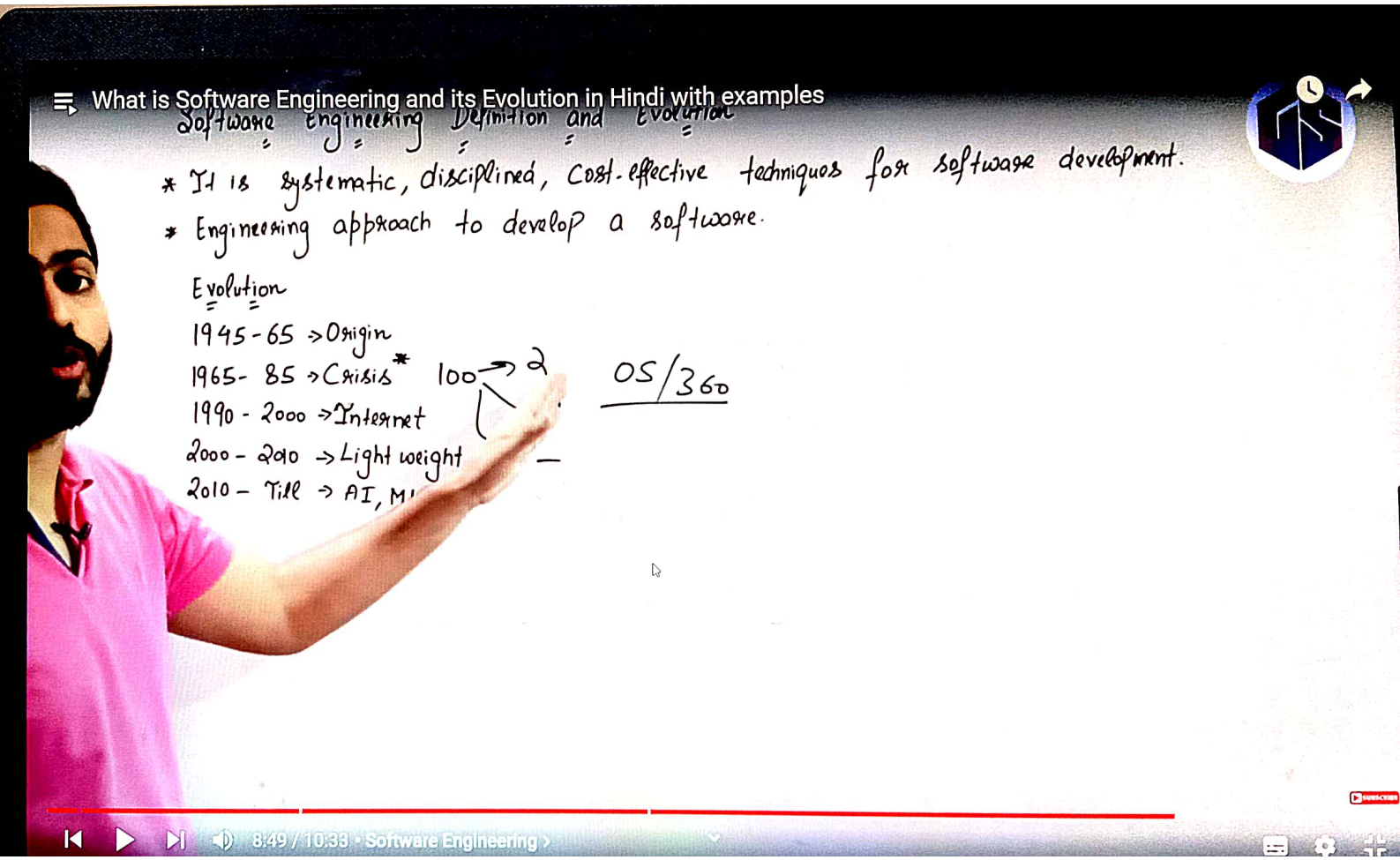
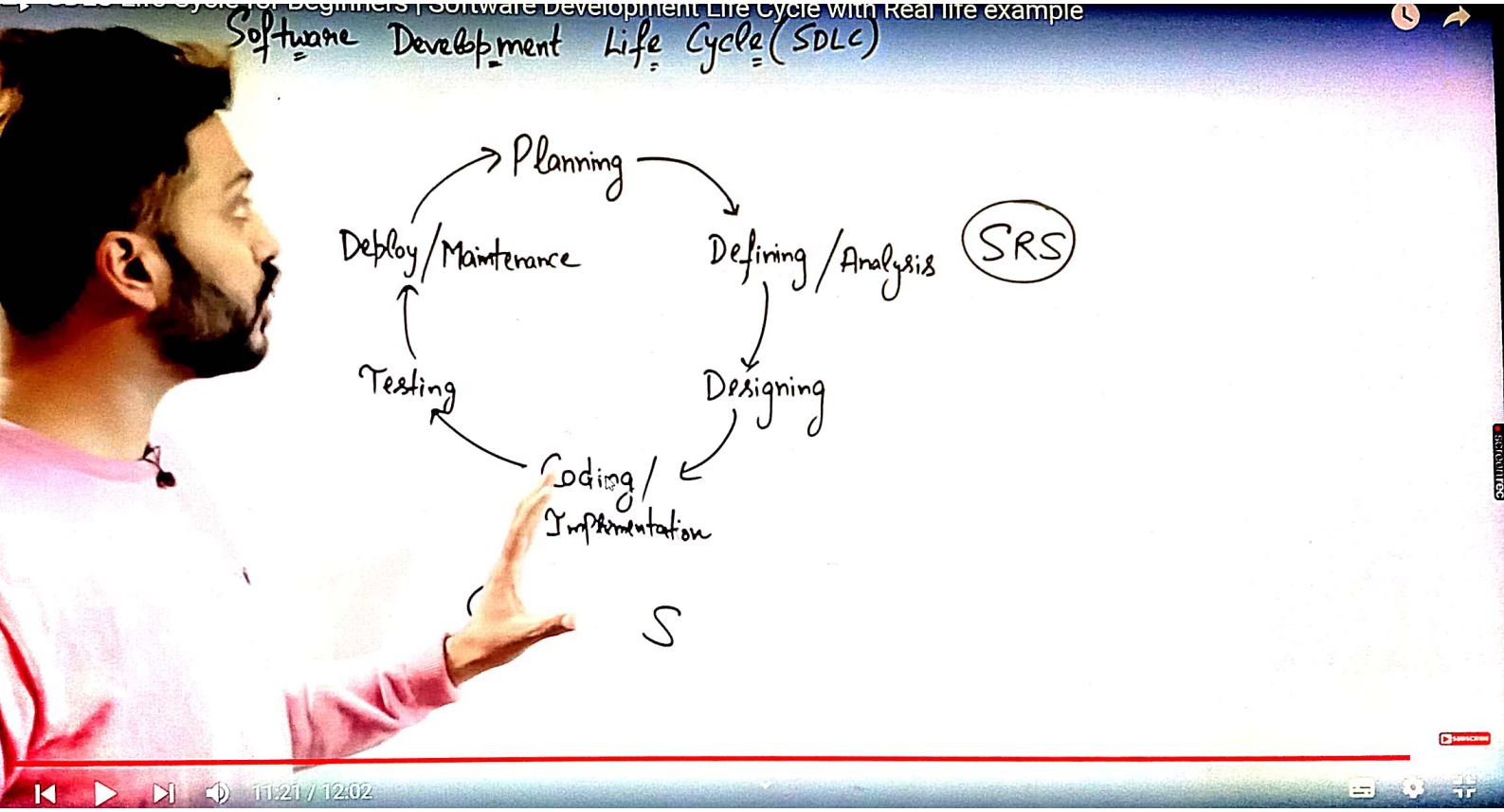
