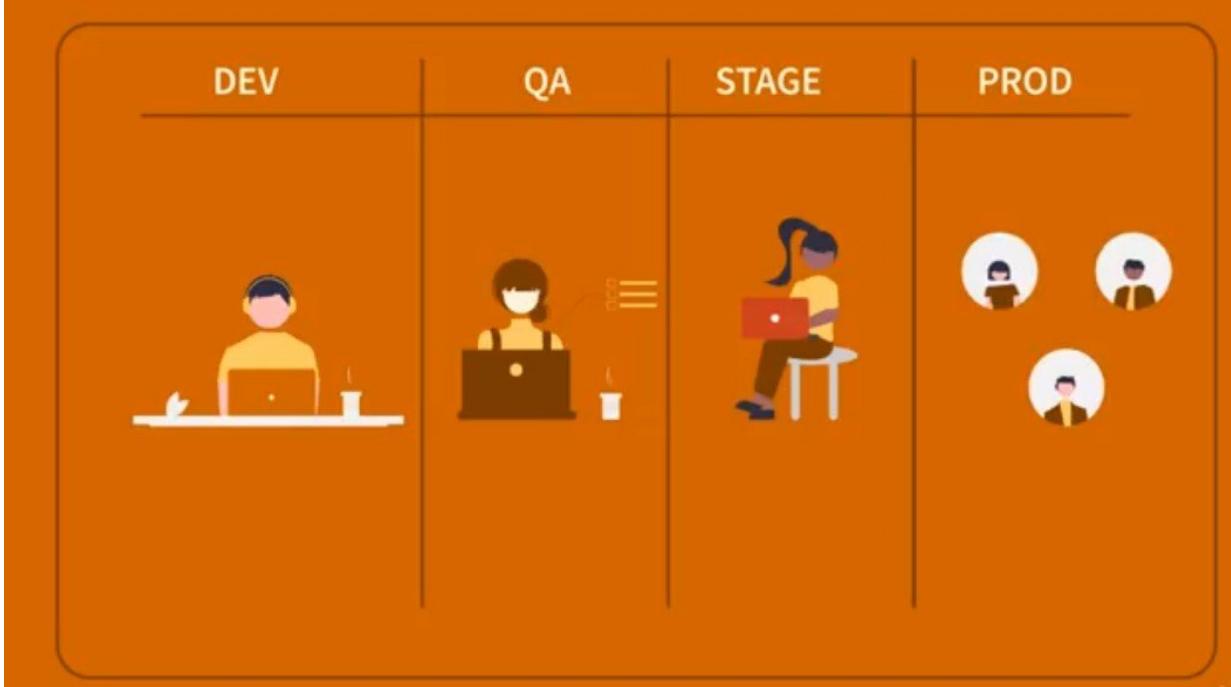
Made by Architect

#### Low Level Design

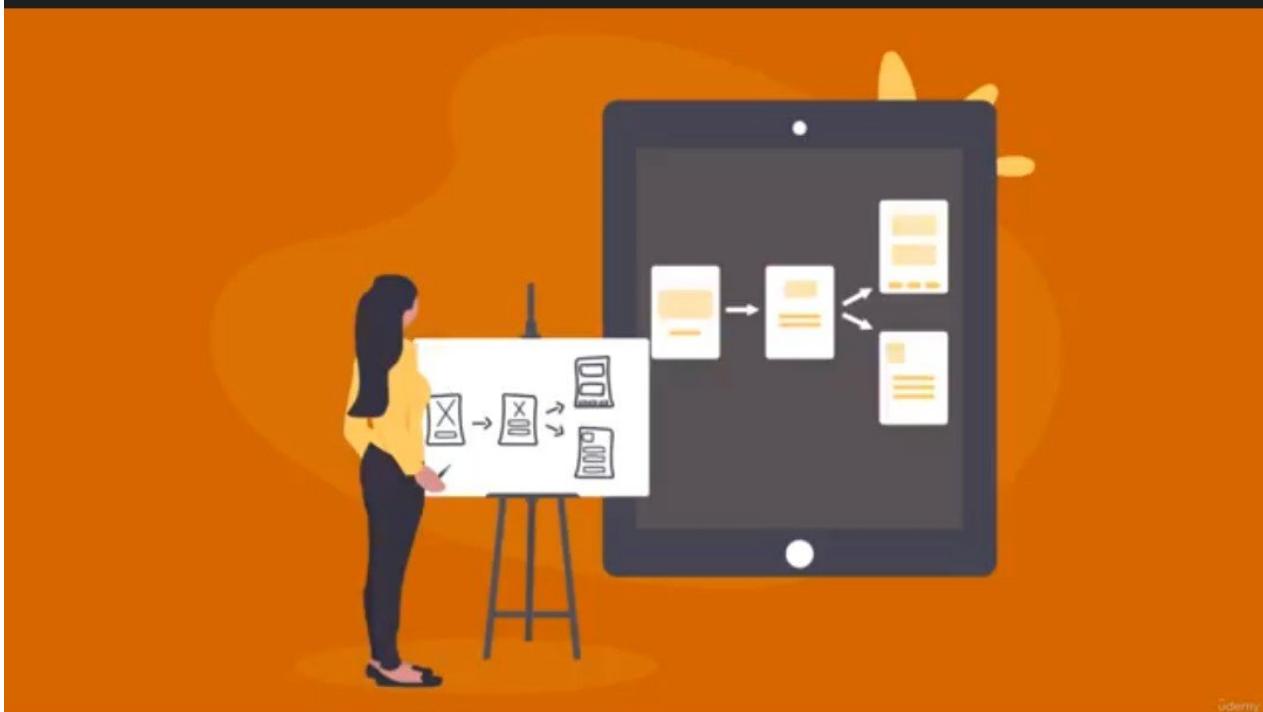
Each Module Design  
and Specifications

Made by Lead Developer

## ← 8. Third Phase Des...



## ← 8. Third Phase Des...



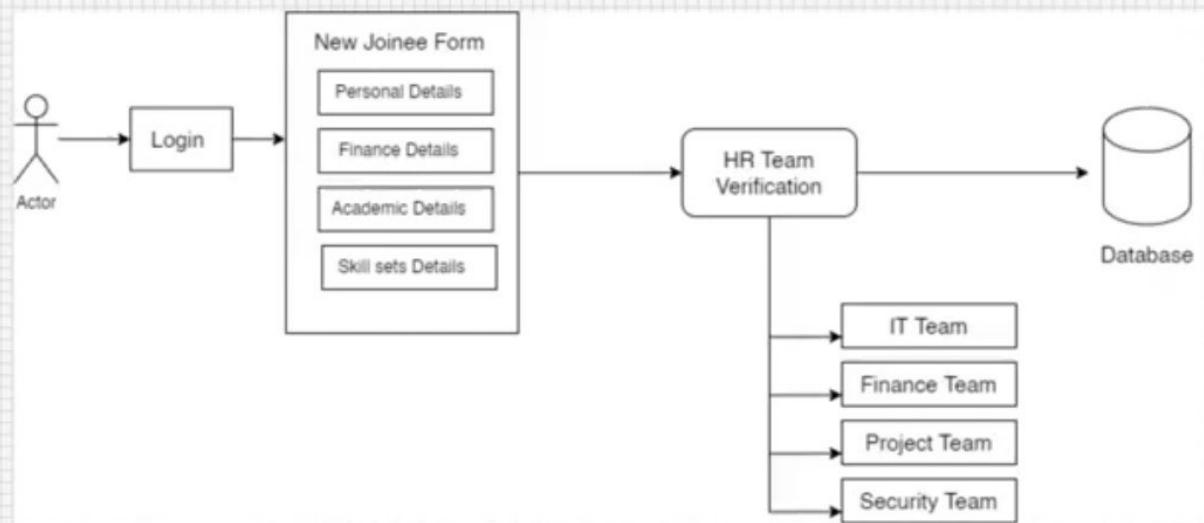
High Level Diagram

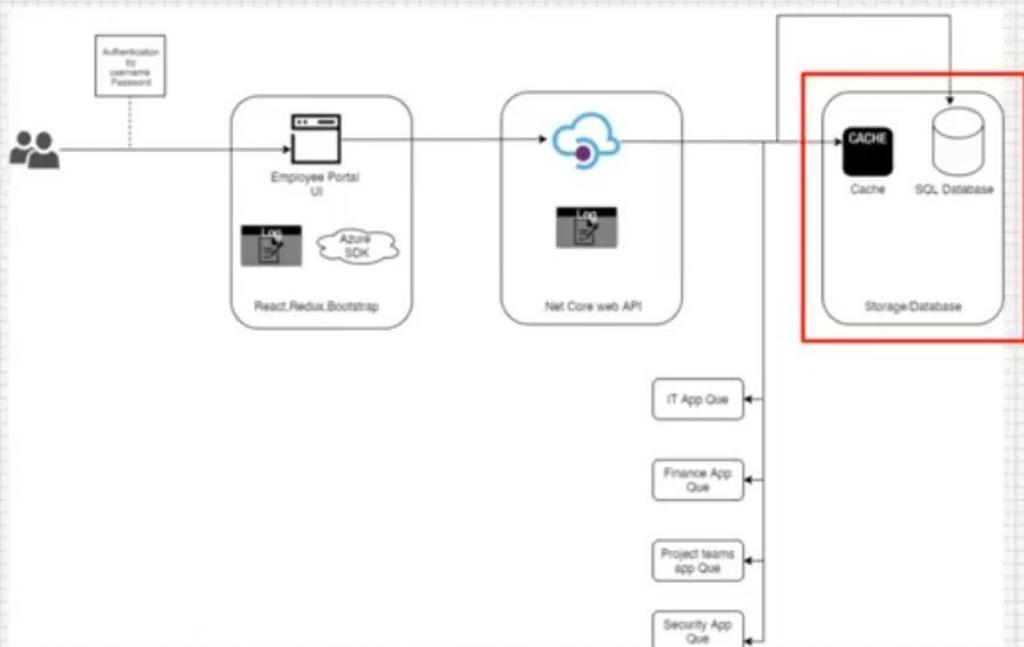
