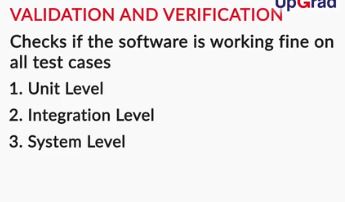
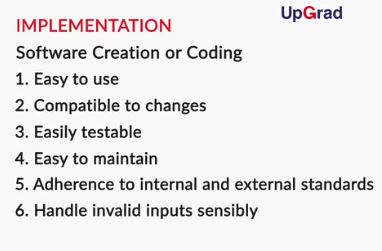
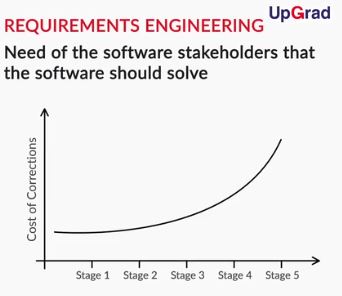
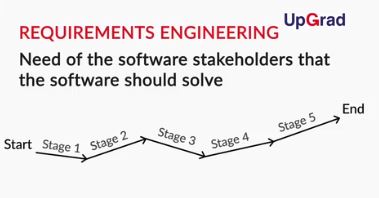
# Introduction to SDLC

Welcome to the module on ‘Agile and Extreme Programming’. Let’s start with a quick introduction.

In this module, you will first be introduced to the concept of the ‘software development life cycle’ and the various steps involved in the development of software. Then, you will learn about Agile methodologies in detail. Basically, there are various ways to implement Agile, out of which we will discuss ‘extreme programming’ in depth.

The first session of this module is dedicated only to the basic idea of the software development lifecycle and the steps involved in building software.

So, let’s get started!



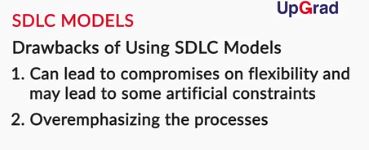
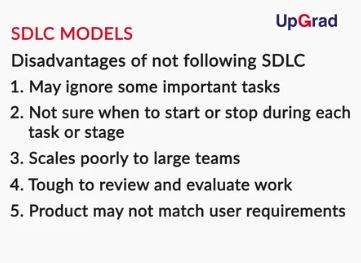
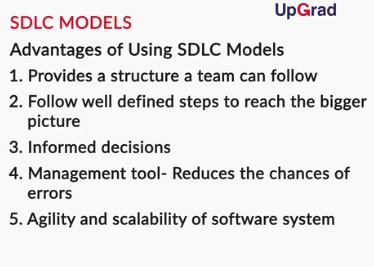
<https://www.geeksforgeeks.org/software-testing-basics/>

In this video, you learnt something counter-intuitive! Software engineering is not limited to  coding only. There is much more to it than this. You also explored the various steps involved in creating software, which are —

1. Requirements engineering
2. Design
3. Implementation
4. Verification and validation
5. Maintenance

# Why SDLC Models?

In this segment, Professor will discuss the merits and demerits of using and not using the SDLC models. This will give you a good rationale of why is it advantageous to follow a certain SDLC model to develop a software



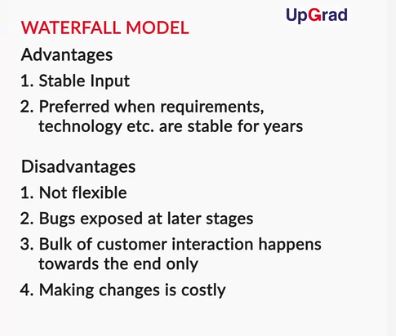
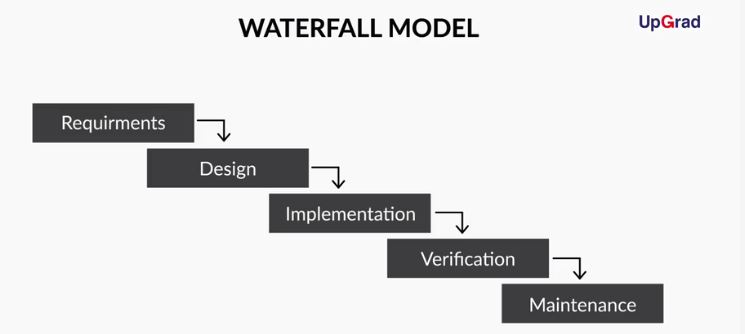
So, in this video, you learnt that the advantages of following an SDLC model outweigh the disadvantages of following it. Following are few of the advantages discussed in the video:

* Provides a structure for the team  to follow and work
* Forces you to think about the big picture and to follow well-defined steps to reach it
* Without it, you may make decisions that are may seem good at the moment but maybe globally misdirected
* It is also a management tool that can help you plan your work development and thus reducing the risk of things going wrong
* Correct application of SDLC model contributes to the agility and scalability of a software system

In the next session, we will move on to a particular software development methodology called Agile, which takes care of all these steps, but in its own efficient way.