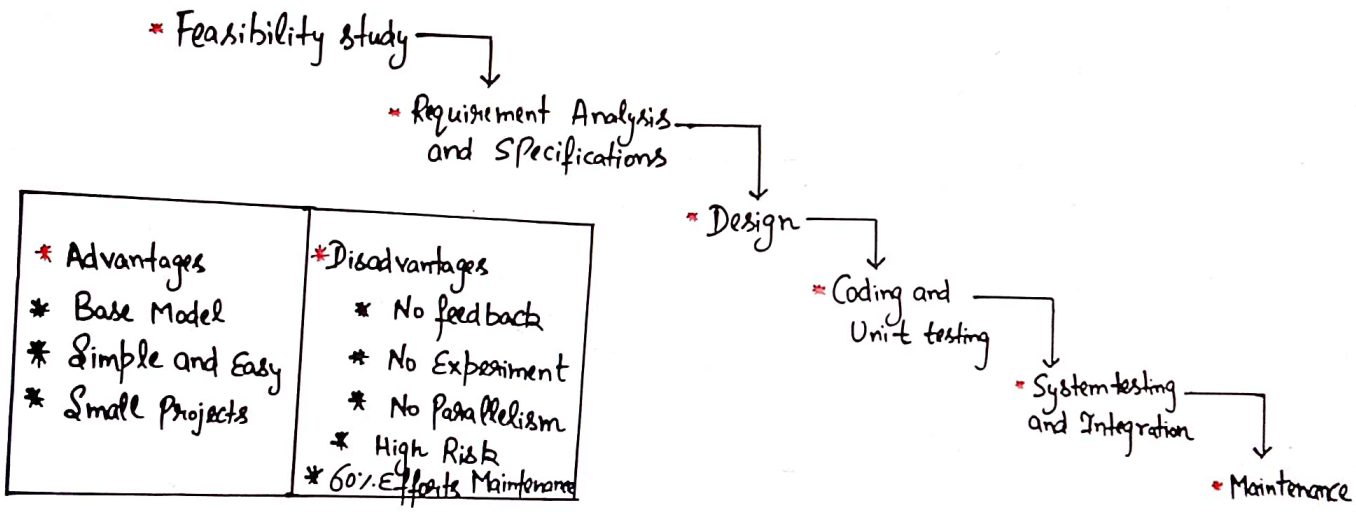