Low Level Diagram

Data Flow Diagram

Database Schema Diagram

Prototype of Login Page

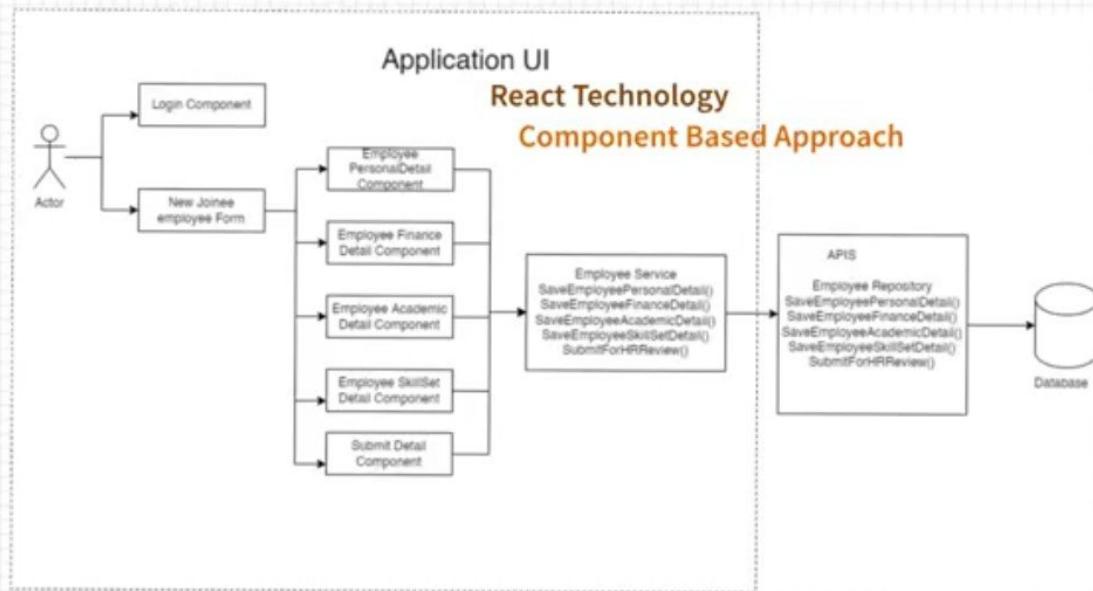




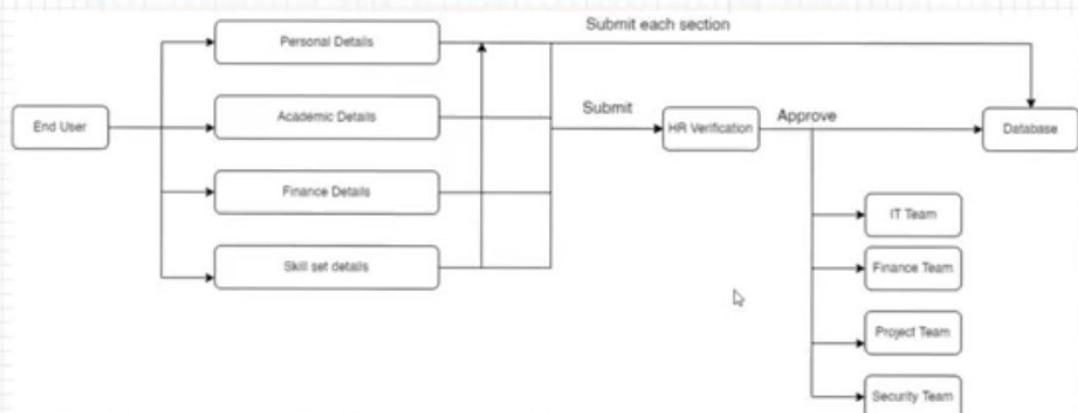
## ← 10. Sample Low Le...



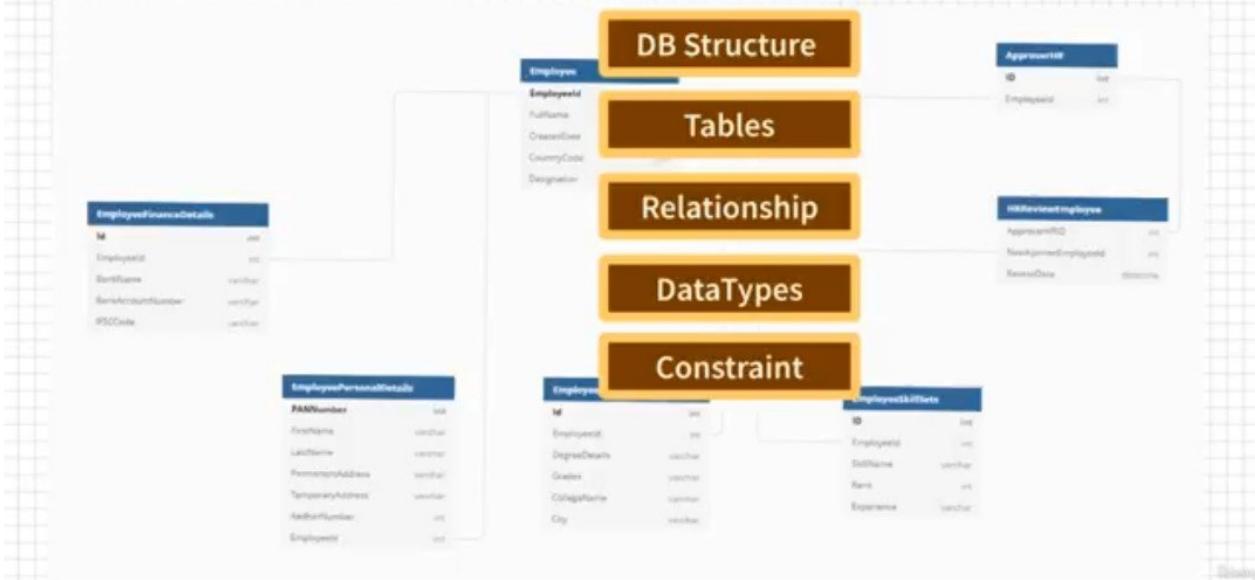
### Sample Low Level Diagram



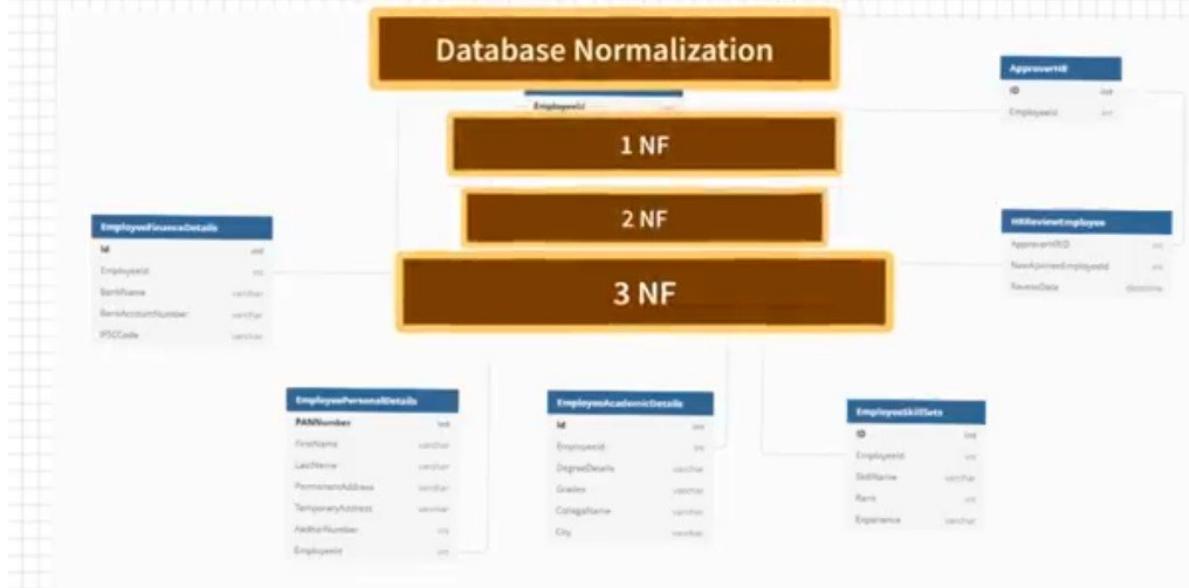
## Sample Data Flow Diagram(DFD)



