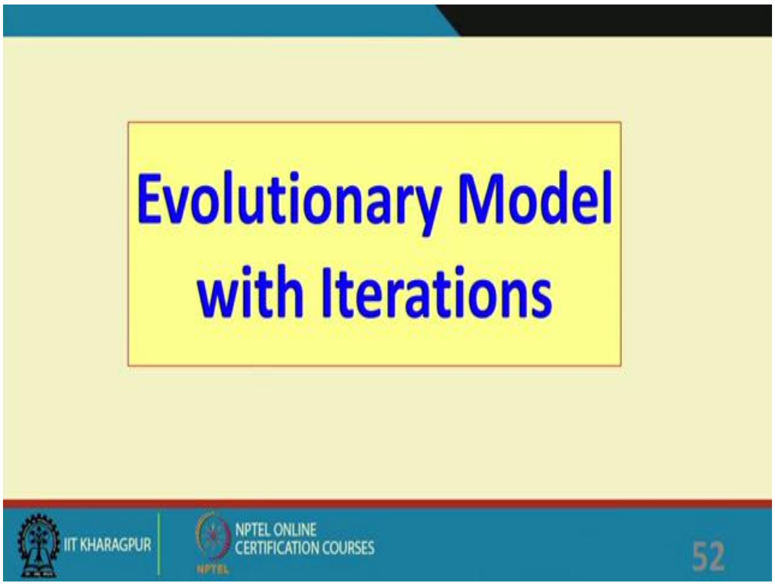
Software Project Management
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 08
Life Cycle Models – IV

Welcome to this lecture Over the last few lectures, we had looked at few important software development lifecycle models. We had started looking at the classical model which is very intuitive, the classical waterfall model and the derivatives of this model namely the iterative waterfall model, looked at the V model, prototyping model and so on.

And over the years few more models came up as the shortcomings of the waterfall model were noticed. These were the RAD model, the incremental model, incremental with iteration and the evolutionary model. In the last lecture, we had looked at the incremental model. Today, we will first look at the evolutionary model and then, we will look at the agile development model.

(Refer Slide Time: 01:25)

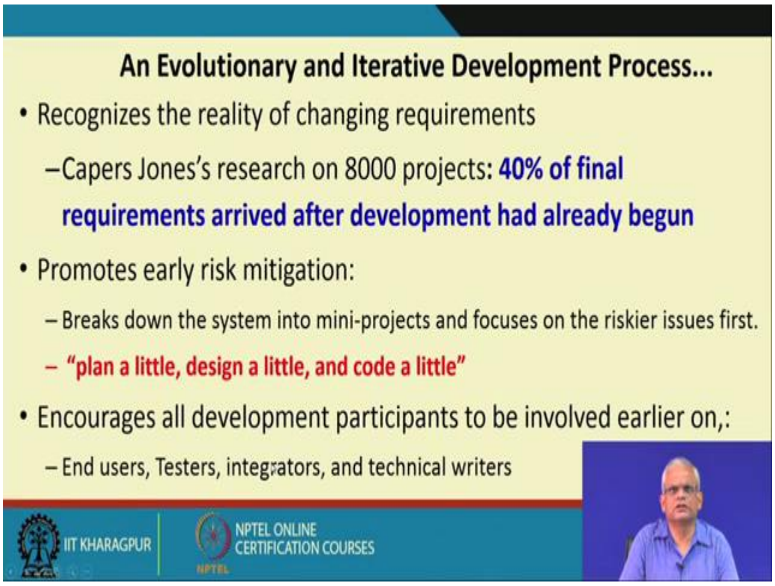


Evolutionary Model with Iterations

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
52

This is in fact, an evolutionary model with iteration; remember in the last class we had said that the term iteration is used when existing functionality which was delivered developed and delivered is iterated and refined based on customer feedback. Now let us look at the evolutionary model with iteration.

(Refer Slide Time: 02:01)



An Evolutionary and Iterative Development Process...

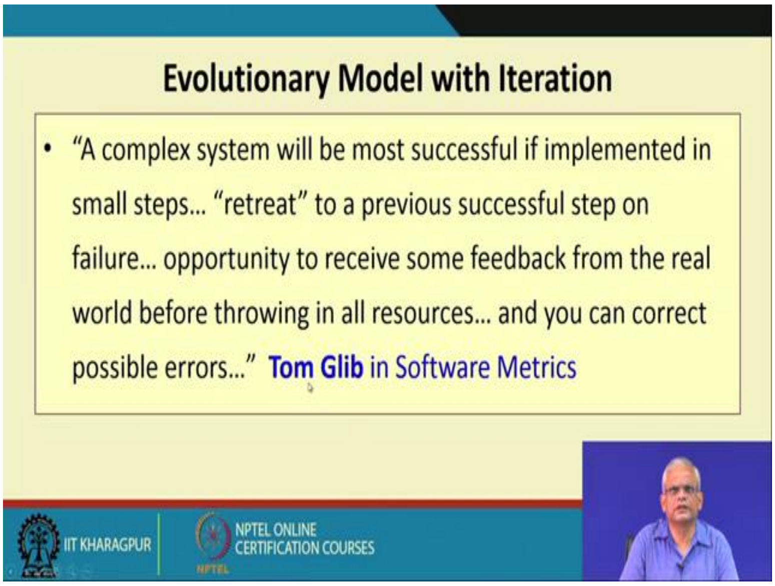
- Recognizes the reality of changing requirements
  - Capers Jones's research on 8000 projects: **40% of final requirements arrived after development had already begun**
- Promotes early risk mitigation:
  - Breaks down the system into mini-projects and focuses on the riskier issues first.
  - "plan a little, design a little, and code a little"
- Encourages all development participants to be involved earlier on,:
  - End users, Testers, integrators, and technical writers

One of the main problem that we had said about the waterfall based model is that the requirement is defined and frozen at the start of the project. But then, look at here about the practical situation here. Capers Jones researched on 8000 projects and he found that 40 percent of the requirements were arrived, when the development had already begun. If we had frozen the requirements, the start; then, this 40 percent changes will be very difficult to incorporate. Because in reality this is the average figure; 40 percent, it can even be more for some projects.

And therefore, there is need for a model which can effectively handle requirement changes. Requirements change due to various reasons maybe the business changes very rapidly. Once they have defined a business procedure may be after a month the business procedure changes or maybe the technology changes or maybe they might have not understood what is required and told something else, maybe they had forgotten to tell something and so on.