What is Software Engineering and its Evolution in Hindi with examples
Software Engineering Definition and Evolution
* It is systematic, disciplined, cost-effective techniques for software development.
* Engineering approach to develop a software.
Evolution
1945-65 -> Origin
1965- 85 -> Crisis*
1990 - 2000 -> Internet
2000 - 2010 -> Light weight
2010 - Till -> AI, M
100 -> 2
OS/360
8:49 / 10:33 • Software Engineering >


Software Development Life Cycle (SDLC)
Planning
Defining / Analysis
SRS
Designing
Coding / Implementation
Testing
Deploy / Maintenance
11:21 / 12:02

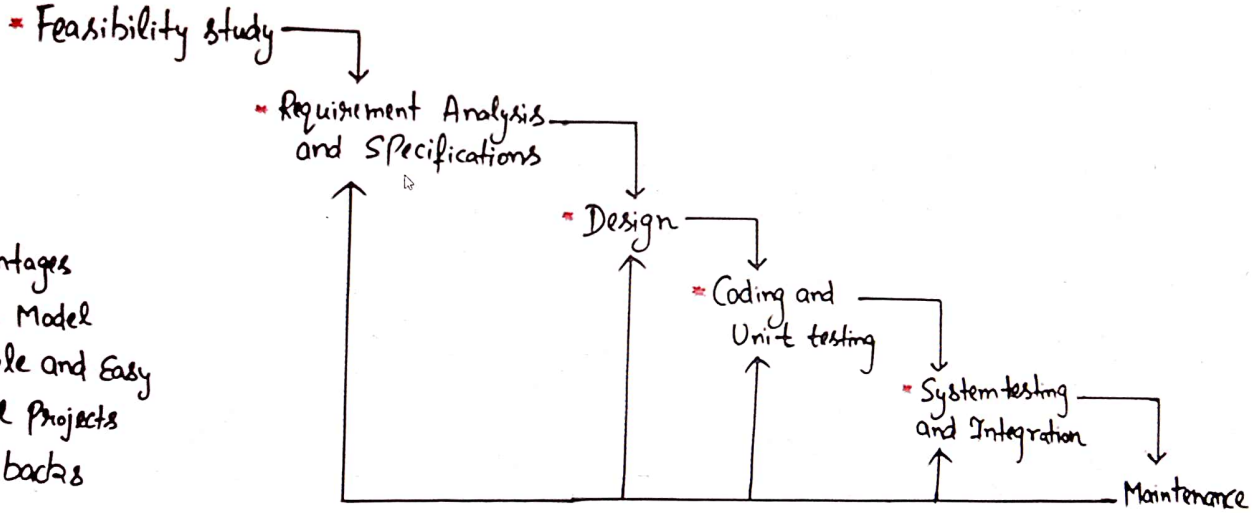
Classic Waterfall model in Software Engineering

"Classical Waterfall Model"



Iterative Waterfall Model with example in Hindi

Iterative Waterfall Model



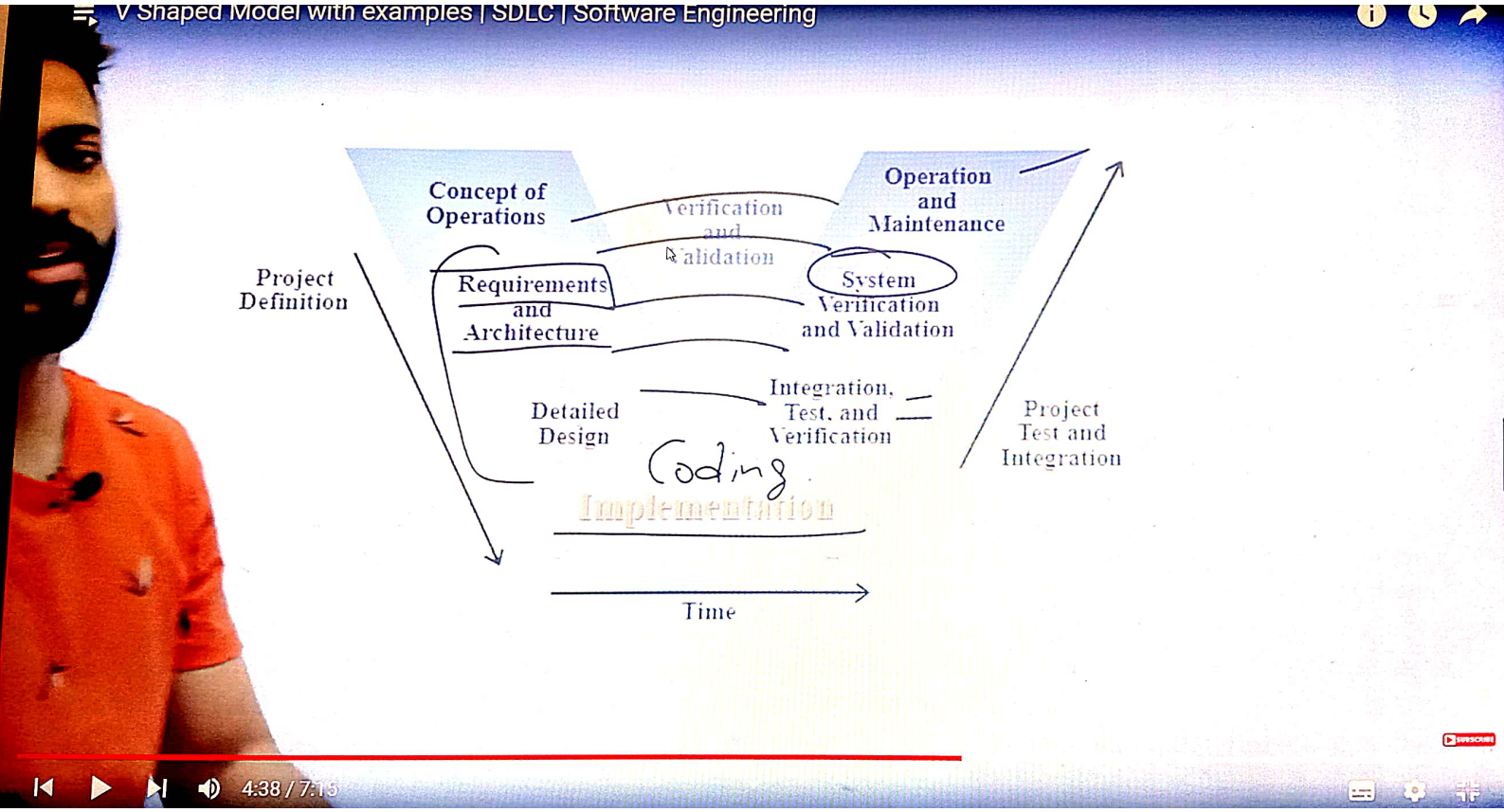
- Advantages
- Base Model
- Simple and Easy
- small Projects
- Feedbacks

- disadvantages
- No phase overlapping
- No Intermidiate Delivery
- Rigid (No changes)
- Less Customer Interaction

1:10 / 4:36

V-Shaped Model

- Also known as Verification & Validation Model
- Extension of Waterfall model.
- Testing is associated with every phase of lifecycle.
- Verification Phase(Requirement analysis, System design, Architecture design, Module design)
- Validation Phase(Unit testing, Integration, System, Acceptance Testing)