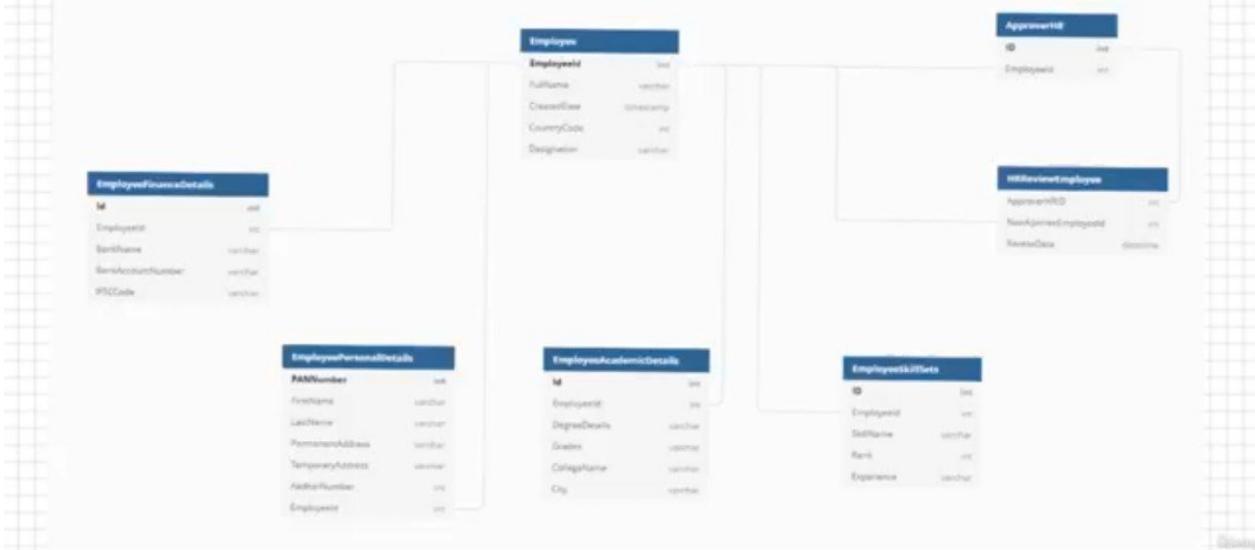
# Sample Database Schema Diagram



# Sample Database Schema Diagram



# Sample Database Schema Diagram



# Sample Prototype/Mockup

## Login Page



Adobe XD  
Invision Studio  
Figma  
Draw.io

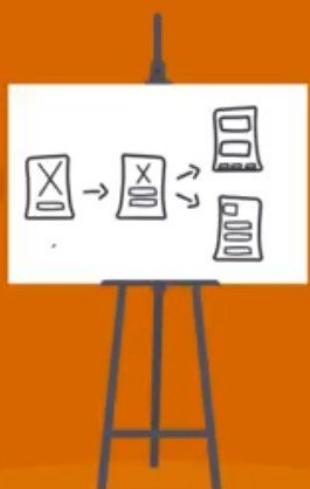
# Sample Prototype/Mockup

## Login Page

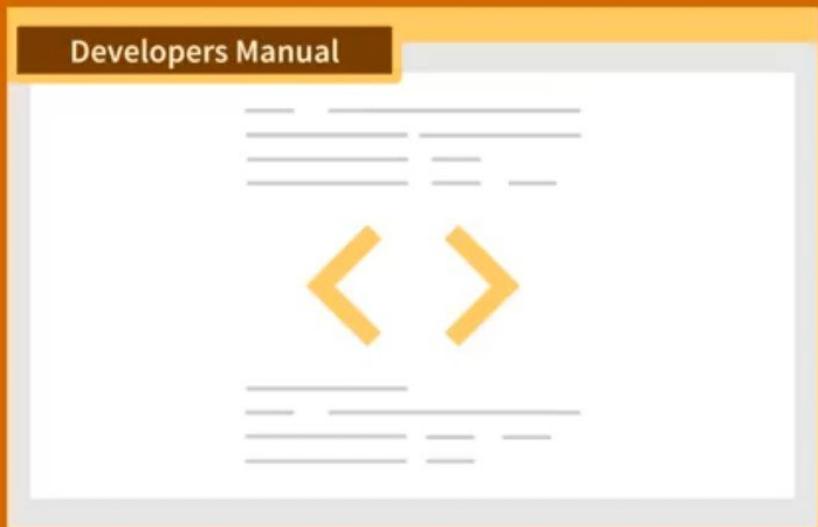


Wireframe  
Diagram

## Software Development



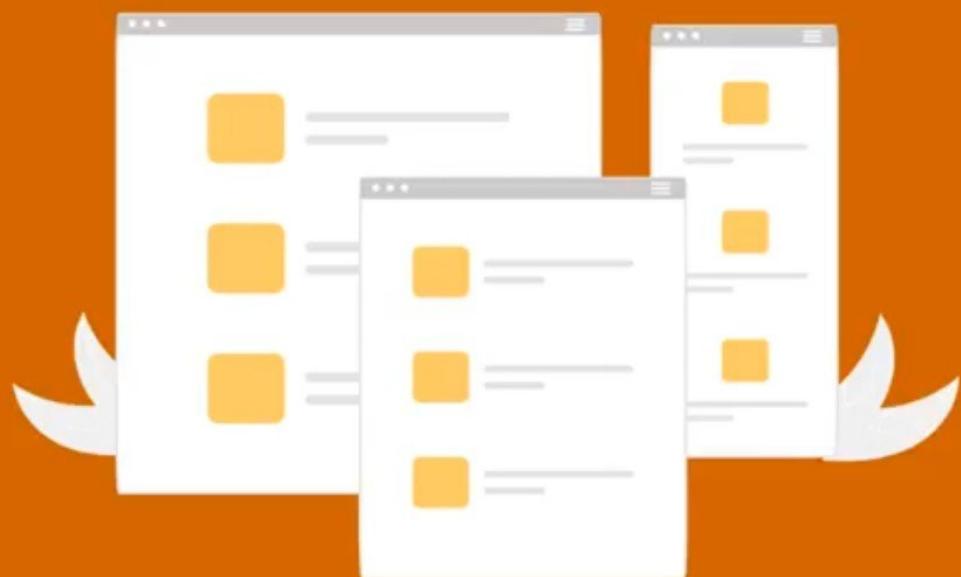
## Software Development



Udemy

## Software Development





Udemy

Testing

QA Testing

## Testing

Functional Testing

Integration Testing

Performance Testing

Load Testing

Penetration Testing

## ← 15. Fifth Phase Tes...



Testing

Testing Sign-Offs

.....

.....

.....

.....

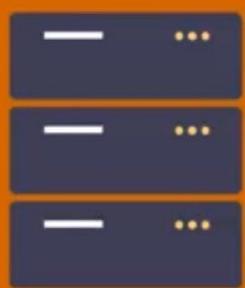
.....

.....