So, the requirements change due to various reasons and we need a model which can effectively handle requirement changes. One way that has been considered very promising is that each time do only small part of the system. Those are riskier likely to change the customer is not sure about it. Plan a little for this small work, design a little, code a little and once the small part is completed, give it to the customer for his evaluation and feedback. Here the customer is encouraged to participate. The end user the tester, integrator, technical writer all are involved in the development.

(Refer Slide Time: 04:54)



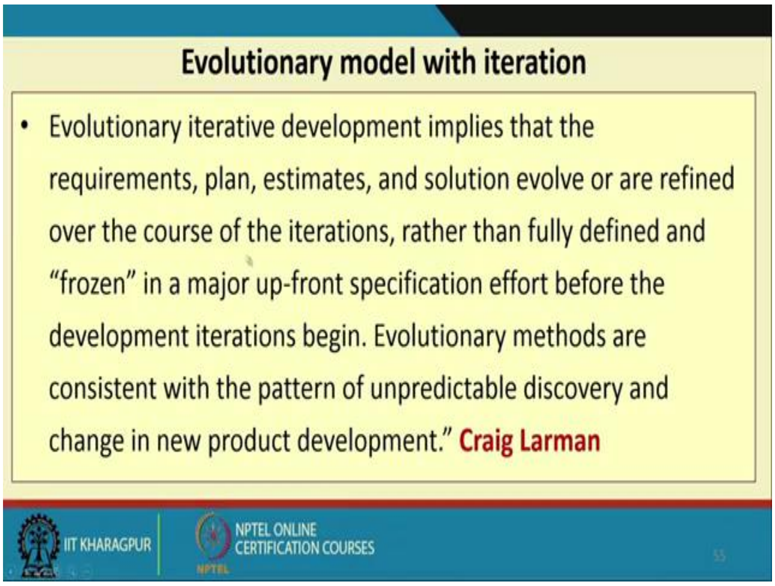
Evolutionary Model with Iteration

- "A complex system will be most successful if implemented in small steps… "retreat" to a previous successful step on failure… opportunity to receive some feedback from the real world before throwing in all resources… and you can correct possible errors…" **Tom Glib** in Software Metrics

Let us see what very distinguished person in this area. Tom Glib, what he has to say on the evolutionary model. He says that "A complex system will be most successful if implemented in small steps... "retreat" to a previous successful step on failure... If some functionality is totally not required, then at least you can go back to the previous version which is acceptable and then develop from there opportunity to receive feedback from the real world before throwing in all resources.. and you can correct possible errors.."

So, according to Tom Glib, the evolutionary model which he captures here in this few lines is a very promising model to handle changes to the requirement due to various reasons.

(Refer Slide Time: 06:04)



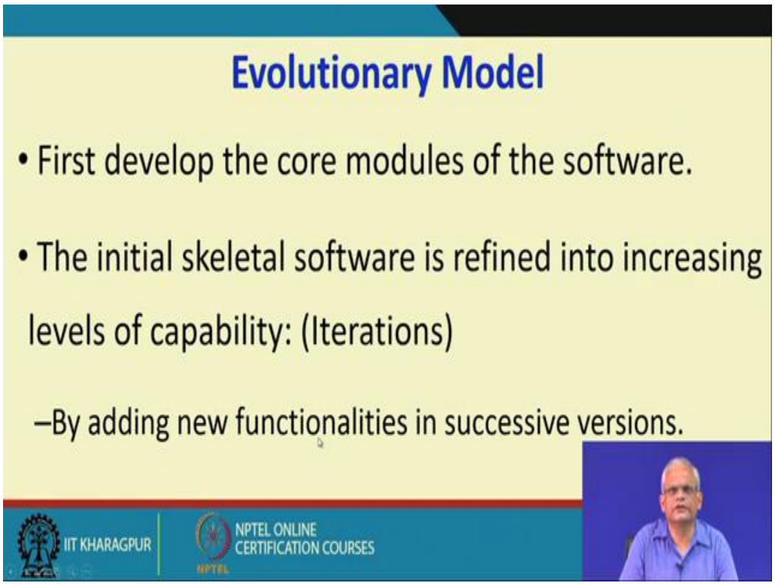
Evolutionary model with iteration

- Evolutionary iterative development implies that the requirements, plan, estimates, and solution evolve or are refined over the course of the iterations, rather than fully defined and "frozen" in a major up-front specification effort before the development iterations begin. Evolutionary methods are consistent with the pattern of unpredictable discovery and change in new product development." **Craig Larman**

Let us see what Craig Larman has to say about evolutionary model with iteration. "Evolutionary iterative development implies that the requirements, plan, estimates, and solutions evolve or are refined over the course of iteration, rather than fully defined and "frozen" in a major up-front specification before the development iterations begin. Evolutionary methods are consistent with a pattern of unpredictable discovery and change in new product development."

So, Craig Larman says that when a problem when a project starts it is very difficult to visualize all that is required. Only when you start doing, then you can know that what is wrong; what is really required and so on. So, is very critical about the waterfall model, where the requirements are fully defined and frozen before the development starts. And therefore, evolutionary model is the way to go and true to his words, most of the development models that are being used now which have become popular, use the evolutionary model in some form.

(Refer Slide Time: 07:34)

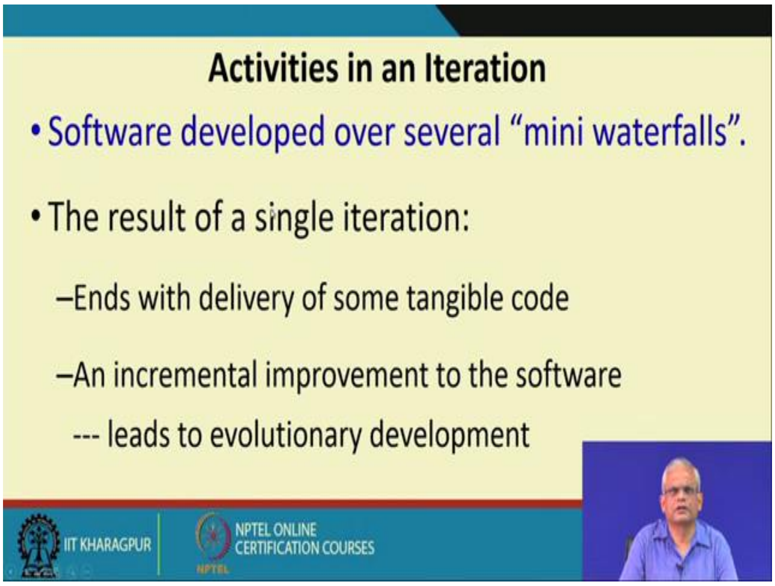


Evolutionary Model

- First develop the core modules of the software.
- The initial skeletal software is refined into increasing levels of capability: (Iterations)
  - –By adding new functionalities in successive versions.