V Shaped Model with examples | SDLC | Software Engineering
Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Coding
Implementation
Time
4:38 / 7:15

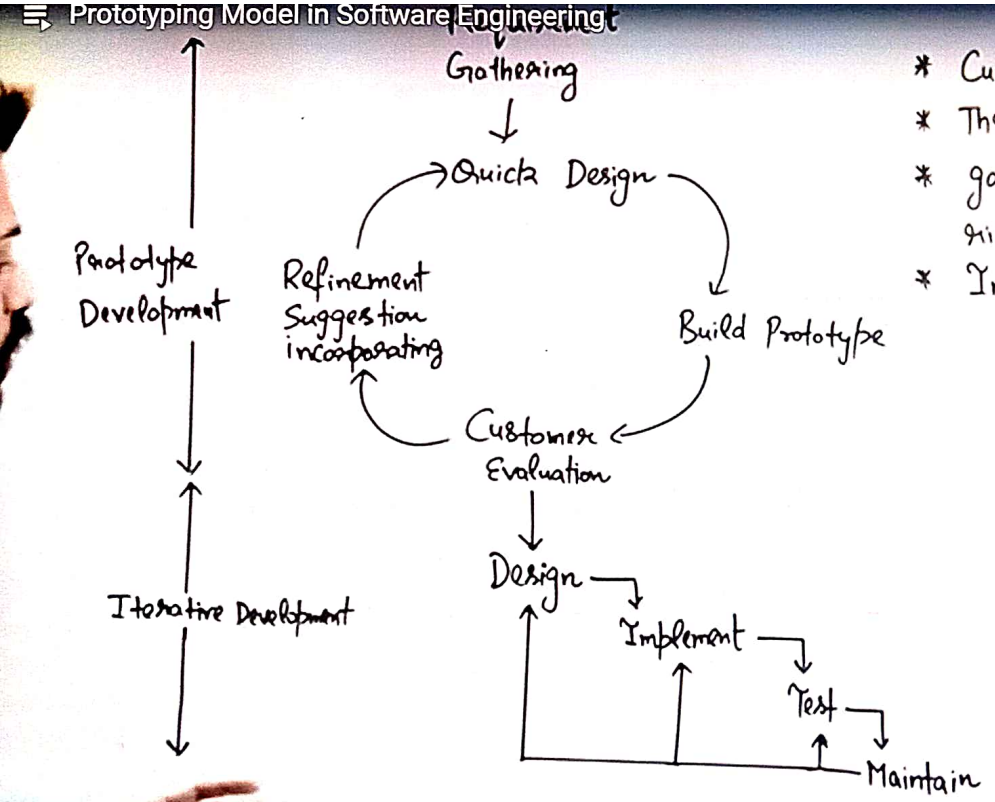
Advantages

- Time saving
- Good understanding of project in the beginning.
- Every component must be testable.
- Progress can be tracked easily.
- Proactive defect tracking

Disadvantages

- No feedback so less scope of changes
- Risk analysis not done
- Not good for big or object-oriented projects

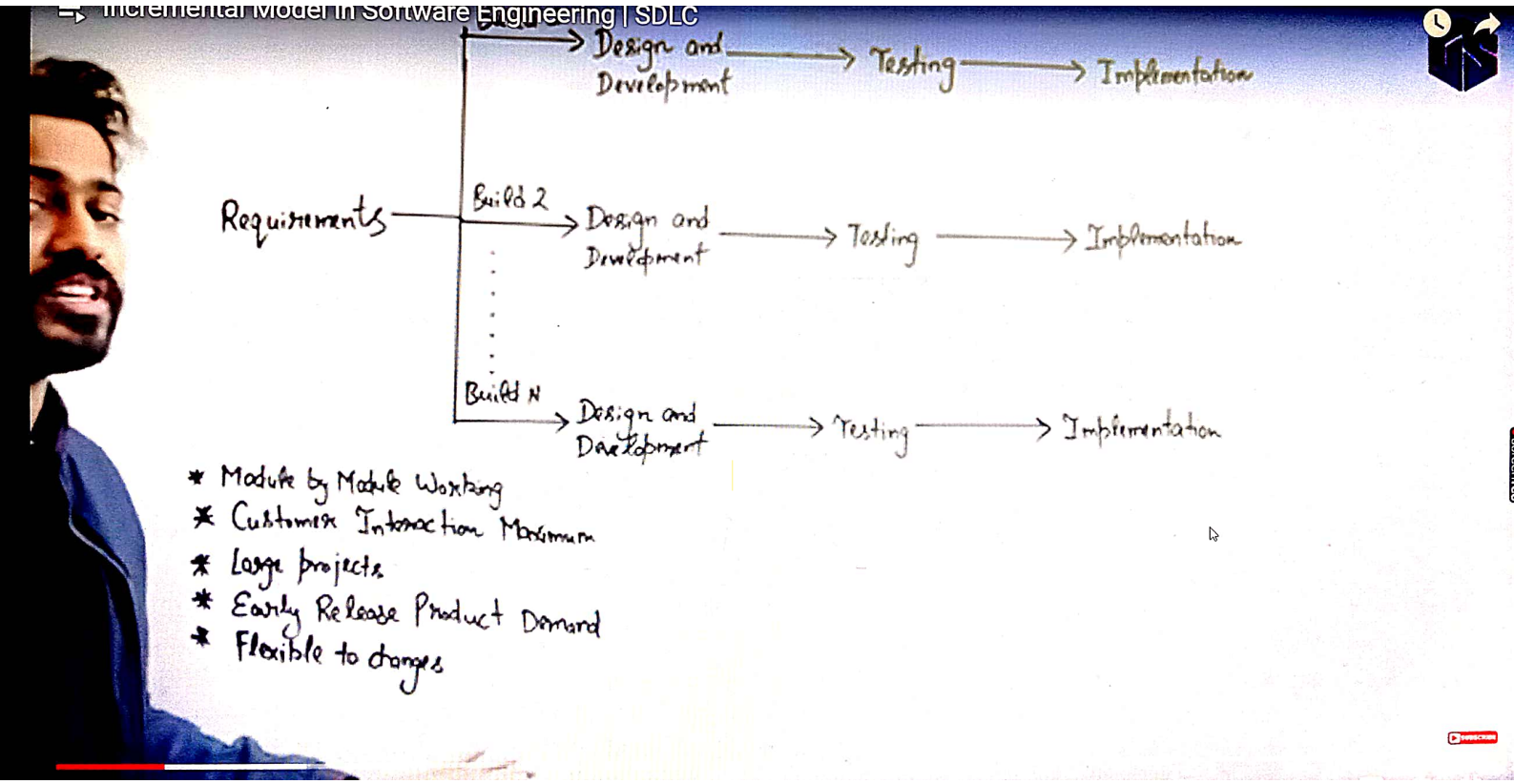
Prototyping Model in Software Engineering