Udemy

## Deployment

### Production Deployment



Udemy

# ← 16. Sixth Phase De...



## Deployment



DevOps Team / Change  
Management Team

## Request For Change





## Maintenance and Updates



## Logging Application Issues

Remedy, ServiceNow

Service Level Agreement(SLA)

Critical - SLA 3 Hours

High- SLA 1 Day

Medium- SLA 5 Days

Low- SLA 2 Weeks



Remedy

## Maintenance and Updates

Application Updates

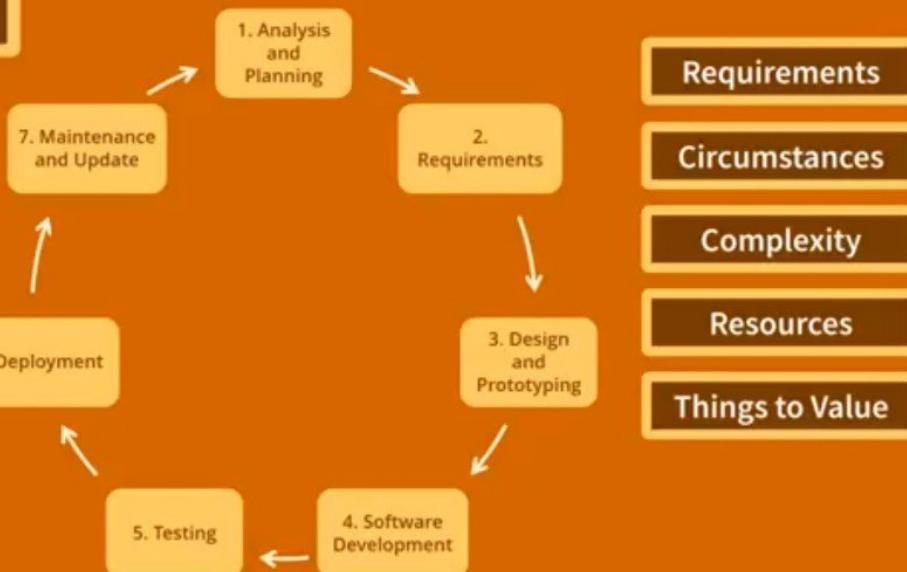
Server Updates

OS updates

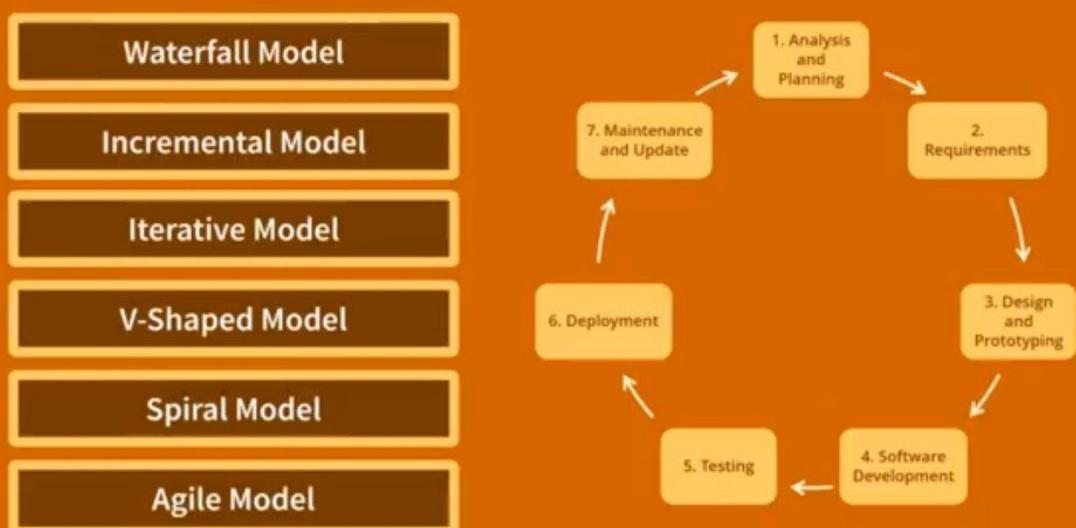
Application Framework Upgrades



## SDLC Models



## SDLC Models



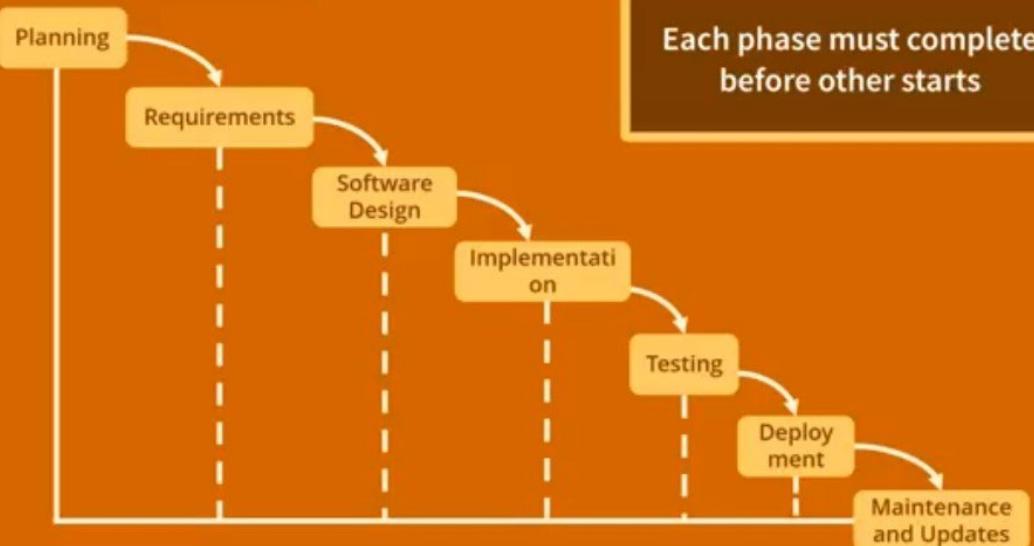
## Waterfall Model

First,Oldest ,Traditional

Linear-Sequential Model



## Waterfall Model



## When & Where Waterfall Model

Requirements Clear and Fixed

Application is not Complicated

Short Duration Projects

Enough Resources

Mission Critical Projects

## Pros and Cons of Waterfall Model

### Pros

- Simple and Easy
- Manageable
- Phases Completed One at a Time
- Scope is Fixed
- Clear Phases/Stages/Goals
- Visibility on Progress
- Well Documentation

### Cons

- No Working Software in Early Stages
- Tracking Progress Within Phase is Tough
- Under Utilization of Resources
- Not fit for Project where Requirements Changes
- Poor Model for Long term Projects
- No Scope Adjustment

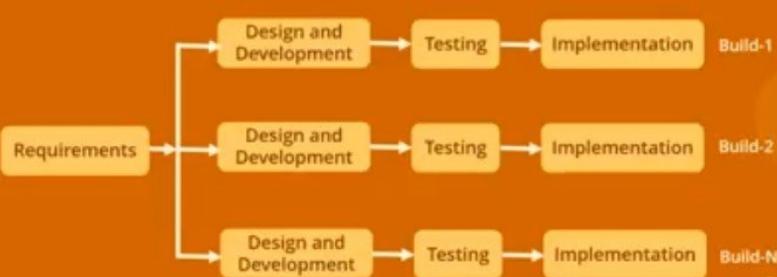
# ← 1. What is Incremental...