Let us look at the evolutionary model. Here the core modules of the software is first developed and this is refined into increasing levels of capabilities which are called as iterations. In iteration new functionalities will be added and also the existing functionalities can change based on the user feedback. And each development, each iteration is developed through a mini waterfall model.

(Refer Slide Time: 08:04)



Activities in an Iteration

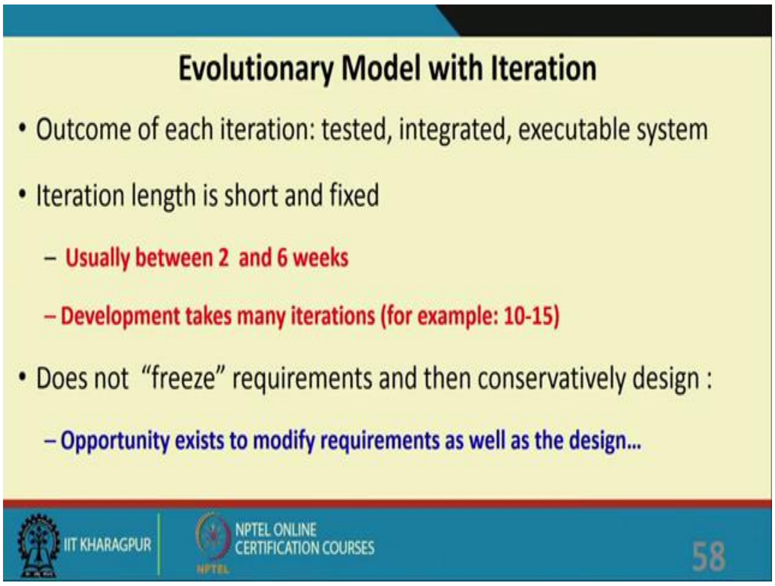
- Software developed over several "mini waterfalls".
- The result of a single iteration:
  - –Ends with delivery of some tangible code
  - –An incremental improvement to the software
    --- leads to evolutionary development

At the end of a iteration, some code is delivered to the customer invariably. At the end of development maybe the existing functionalities might have changed. It is possible that no new functionalities might have got added, but only the existing functionalities have got refined or it may so happen that existing functionalities have got changed a little bit and new functionalities have been added. So, these are an increment and therefore, it has some incremental development.

But then, the question is that how is the incremental development and the evolutionary development differ? Because as you can see that in both these models incremental and evolutionary, there are small increments over which the system is developed and these increments are deployed at the customer site and feedback obtained.

But let me mention here that in the purely incremental model as you might have remember from our last discussion that all those requirements are identified and these are planned into small increments. Each increment is developed deployed feedback obtained and changes can happen. But in an incremental model, all the requirements are gathered to start with whereas, in evolutionary model, it is not the case. Here after an overall understanding of the system we do not really capture all the requirements, but then start with some of the core or riskier modules.

(Refer Slide Time: 10:18)



Evolutionary Model with Iteration

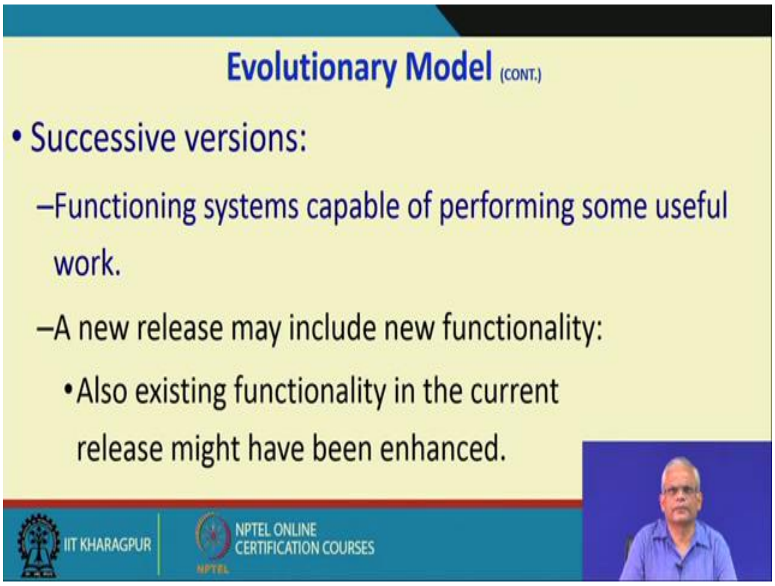
- Outcome of each iteration: tested, integrated, executable system
- Iteration length is short and fixed
  - Usually between 2 and 6 weeks
  - Development takes many iterations (for example: 10-15)
- Does not "freeze" requirements and then conservatively design :
  - Opportunity exists to modify requirements as well as the design...

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | 58

At the end of each iteration, a tested integrated executable system is developed, deployed at the customer site. The length of the each iteration is typically fixed, something like a 2 week to 1 month or maybe 1.5 month, 2 to 6 weeks and the development proceeds over many iterations. For example, a system may get developed over 10 to 15 iterations.

As you can see the evolutionary model, the requirements are not frozen. But here the customer is encouraged to participate, to give feedback based on which the further developments occur and the customer feedback is taken into account and functionalities get modified. So, here the requirements are modified in a iteration, some of the existing functionalities as well as new functionalities are delivered.

