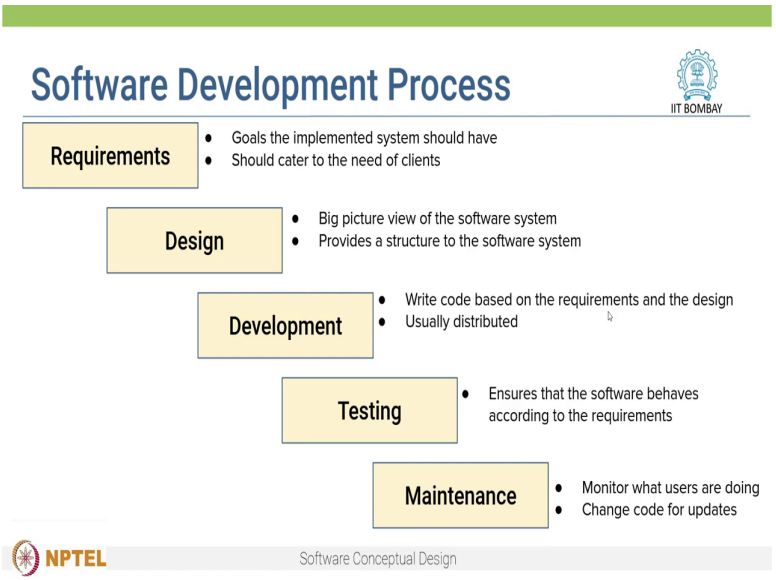
Software Conceptual Design

Dr. Sridhar Iyer
Dr. Prajish Prasad
Dr. T. G. Lakshmi
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture - 06
Software Development Models

(Refer Slide Time: 00:05)

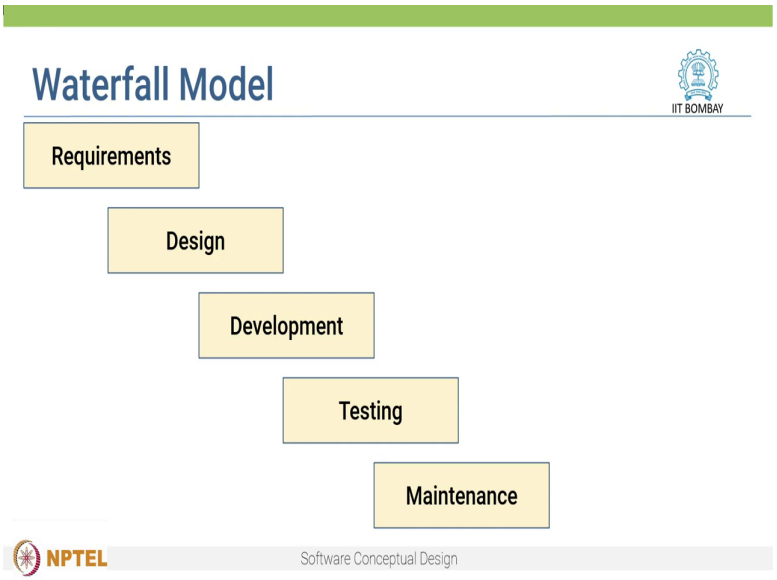


In the previous video lectures, we spoke about the Software Development process and the various phases in this process. We looked at requirements, which are the goals which the implemented system should have, and these requirements should cater to the need of the clients. Once we have gathered the requirements from the clients, we then come up with the design of the software system.

The design is the big-picture view of the software system and provides a structure to the entire software. Once the design is ready, we then write code in the development phase.

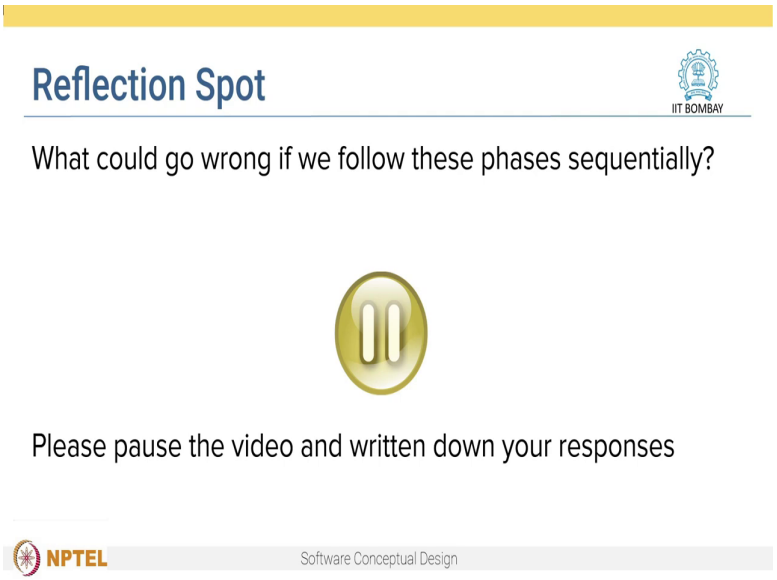
After we have written code, we rigorously test our system to ensure that it behaves according to the requirements. We then release the software to the clients, and in the maintenance phase, we monitor what the users are doing and change the code for updates and upgrades.