## Problems with Waterfall Model



## Incremental Model



## ← 2. When and Where Incremental Model?



### When and Where Incremental Model?

Requirements are Well Understood

Limited Resources

Scope Changes to Some Extent

## Pros & Cons of Incremental Model?

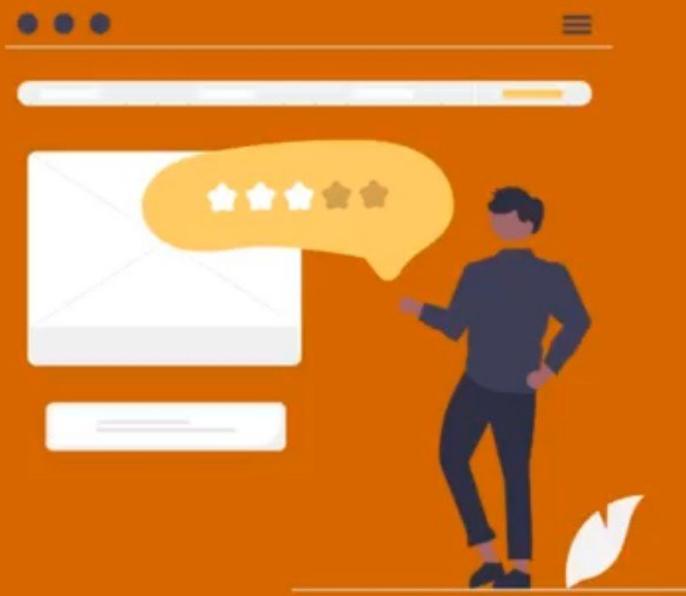
### Pros

- Save Time
- Slight Scope Adjustment
- Errors Identified Early
- Feedback

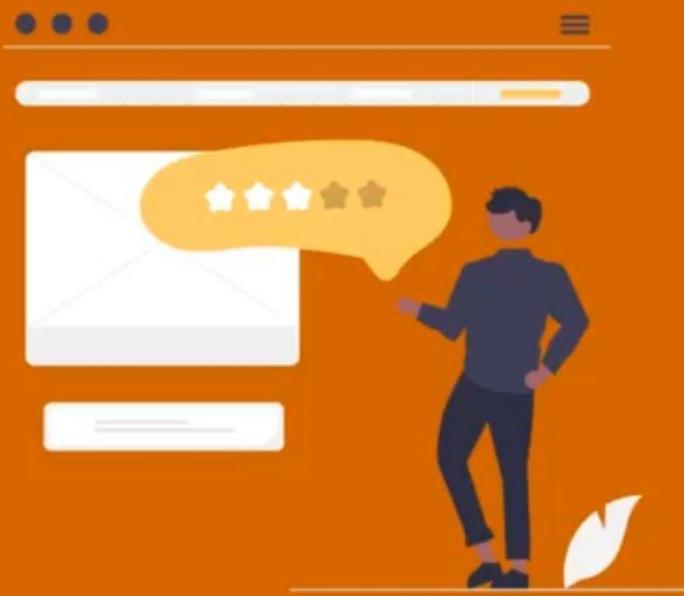
### Cons

- Late Customer Review
- Rework if Scope Changes

## Problems :



## Problems :



## Iterative Model

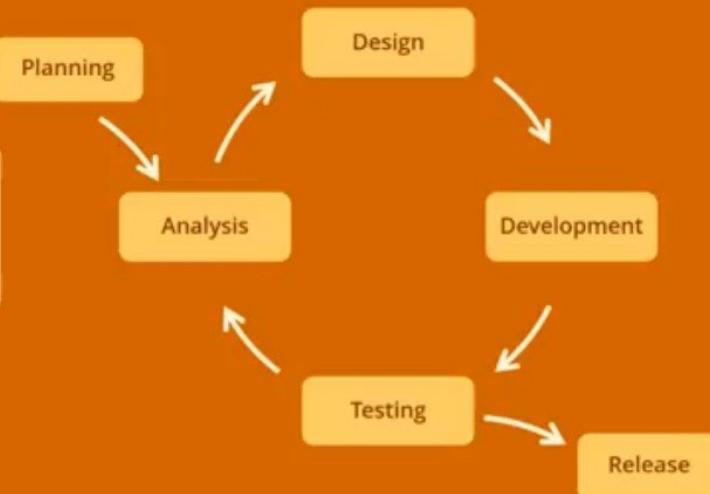
Iterations :Refining something repeatedly until satisfied

# ← 1. What is Iterative...



## Iterative Model

Incremental &  
Iterative Model



## When & Where Iterative Model

Startup Projects or Product based Companies

When all Projects Requirements are not Clear

When we don't have many Resources Available

### ← 3. Pros and Cons o...



#### Pros & Cons of Iterative Model

##### Pros

Working Software in Early Stages

Early Feedback

Flexible Scope and Requirements

Assist in Early Decision Making

##### Cons

Impact of Requirement Change

Impact of Technology Changes

More Management Attention Needed

System Design Issues will come Later

Cost more

Unclear Schedule

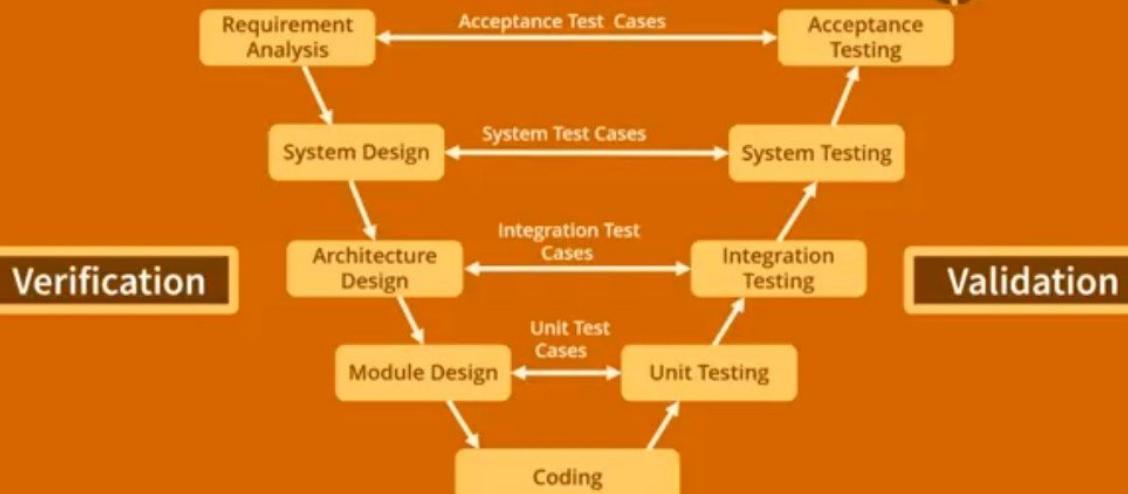
## Problem :