(Refer Slide Time: 11:32)

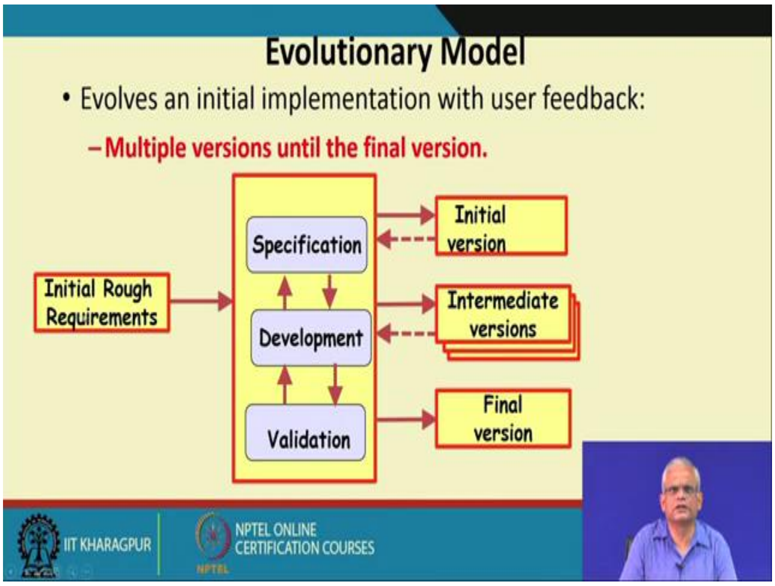


Evolutionary Model (CONT.)

- Successive versions:
  - Functioning systems capable of performing some useful work.
  - A new release may include new functionality:
    - Also existing functionality in the current release might have been enhanced.

Here, successive versions are developed and deployed at the cost customer site. Each version is capable of performing some useful work which the customer can use and give feedback and a new release can include a new functionalities and also existing functionalities may get modified and refined.

(Refer Slide Time: 12:05)

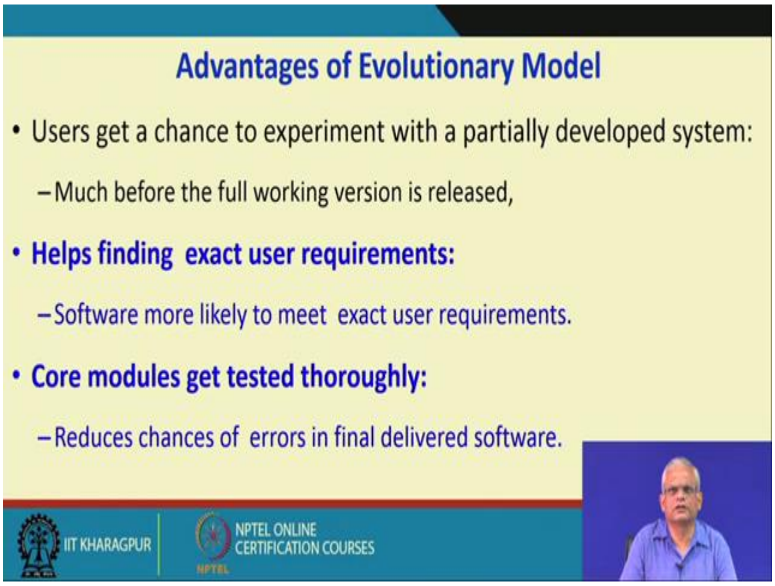


Evolutionary Model

- Evolves an initial implementation with user feedback:
  - Multiple versions until the final version.

We can diagrammatically represent it like this that initially rough requirements, just understand what the system needs to do and then, there are iterations. The iterations are shown here. In each iteration is a mini waterfall, where need to specify a small part; develop, validate and deploy at the customer site.

(Refer Slide Time: 12:46)



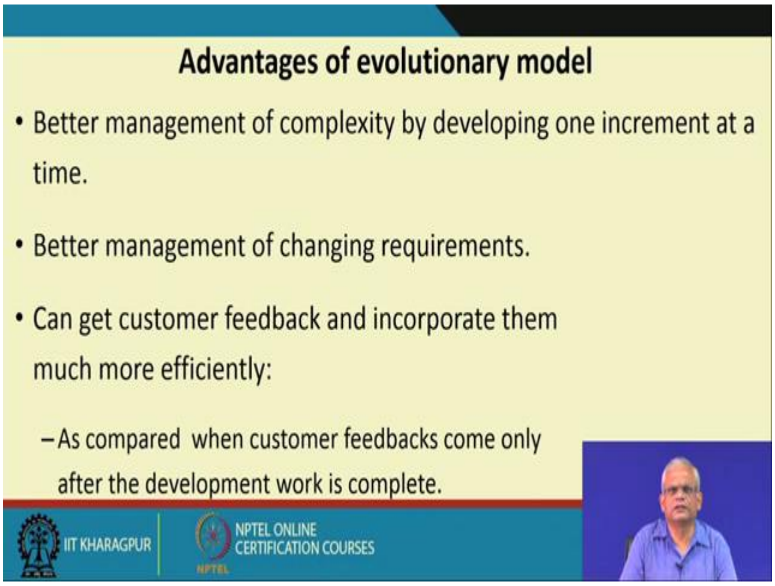
Advantages of Evolutionary Model

- Users get a chance to experiment with a partially developed system:
  - Much before the full working version is released,
- **Helps finding exact user requirements:**
  - Software more likely to meet exact user requirements.
- **Core modules get tested thoroughly:**
  - Reduces chances of errors in final delivered software.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

To start with the initial version, the feedback is obtained. There are several intermediate versions and then, the final version is deployed at the customer site. Evolutionary model has the advantage that, the customer can use the system and give feedback. And therefore, the evolutionary model is best suited to find the user requirements, the exact user requirements and once it is delivered to the customer, the customer to meet the customer requirements. Also since these are each time tested, each increment is tested and deployed at customer site. Therefore, multiple test activities, integration activities occur and therefore, it is less likely to contain bugs because these are tested thoroughly.

(Refer Slide Time: 13:43)

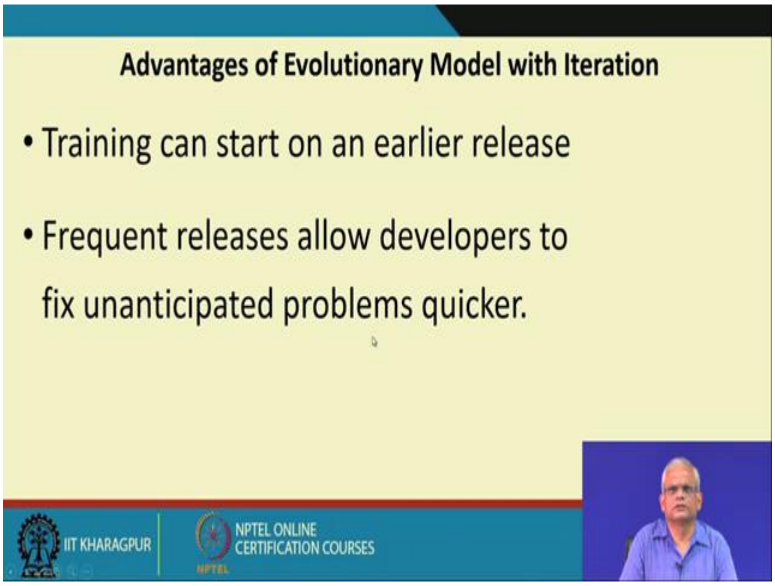


Advantages of evolutionary model

- Better management of complexity by developing one increment at a time.
- Better management of changing requirements.
- Can get customer feedback and incorporate them much more efficiently:
  - As compared when customer feedbacks come only after the development work is complete.