* Customer Not clear with idea
* Throwaway Model
* good for technical and requirement risks
* Increase in Cost of Development.

0:28 / 4:51

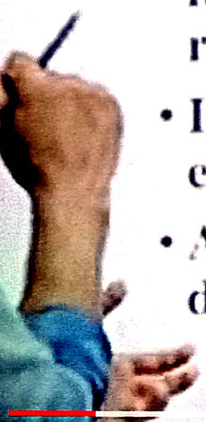
Incremental Model in Software Engineering | SDLC



* Module by Module Working
* Customer Interaction Maximum
* Large projects
* Early Release Product Demand
* Flexible to changes

Evolutionary model

- Evolutionary model is a combination of Iterative and Incremental model of software development life cycle.
- Incremental model first implement a few basic features and deliver to the customer. Then build the next part and deliver it again and repeat this step until the desired system is fully realized. No long-term plans are made.
- Iterative model main advantage is its feedback process in every phase.
- Also known as "Design a little, build a little, test a little, deploy a little model".

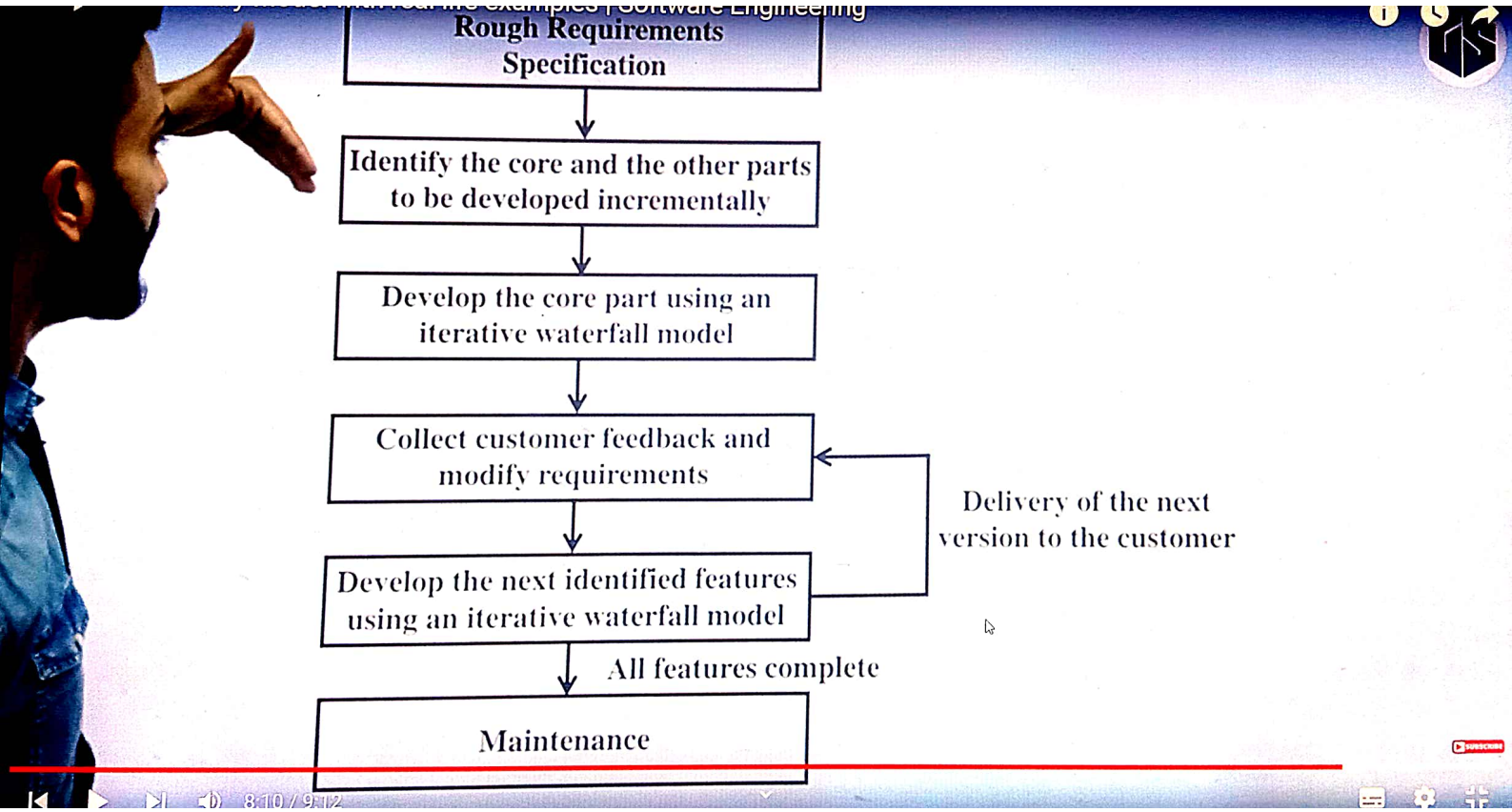


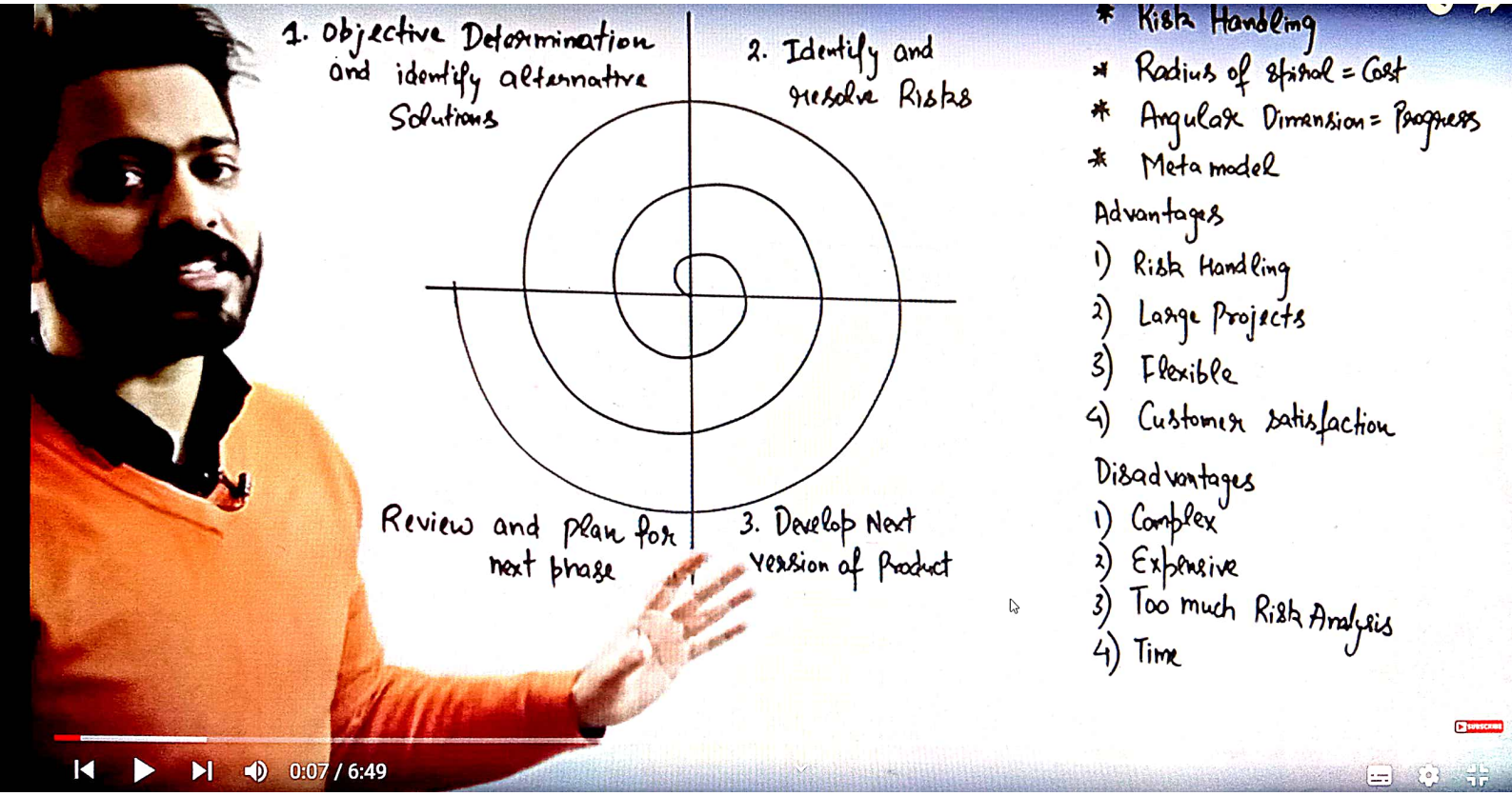
Advantages

- Customer requirements are clearly specified.
- Risk analysis is better.
- It supports changing environment.
- Initial operating time is less.
- Better suited for large mission-critical projects.

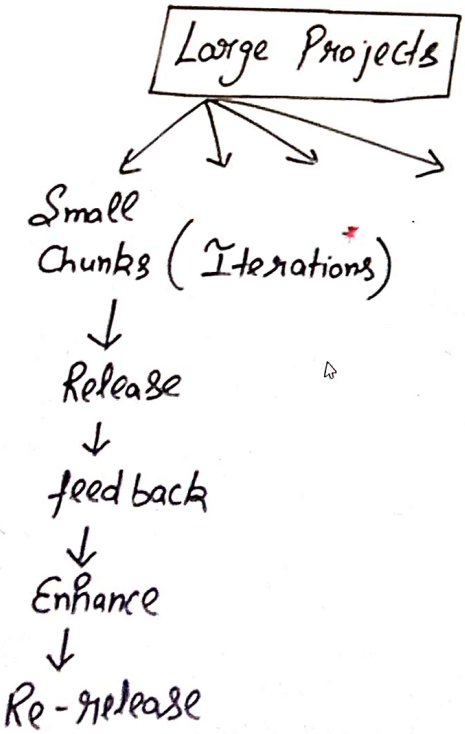
Disadvantages

- Not suitable for smaller projects.
- Cost
- Highly skilled resources are required.