## Problem :



## V-Model



## When & Where V-Model

Small and Medium Size Projects

Medical Industry Projects

Compliance & Accuracy Demanding Projects

### ← 3. Pros and Cons o...



#### Pros & Cons of V-Model

##### Pros

Phases Completed One at a Time

Work Well for Smaller Projects

Simple and Easy to Use

Easy to Manage

##### Cons

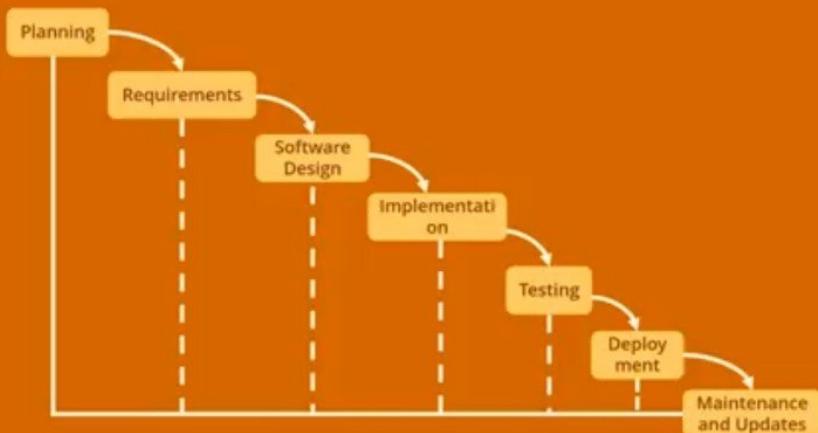
Not Good for Complex Projects

Poor Model for Long Ongoing Projects

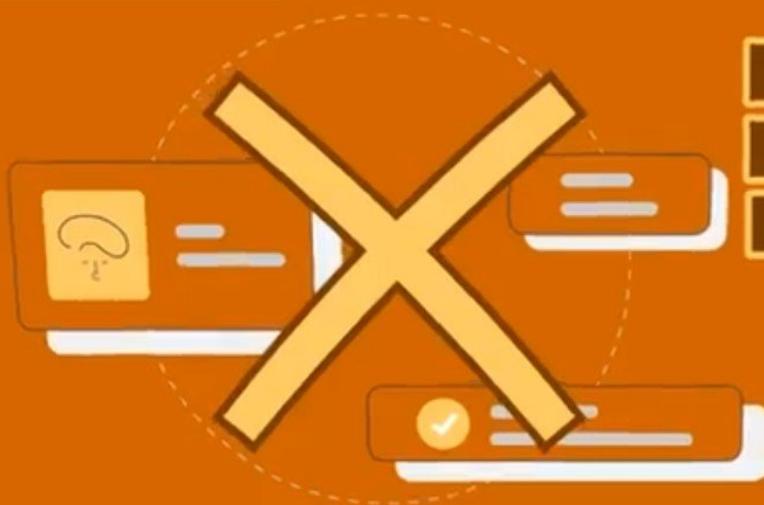
No Scope for Requirement Change

Difficult to Change Functionality at Later stage

## Problems :



## Problems :

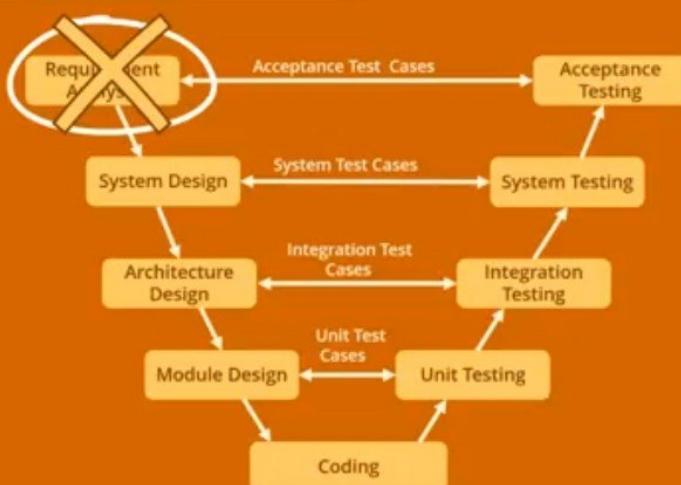


Technology Issue

Compliance Issue

Causing Data Leaks

## Problems :

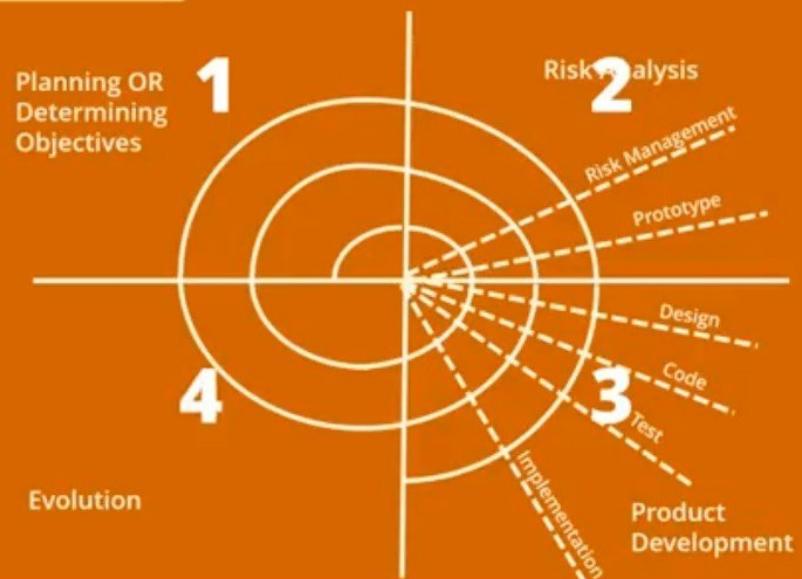


## Spiral Model

Avoid Risk /Highlight Risk at Early Stages

Flexible Scope and Requirements

## Spiral Model



## Spiral Model



## Spiral Model



## When & Where Spiral Model

High Risk Projects

Risk & cost Evaluation is Priority

Flexible Requirement & Scope

IT Companies

## Pros & Cons of Spiral Model

### Pros

- Risk Assessment
- Most Useful for Large & High Risk Projects
- Flexible Requirement and Scope
- Users Sees System Early
- Easy Cost Estimation
- Always Space for Customer Feedback

### Cons

- Expert People Required
- Not Useful for Small Projects
- Challenging to predict Duration of Project
- Bad Image to Customer
- Risk of not meeting Schedule & Budget

## Traditional Vs Agile OR Problems with Previous Models:

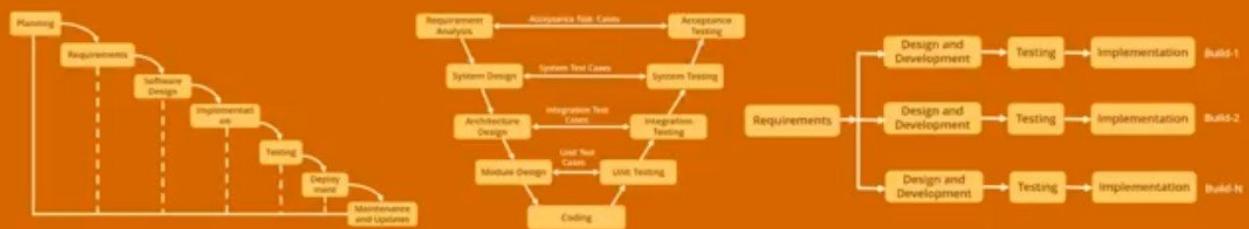
### Planning & Requirements



## Traditional Vs Agile OR Problems with Previous Models:

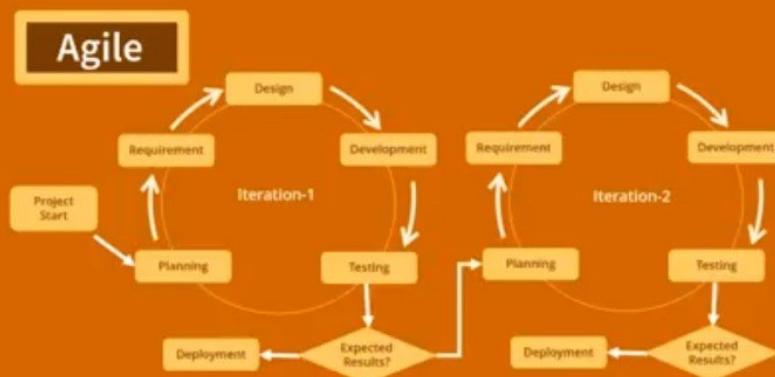
Predictive Models

Adaptive Models

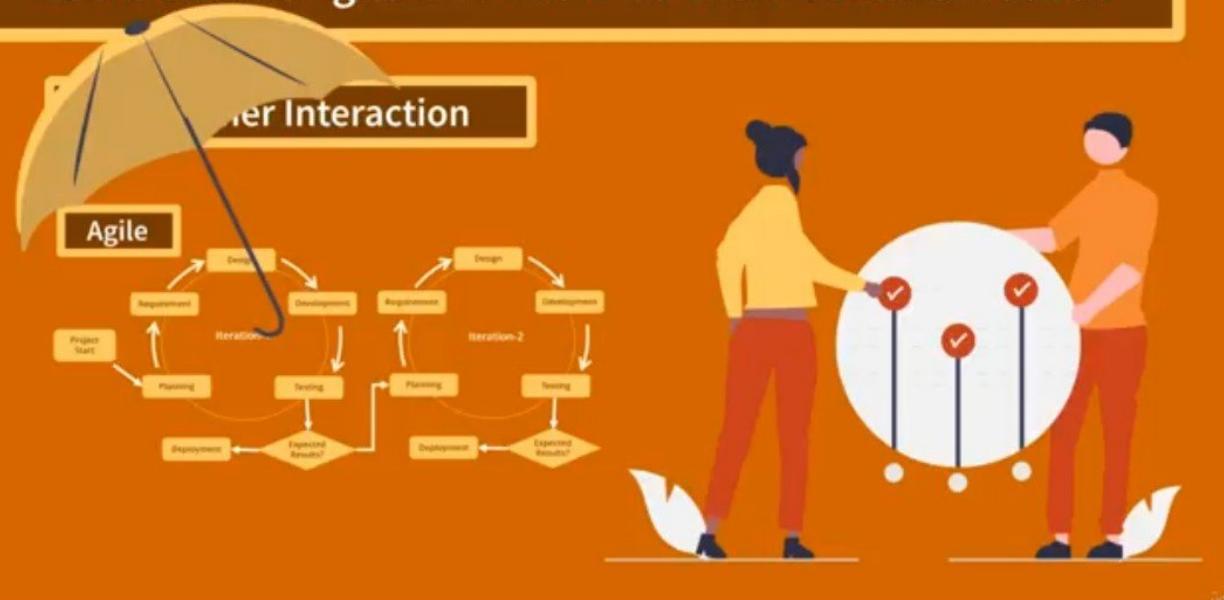


## Traditional Vs Agile OR Problems with Previous Models:

### Adaptive Models



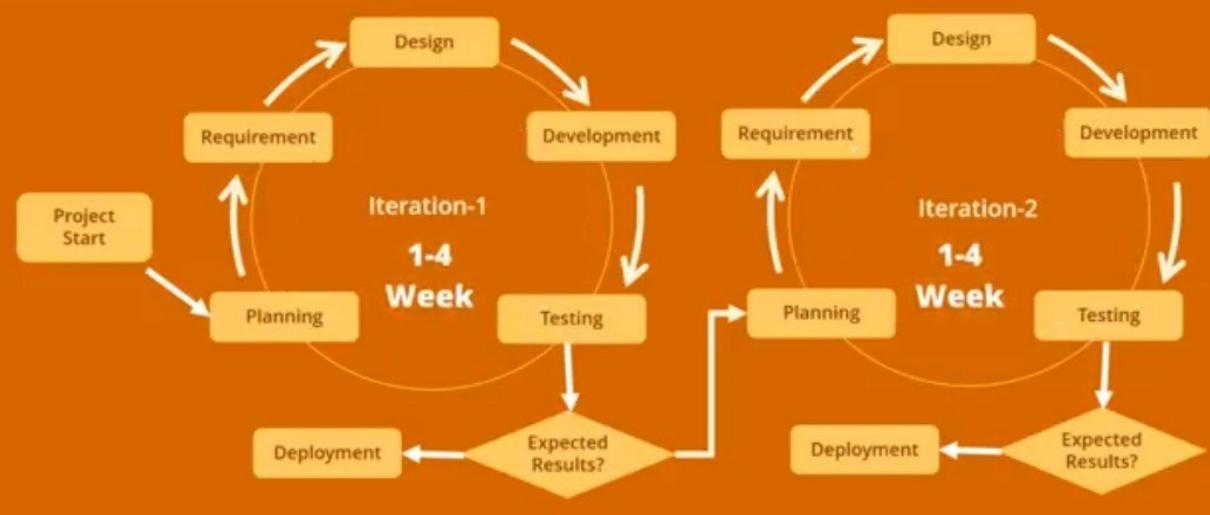
## Traditional Vs Agile OR Problems with Previous Models:



## What is Agile Model



## What is Agile Model



## What is Agile Model

### Agile Methodology

SCRUM

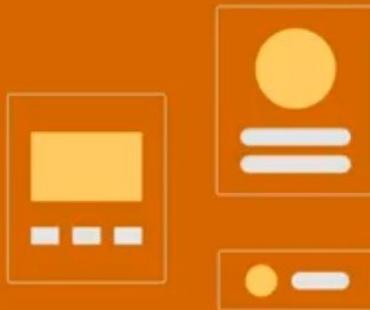
Crystal

Dynamic Software Development  
Method

Feature Driven Development(FDD)

Lean Software Development

Extreme Programming



## When & Where Agile Model

Project where Scope is not fixed and Requirements Changing

Project with tolerance to Bugs or Defects

Customer who wants to make Early Decision/ Start-up Companies

Small Projects with Small Team

## Pros & Cons of Agile Model

### Pros

Customer Collaborations

Fast Project Delivery

Flexible Scope and Requirements

Minimized Risk

Early Customer Feedback

Less Documentations

### Cons

Tough to Estimate Cost and Time for Big Project

Collaboration is Difficult to maintain

Poor Resource Planning

Fragmented Output

No Finite End

## ← 5. Agile Manifesto....



Before Agile

2001



## Agile Manifesto



## ← 5. Agile Manifesto....



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:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- 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.

Individuals and interactions over processes and tools

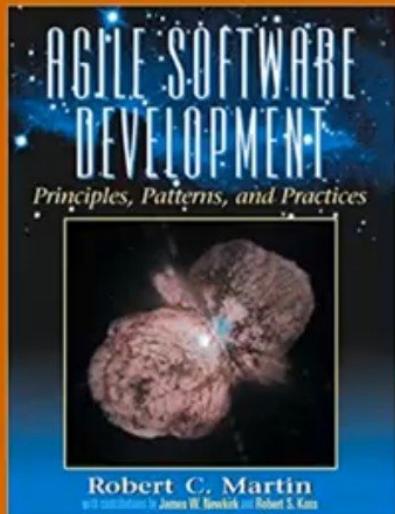


UpNext

## ← 5. Agile Manifesto....



Individuals and interactions over processes and tools



Good Process will not save the project if your team doesn't have strong player.

Bad process will even make strongest player ineffective . Even the group of strong player will fail if they don't work as team.

Team with average people who communicate well are more likely to succeed than strong player who fail to interact as a team.

Working Software over comprehensive documentation



### Customer Collaboration over contract negotiation



©demyl

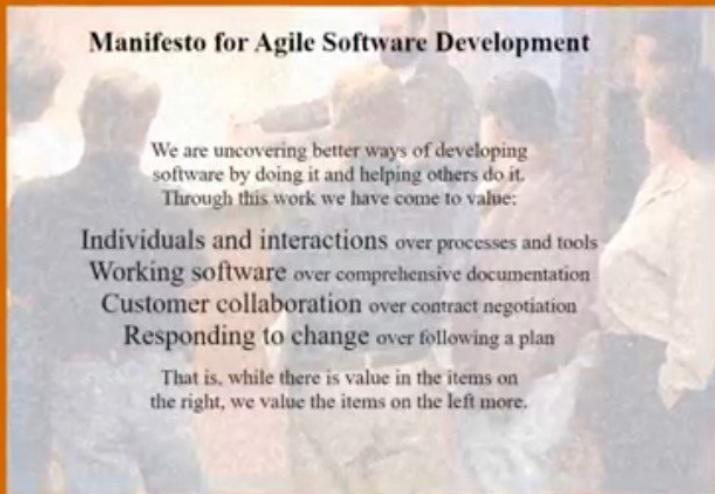
## Customer Collaboration over contract negotiation



## Responding to change over following the plan



## Twelve Principles of Agile Softwares



## Lean Start-up



Business-Hypothesis-Driven Experimentation

Iterative Product Releases

Validated Learning

# ← 1. Minimum Viable...



## Minimum Viable Product(MVP)

Least version of product, which can be given to customer for his/her use

## User Story



## ← 2. User Story.mp4



### Role-Action-Benefit Template

As a role... I want to action... So That benefit...

As a role (Who want to accomplish something)

I want to Action ( What they want to accomplish)

So that benefit (Why they want to accomplish that thing)

For Example :  
As a bank customer, I want to  
withdraw money from an ATM ,  
So that I am not constraint by  
opening hours or lines at the banks

## INVEST Formula

I : Independent

N : Negotiable

V : Valuable

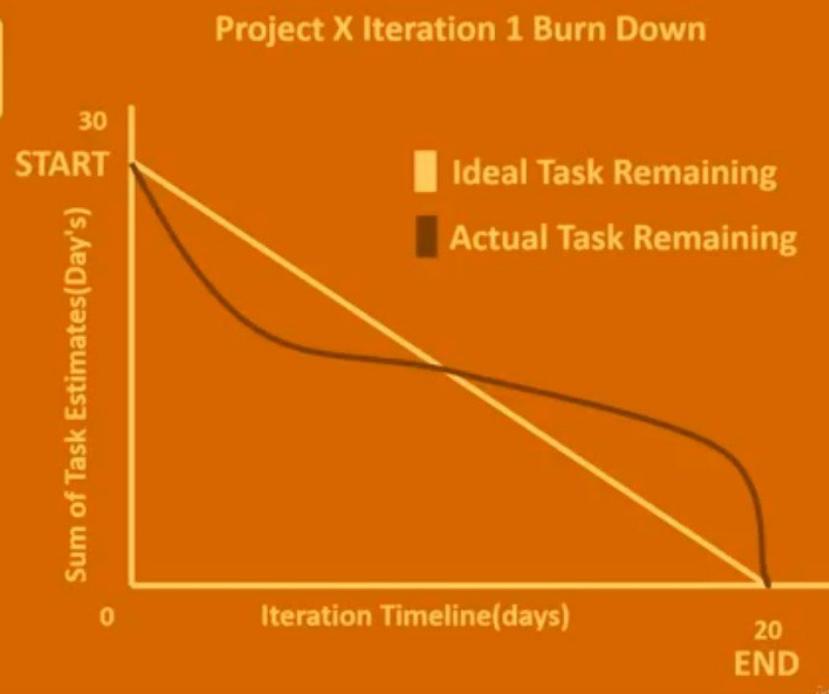
E : Estimable

T : Testable

For Example :  
As a bank customer, I want to  
withdraw money from an ATM ,  
So that I am not constraint by  
opening hours or lines at the banks

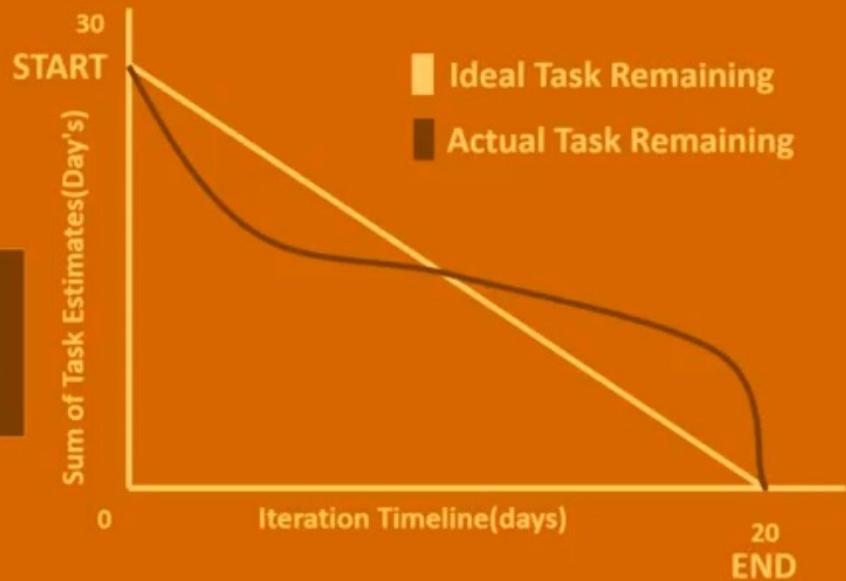
User Acceptance Criteria

## Burndown Chart



### Project X Iteration 1 Burn Down

Goal :  
1. Display Progress  
2. Estimate of Completion



9:05 PM

86.5K/s 4G VoLTE 60%



Scrum

Kanban

XP

9:07 PM

12.7K/s 4G VoLTE 60%

TELUSKO

# WHAT IS SCRUM?

A Better Way Of Building Products



[scrum.org](https://scrum.org)