And since each time the developers focus each time on a small part, it helps in managing complexity; better manage changing requirements; customer feedbacks are obtained and incorporated.

(Refer Slide Time: 17:05)

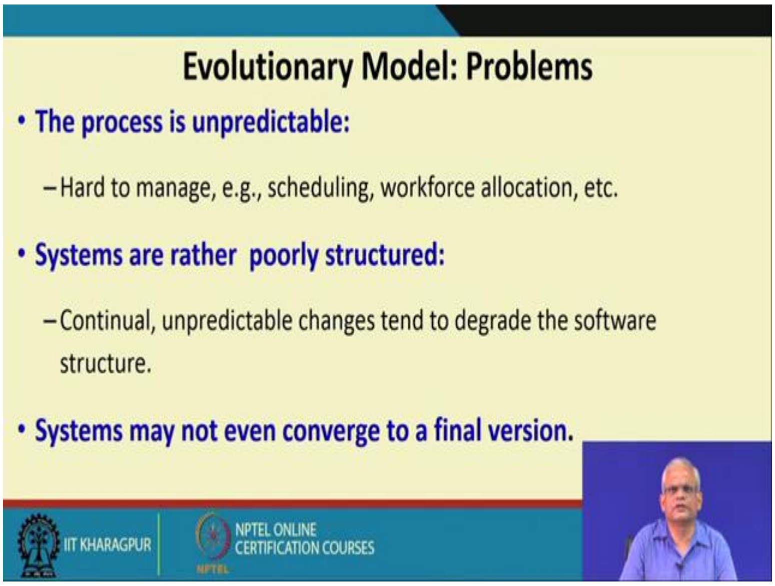


Advantages of Evolutionary Model with Iteration

- Training can start on an earlier release
- Frequent releases allow developers to fix unanticipated problems quicker.

The customer gets version quickly. They get trained on using the software. If there are bugs, these are fixed quickly in the next iteration.

(Refer Slide Time: 14:24)



Evolutionary Model: Problems

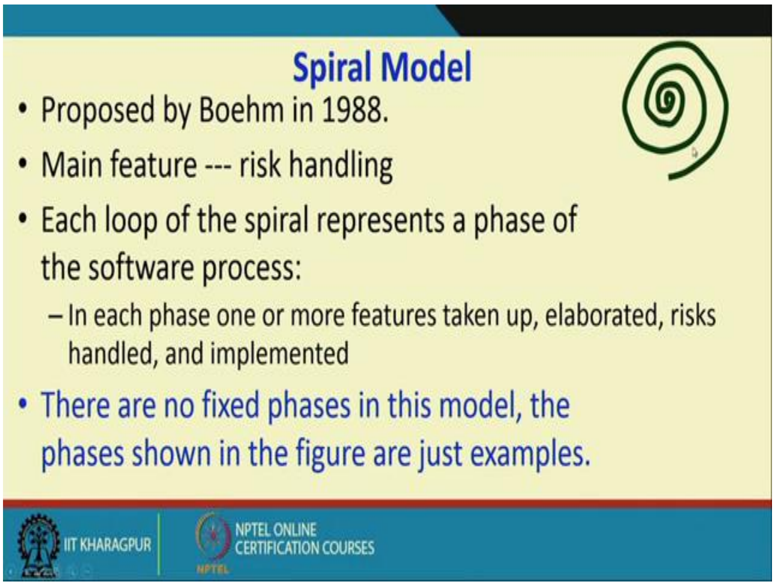
- The process is unpredictable:
  - Hard to manage, e.g., scheduling, workforce allocation, etc.
- Systems are rather poorly structured:
  - Continual, unpredictable changes tend to degrade the software structure.
- Systems may not even converge to a final version.

But then, there are some problems with the evolutionary model. Let us be aware about the problems of evolutionary model. You had seen that it is lot of advantages, but then here the process is unpredictable that is each time, we develop a increment deploy and get the customer feedback.

Now, what if the customer just keeps changing the requirements; keeps on asking for modifications and so on. Therefore, the process is unpredictable. We don’t know when it will finish, because the customer may keep on changing the requirements and also if we have a long term plan, then it becomes easier to deploy manpower, recruit them, schedule work monitor and so on. Here all these are problematic.

The system is poorly structured because there is no overall design made its only small parts that are designed and integrated. Since, the code is changed continually, the code structure degrades and the system may not even converge to a final version if each time the customer keeps on asking for changes.

(Refer Slide Time: 16:04)



Spiral Model

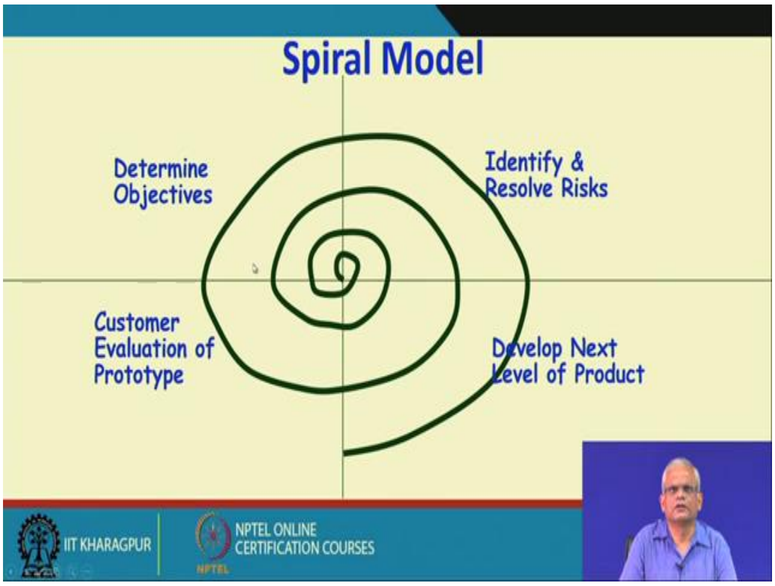
- Proposed by Boehm in 1988.
- Main feature --- risk handling
- Each loop of the spiral represents a phase of the software process:
  - In each phase one or more features taken up, elaborated, risks handled, and implemented
- There are no fixed phases in this model, the phases shown in the figure are just examples.