(Refer Slide Time: 01:24)



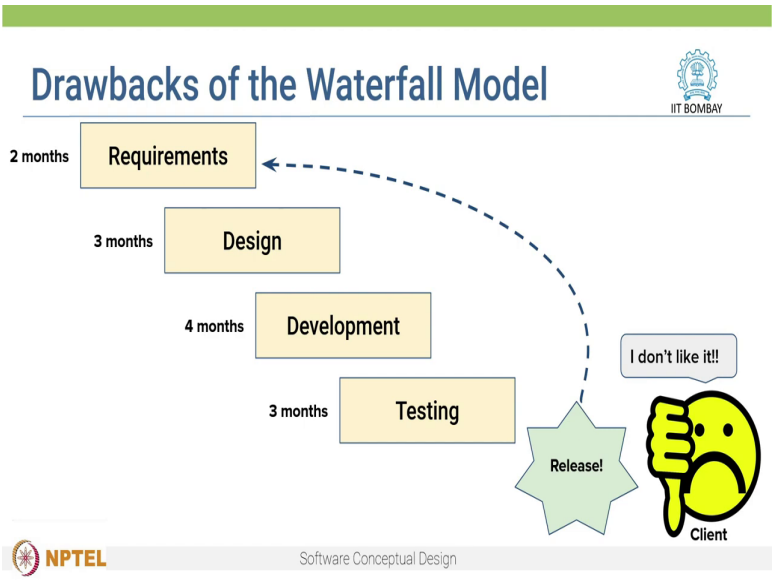
This process is commonly known as the waterfall model, where we start from the requirements, move to the design, the development, testing and finally, to the maintenance phase. We see that each of these phases occurs sequentially. Learners, I want you to now pause and reflect on this question. What can go wrong if we follow this process sequentially while creating a software system?

(Refer Slide Time: 02:04)



Please pause the video and write down your responses before you proceed.

(Refer Slide Time: 02:29)

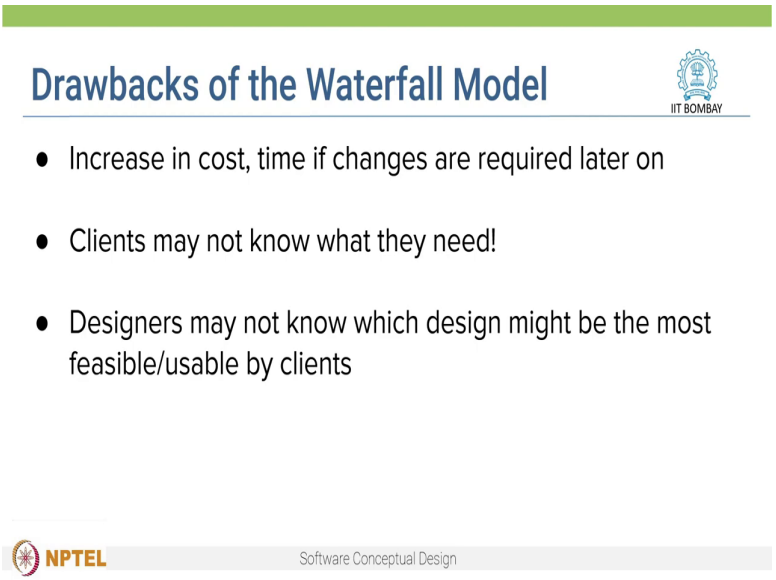


So, what are the drawbacks of this waterfall model? So, let us say, you are coming up with a software system, you have gathered the requirements and let us say it took you 2 months to do that. After that, you came up with a design which took around 3 months for your team. Now that you have the requirements and the design, you start developing the system. You write code for it, and this takes around 4 months.

Once the system is developed, then you rigorously test it identify bugs and defects and fix it. This can take, let us say, around 3 months. Now finally, your software is set for release, and you give it to your clients. Now, the clients are seeing your software for the first time, and unfortunately, they do not like it. It can be a particular feature which they do not like, it might be the entire user interface and so on.

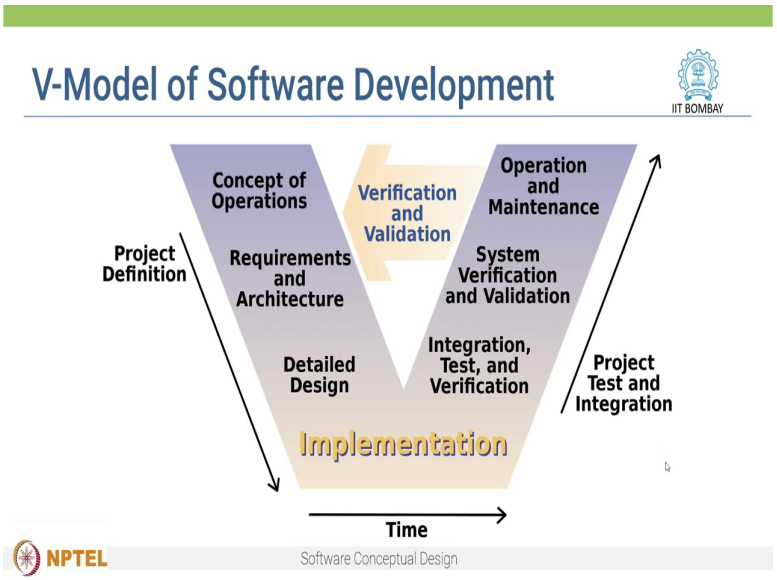
So, your software team has spent around 12 months building this system, and now you have to go back to the drawing board and refine your requirements based on the current needs of the clients.