Navin Reddy

9:07 PM

0.19K/s 4G VoLTE 60%

TELUSKO

**Lightweight**  
**Simple to Understand**  
**Difficult to Master**



Navin Reddy

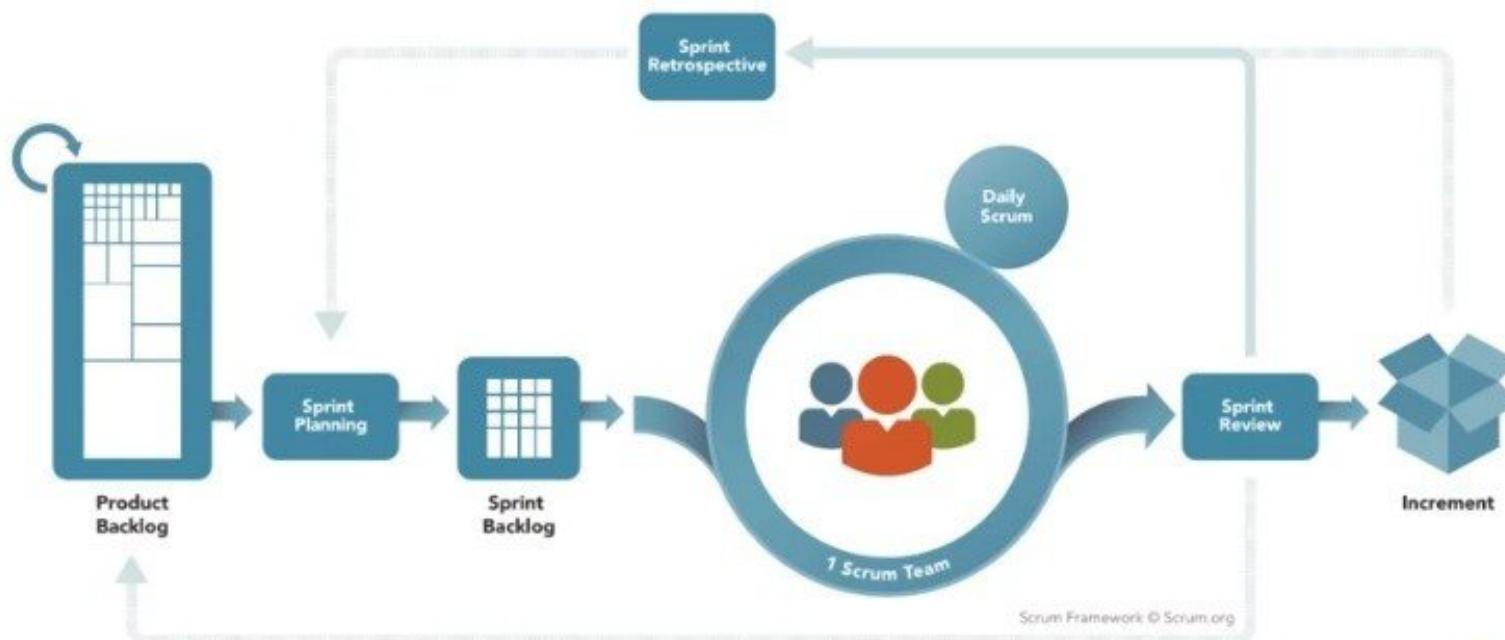
# Scrum is Everywhere

90% - Estimated Agile Teams use Scrum

12M+ Estimated Using Scrum Daily  
Practiced Everywhere

## One Scrum Guide

## SCRUM FRAMEWORK



9:11 PM

79.2K/s 4G VoLTE 60%

TELUSKO

# Sprint

1 or 2 or 4 Weeks



Navin Reddy

## A Typical Sprint Retrospective Model

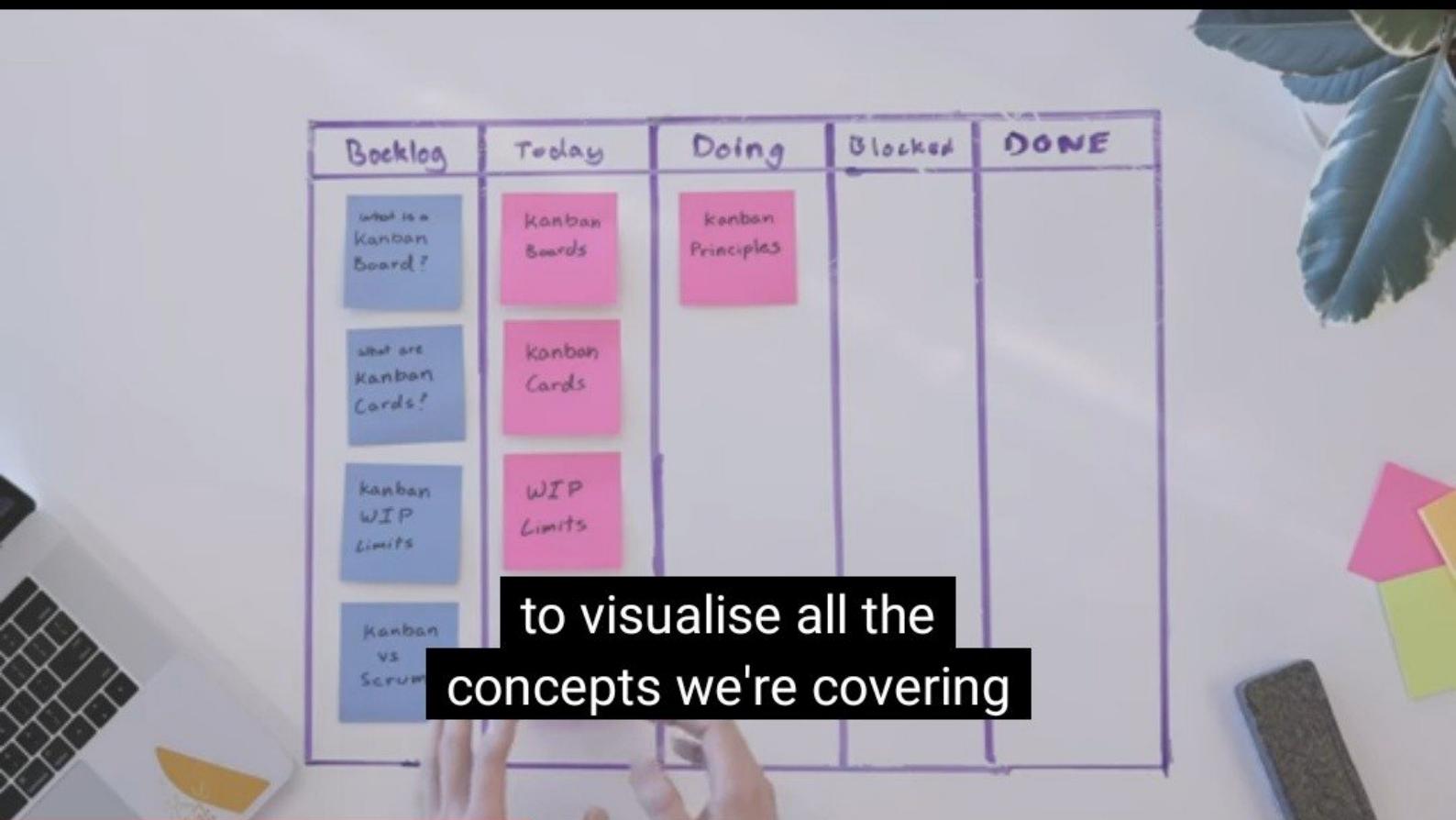
What worked well?

What could be improved?

What will we commit to  
doing in the next Sprint?

*Scrum Team members  
make actionable  
commitments*





## What is Kanban? - Agile Coach (2019) ^

413,194 views



6.1K



Share



Download



Save

## Extreme Programming (XP)



Udemy

## ← 3. Extreme Progra...



### Extreme Programming (XP)



### ← 3. Extreme Progra...



#### Extreme Programming (XP)



## Extreme Programming (XP)

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software and higher quality of life for the development team .XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

-Agile Alliance.

## 1. Customer Team Member