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Now, let us look at the spiral model. The main feature of the spiral model is risk handling. Just look at the spiral here diagram the model follows spiral, we will see that. It was proposed by Boehm. Here each loop of the spiral is called as a phase and the system gets developed over many phases. There is no fixed number of phases. As the system develops more and more phases are created. There can be many loops on this spiral until the final delivery takes place. But over each phase that is each loop of the spiral, one feature is identified and is developed.

(Refer Slide Time: 17:11)



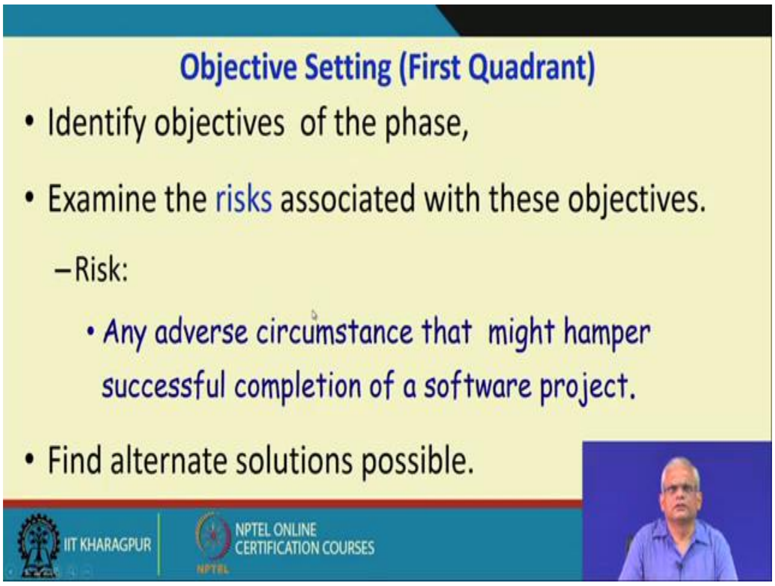
Spiral Model

Determine Objectives
Identify & Resolve Risks
Customer Evaluation of Prototype
Develop Next Level of Product
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

If you can see there are four quadrants here on the spiral model, the first quadrant what needs to be done in that phase is identified and then, if there are any uncertainties. Technical uncertainties or the user requirements are not clear. A prototype is developed to resolve the risk and then, the next iteration is developed given to the customer for his evaluation and then, another feature is taken up.

So, over each iteration one or more features are taken up. The risk is resolved and then the next development occurs and customer feedback. As can be seen here, this model is ideally suited for very risky projects, where the technology is not known. It is possibly the first time that this kind of software is being developed. The technology is not certain and so on. So, each time some features the most riskiest feature is identified. Risk is resolved, it is developed and given for customer feedback and the next set of features are taken up.

(Refer Slide Time: 18:49)



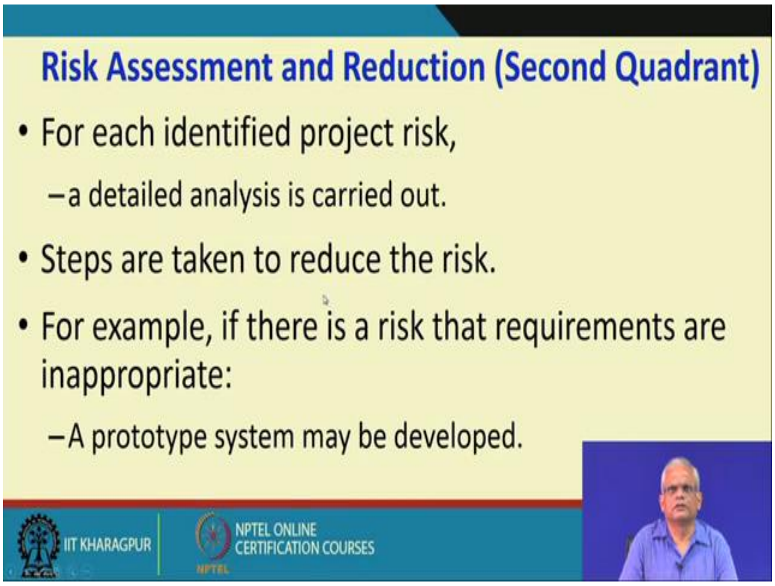
Objective Setting (First Quadrant)

- Identify objectives of the phase,
- Examine the risks associated with these objectives.
  - Risk:
    - *Any adverse circumstance that might hamper successful completion of a software project.*
- Find alternate solutions possible.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Here, the risk is any adverse of circumstance that might hamper the successful completion of the project. It can be technical issues are not clear, it can be that the customer is not able to clearly specify what is required and here a prototype is developed to overcome the risk.

(Refer Slide Time: 19:18)



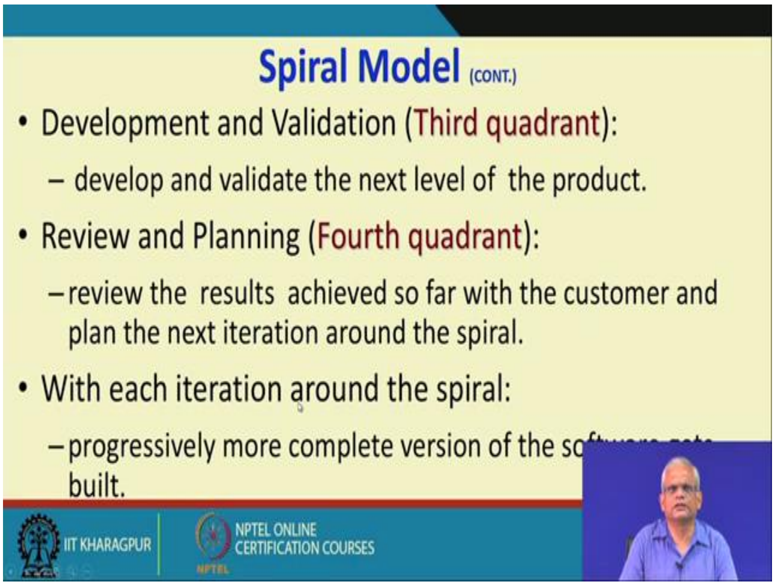
Risk Assessment and Reduction (Second Quadrant)

- For each identified project risk,
  - a detailed analysis is carried out.
- Steps are taken to reduce the risk.
- For example, if there is a risk that requirements are inappropriate:
  - A prototype system may be developed.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

If the risk is that requirements are inappropriate, prototype is developed given to the customer so that he can more clearly say what is required.