Rough Requirements Specification
Identify the core and the other parts to be developed incrementally
Develop the core part using an iterative waterfall model
Collect customer feedback and modify requirements
Develop the next identified features using an iterative waterfall model
Delivery of the next version to the customer
All features complete
Maintenance


1. Objective Determination and identify alternative Solutions
2. Identify and resolve Risks
3. Develop Next version of Product
Review and plan for next phase
* Risk Handling
* Radius of spiral = Cost
* Angular Dimension = Progress
* Meta model
Advantages
1) Risk Handling
2) Large Projects
3) Flexible
4) Customer satisfaction
Disadvantages
1) Complex
2) Expensive
3) Too much Risk Analysis
4) Time
0:07 / 6:49

"Agile" (Move Quickly)



Advantages: 1) Frequent Delivery
2) Face to face communication with client
3) Changes
4) Time

Disadvantage: 1) Less documentation
2) Maintanance Problem

SCRUM Model in Software Engineering | Agile Technology

SCRUM

- One of the most popular agile methodology.
- Scrum is a lightweight, iterative and incremental framework.
- Scrum breaks down the development phases into stages or cycles called "sprints".
- The development time for each sprint is maximized and dedicated, thereby managing only one sprint at a time.
- Scrum Team has scrum master and product owner with constant communications on the daily basis.
- Keywords: Backlog, Sprint, Daily Scrum, Scrum master, Product owner,

Advantages

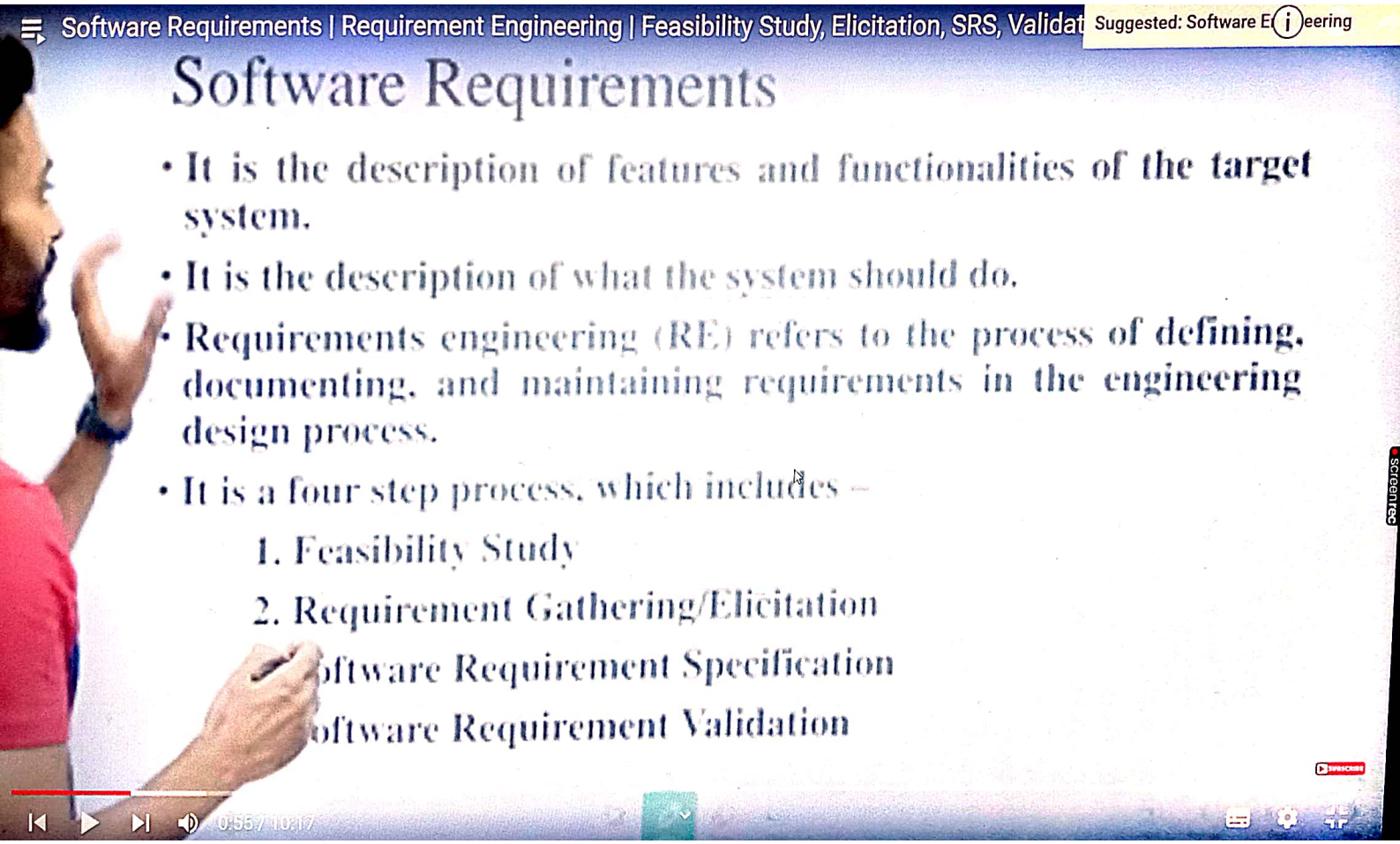
- Freedom & Adaption
- High-quality, low-risk product.
- Reduce the development time up to 40%
- Scrum customer satisfaction is very important.
- Reviewing the current sprint before moving to new one.

sadvantages:

- More efficient for small team size.
- No changes in the sprint.