(Refer Slide Time: 04:12)



So, what are the drawbacks of this waterfall model? As we saw, if changes are required later on, this leads to an increase in cost as well as time. We also saw that clients themselves may not know what they need in a software system, and their requirements can change over time. Third, the designers themselves may not know which design might be the most feasible or the usable by clients.

(Refer Slide Time: 04:55)

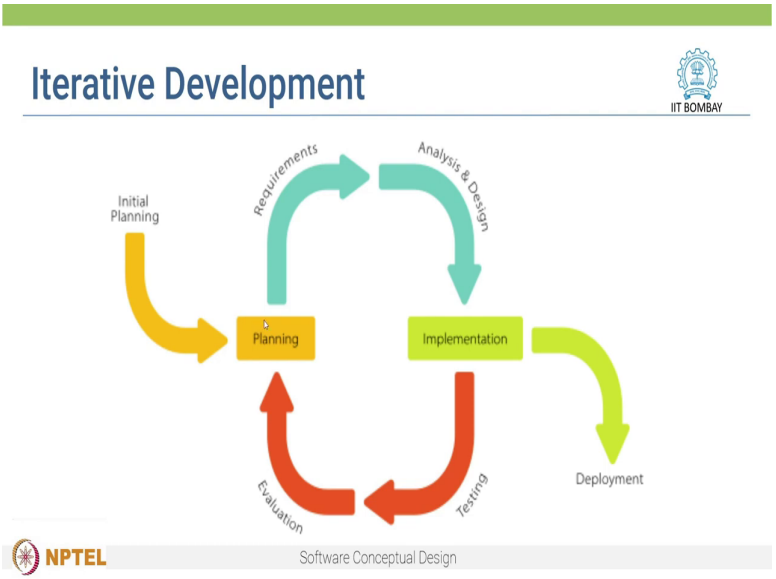


A variation of the waterfall model is the V-model of software development. As we can see in the figure, instead of a sequential process, the process steps are bent upwards after the

implementation phase to form the typical V shape. Like the waterfall model, we start from the requirements and come up with the design and implement the software system.

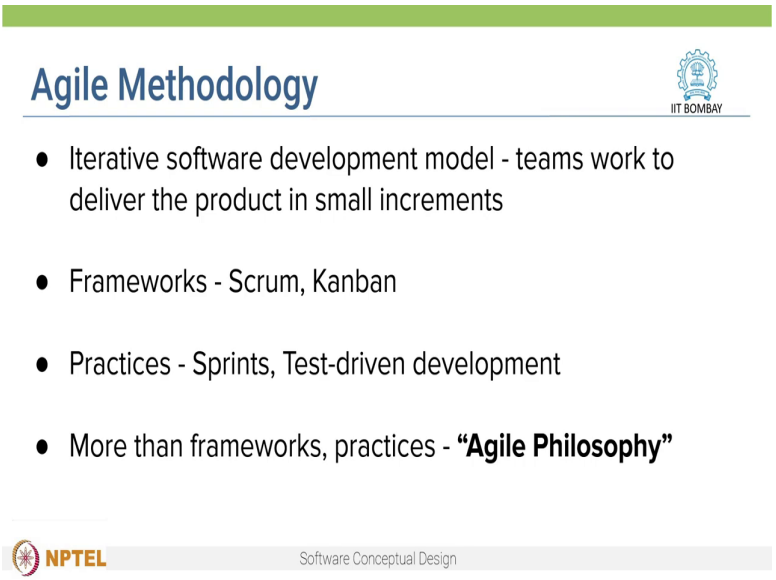
After that, we test and verify individual modules and validate it with the design. After validation of the individual modules, then we then verify and validate the entire system and compare it and validate it with the architecture.

(Refer Slide Time: 05:57)



To elevate the difficulties of the waterfall model and the V model, Iterative software development is a model which is being used widely today. In this model, software projects are built using multiple iterations. For example, in the first iteration, the development team plans and builds features for key requirements they deploy it and show it to the client and get their feedback. Even if the client wants changes, the team can quickly refine it in the next iteration. This process of development and refinement continues until the entire product is developed.

(Refer Slide Time: 07:00)



A popular iterative software development model is the Agile Methodology. In this model, teams work to deliver the product in small increments. Instead of waiting till the end to deliver the entire product, teams develop prototypes of key features and quickly release the prototype. There are several frameworks like Scrum and Kanban and practices such as sprints and test driven development.

The Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles of the agile philosophy. You can learn more about the agile software development philosophy and the frameworks like Scrum and Kanban in the extra resources provided to you this week.