(Refer Slide Time: 19:34)

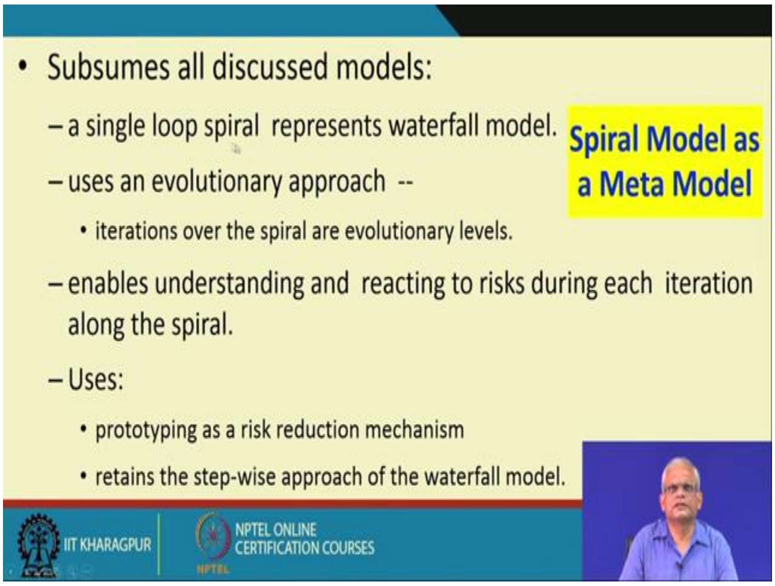


Spiral Model (CONT.)

- Development and Validation (Third quadrant):
  - develop and validate the next level of the product.
- Review and Planning (Fourth quadrant):
  - review the results achieved so far with the customer and plan the next iteration around the spiral.
- With each iteration around the spiral:
  - progressively more complete version of the so
    built.

The third quadrant is developed and validate and the fourth quadrant is give the developed solution to the customer for feedback and plan for the next iteration. We can see that with each iteration around the spiral, more and more complete versions of the software get built.

(Refer Slide Time: 19:56)

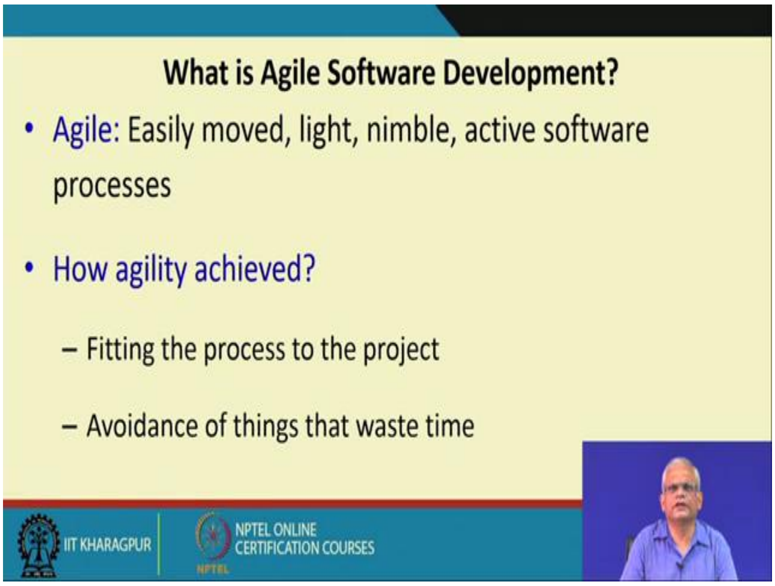


Spiral Model as a Meta Model

- Subsumes all discussed models:
  - a single loop spiral represents waterfall model.
  - uses an evolutionary approach --
    - iterations over the spiral are evolutionary levels.
  - enables understanding and reacting to risks during each iteration along the spiral.
  - Uses:
    - prototyping as a risk reduction mechanism
    - retains the step-wise approach of the waterfall model.

The spiral model is also called as a Meta model because a single loop of the spiral is a waterfall model, there are iterations and therefore, it incorporates evolutionary and incremental models. It uses prototyping. The stepwise approach of the waterfall model is retained and therefore, this model can be degenerated or can be used as any other model. But then, need to remember that the spiral model is suitable for very risky projects and large projects.

(Refer Slide Time: 20:45)



What is Agile Software Development?

- Agile: Easily moved, light, nimble, active software processes
- How agility achieved?
  - Fitting the process to the project
  - Avoidance of things that waste time

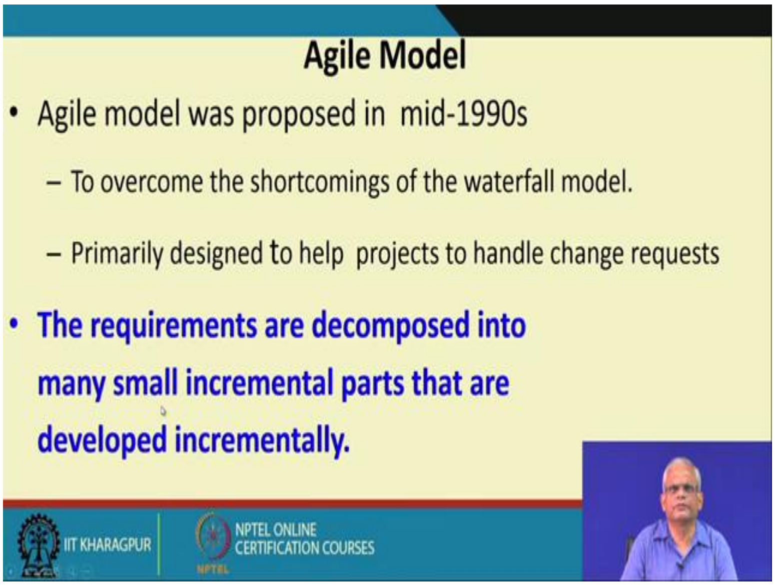
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

Now, let us look at the agile models. If we look at the meaning of agile in dictionary; agile means easily moved, very fast, nimble, active software process etcetera these are some of the terms that are given as explanation for agile model in dictionary. But then, agile means very fast. So, the development in agile model should occur very fast in much less time than it occurs in other development models.