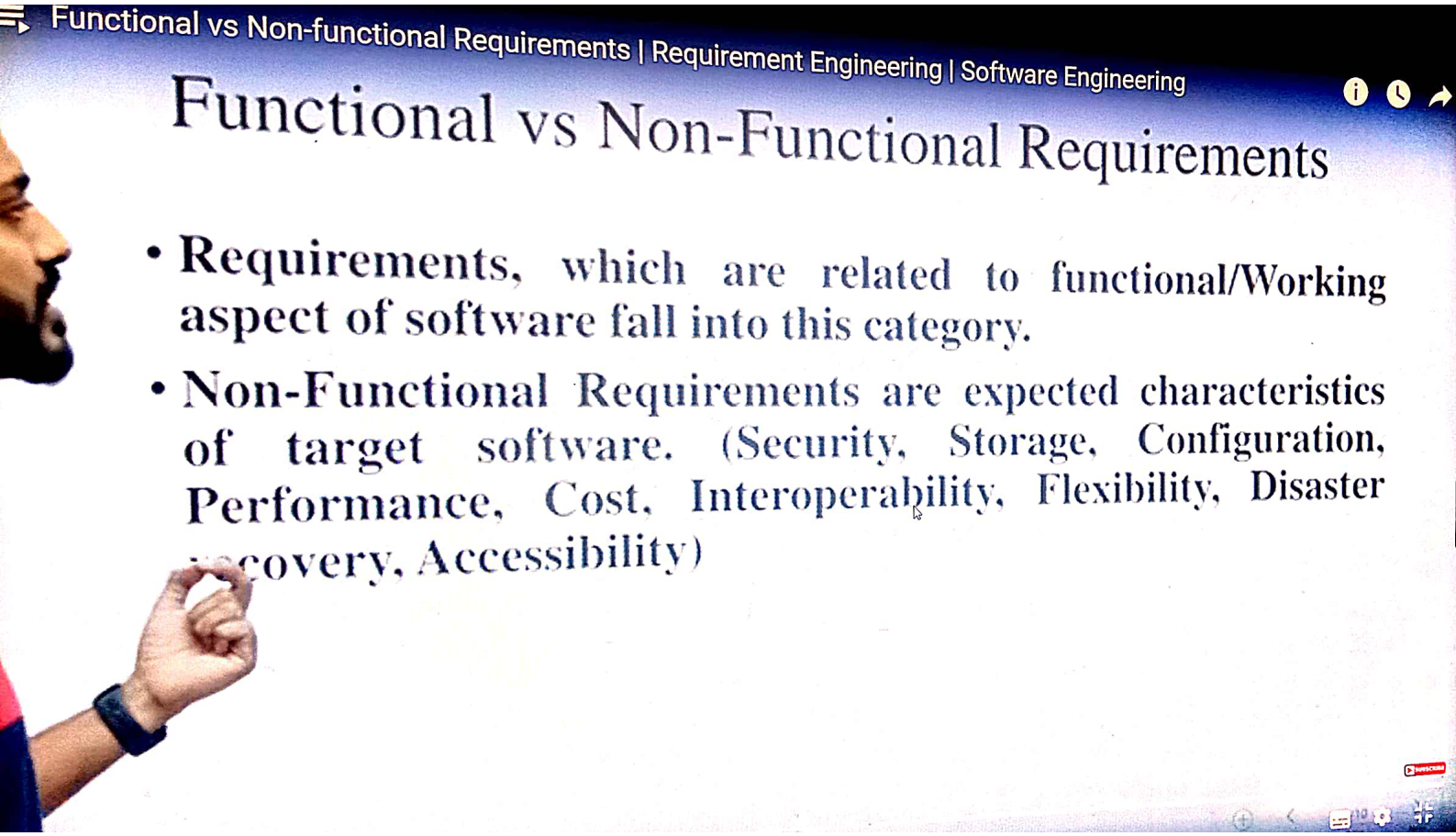
Comparison of All SDLC Models | Waterfall, Iterative, Prototype, Spiral, Increment, RAD, Agile etc.

Classical waterfall	Iterative waterfall	Prototype Model	Incremental Model	Evolutionary Model	RAD Model	Spiral Model	Agile Model
Basic, Rigid, Inflexible, Not for Real Project	Basic, Problem is well understood	User Requirement Not clear, Costly, No Early look on Requirements → High User Involvement → Reusability	Module by Module Delivery, Easy to test and debug	Large Projects	Time and Cost Constraint, User at all levels → Reusability	Risk, Not for small Projects, →No Early look on Requirements, →Less Experience Can work	Flexible, Advanced, Parallel, Process divided into sprints


Software Requirements | Requirement Engineering | Feasibility Study, Elicitation, SRS, Validat
Suggested: Software E eering
Software Requirements
• It is the description of features and functionalities of the target system.
• It is the description of what the system should do.
• Requirements engineering (RE) refers to the process of defining, documenting, and maintaining requirements in the engineering design process.
• It is a four step process, which includes –
1. Feasibility Study
2. Requirement Gathering/Elicitation
oftware Requirement Specification
oftware Requirement Validation
0:55 / 10:17

Tool support for Requirements Engineering

- Observation reports (user observation)
- Questionnaires (interviews, surveys and polls),
- Use cases
- User stories
- Requirement workshops
- Mind mapping
- Role-playing
- Prototyping


Functional vs Non-functional Requirements | Requirement Engineering | Software Engineering
Functional vs Non-Functional Requirements
• Requirements, which are related to functional/Working aspect of software fall into this category.
• Non-Functional Requirements are expected characteristics of target software. (Security, Storage, Configuration, Performance, Cost, Interoperability, Flexibility, Disaster recovery, Accessibility)

Software Requirements Specification(SRS)

- SRS is a description of a software system to be developed.
- It lays out functional and non-functional requirements of the software to be developed.
- It may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

SRS Structure

1. Introduction
    - 1.1 Purpose
    - 1.2 Intended Audience
    - 1.3 Scope
    - 1.4 Definitions
    - 1.5 References
2. Overall Description
    - 2.1 User Interfaces
    - 2.2 System Interfaces
    - 2.3 Constraints, assumptions and dependencies
    - 2.4 User Characteristics

3. System Features and Requirements

- 3.1 Functional Requirements
- 3.2 Use Cases
- 3.3 External Interface Requirements
- 3.4 Logical database requirement
- 3.5 Nonfunctional Requirements

4. Deliver for Approval