But then, how is it developed very fast; how his agility achieved? Agility is achieved because here it gives lot of flexibility to change the process that is if some project requires some work to be done some specific activity, it will be done; if it is not required for another project, it will can be easily skipped because the model is very flexible. Unlike the waterfall model, where there are predefined activities and steps and so on.

Here, we can fit the process to the project, but more important is we can avoid things that waste time. That is to get it done very fast, we need to avoid whatever is a wastage of time. What are the wastage of time? Maybe if we are preparing very elaborate documents, we are mentioning that in the iterative waterfall model 50 percent of the development effort and time goes in preparing documents. Here we will see that it does away in preparing these elaborate documents and saves lot of time.

(Refer Slide Time: 22:47)



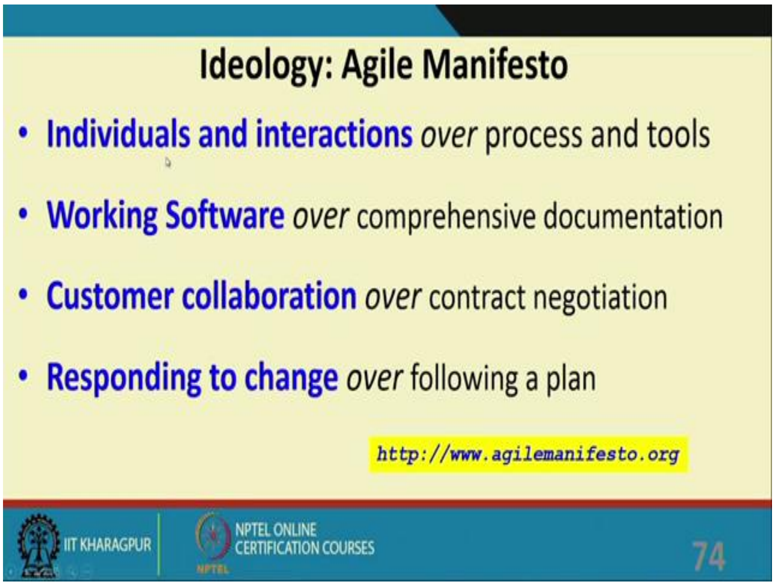
Agile Model

- Agile model was proposed in mid-1990s
  - To overcome the shortcomings of the waterfall model.
  - Primarily designed to help projects to handle change requests
- **The requirements are decomposed into many small incremental parts that are developed incrementally.**

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

It was proposed in mid 90's, mainly to overcome the shortcomings of the waterfall model. One important characteristic is to handle change requests. We had seen that the evolutionary model is a good way to handle change requests and the agile models do incorporate evolutionary development; evolutionary and incremental development. Here the requirements are decomposed into small incremental parts and development precedes incrementally.

(Refer Slide Time: 23:24)



Ideology: Agile Manifesto

- **Individuals and interactions** *over* process and tools
- **Working Software** *over* comprehensive documentation
- **Customer collaboration** *over* contract negotiation
- **Responding to change** *over* following a plan

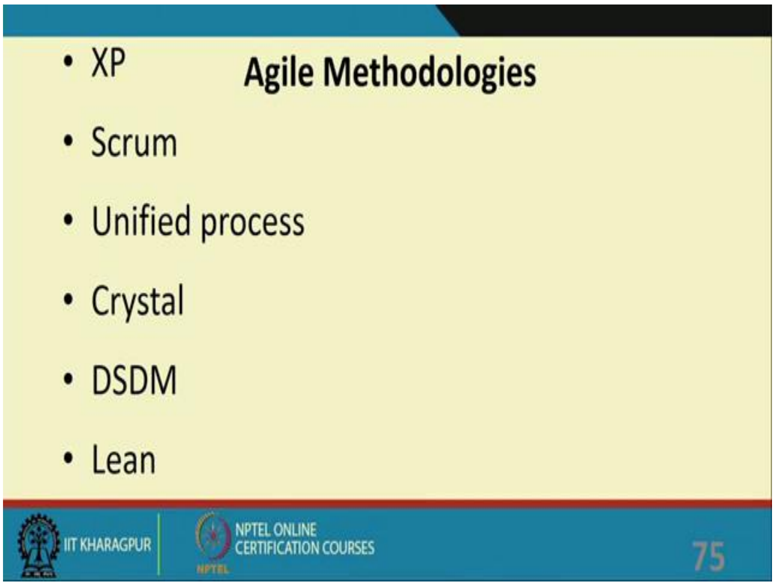
http://www.agilemanifesto.org

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
74

There are a few things which are important here. Instead of producing elaborate documents, following process rigorously here the individuals and they interact with each other rather than passing on a document to each other. They explain to each other through interaction. In a waterfall model, the progress of the development is measured in terms of the documents produced; is the requirement document complete; is the design document complete; is the detailed design complete; is the code documentation complete; test documentation complete. But here, the progress is measured in terms of the working software that has got developed. How many iterations have been completed and each iteration, some working software is deployed at the customer site.

Here, the customer is encouraged to participate in development, is encouraged to collaborate in the development rather than just signing a contract at the start of the project and here, it is assumed that changes will occur, the requirements will change and how to handle those rather than making a long term plan and following that plan, here changes are welcome and how to handle those changes those are important issues here.

(Refer Slide Time: 25:01)



Agile Methodologies

- XP
- Scrum
- Unified process
- Crystal
- DSDM
- Lean

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
75

The agile methodologies is an umbrella term. There are many processes development processes which have come up like extreme programming or XP, Scrum, Crystal, DSDM, Lean etcetera each of them have the features of the agile model. But then but then they differ in some ways, each one has its own focus. New things that are introduced in each of them as you will see new terms, new activities and so on.

But then, they all have the features of the agile methodologies which we just so discussed in the last slide, development over increments, interaction with the customer, less documentation interaction among the developers, testing in integrating and testing or each iteration, deploying getting customer feedback and so on.

For this lecture, just come to the end and in the next lecture, we will discuss more details about the agile development model.

Thank you.