# Waterfall Model

Now that you are aware of the advantages of following an SDLC model, let’s now move on to the video below which discusses a heavyweight model called Waterfall model



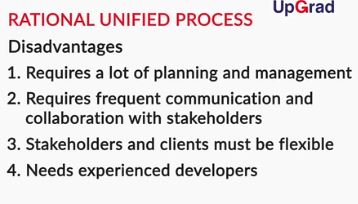
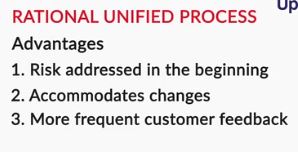
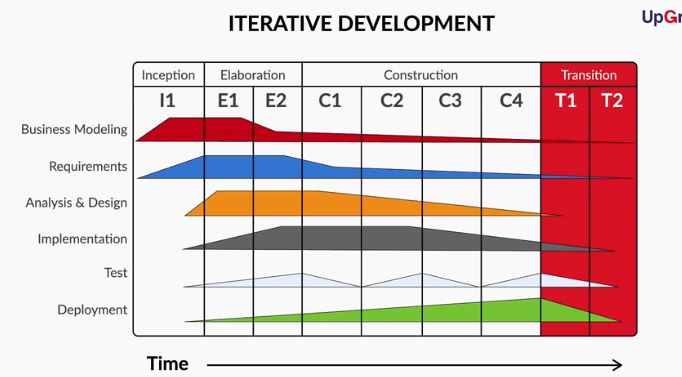
# Rational Unified Process

In this segment, you will learn about a model that tries to take care of few of the disadvantages of Waterfall model. The model we shall discuss here is Rational Unified Process model. Let’s see what it is.

You learnt the following in this video.

RUP works in an iterative way. At each iteration, we have 4 phases. The four phases are Inception, Elaboration, Construction and Transition. In each of the phases, we carry out the basic software development activities. Apart from many advantages this model also had few disadvantages. Following were the advantages discussed:

* A lot of planning and management is a requirement
* Requires frequent communication and collaboration with the client or stakeholders
* Stakeholders and client must be flexible to adopt changes in plans
* Developers need to be experienced, so they can identify risks in important issues early (i.e. not a good model, at least during initial iterations for the inexperienced developer)



Q. **Steps in RUP**

Which of the following statement about iterative development in rational unified process are correct?

**During the inception, the majority chunk of work is in the business modelling and requirements domain**

**Feedback :**

During inception phase, the scope of the project is defined. The stakeholders take a call on the budget of the project and see if it makes any business sense or not. Hence requirements and businees modelling are the major activities carried out in inception stage.

**Correct**



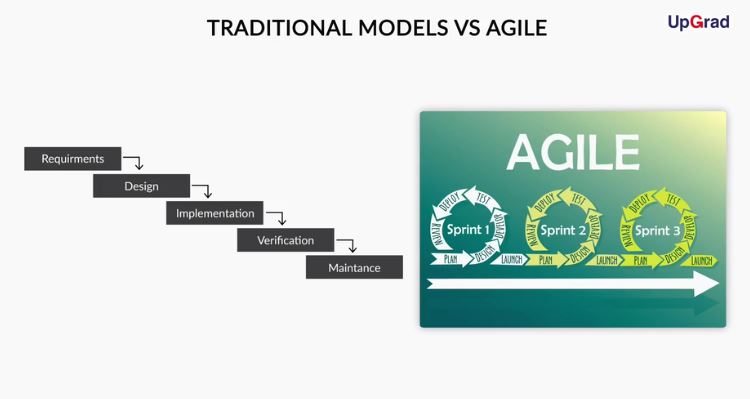
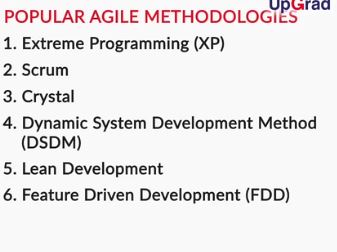
**The BRAITD steps happen at each stage of the process**

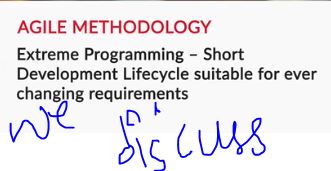
**Feedback :**

Business Modelliing, Requirements, Testing,Analysis and Design,Implementation,Testing,and Design are the activities which are carried out at every stage of the process. The extent to which each of these activities are carried out in each stage may vary from one stage to another.

# Agile and Extreme Programming

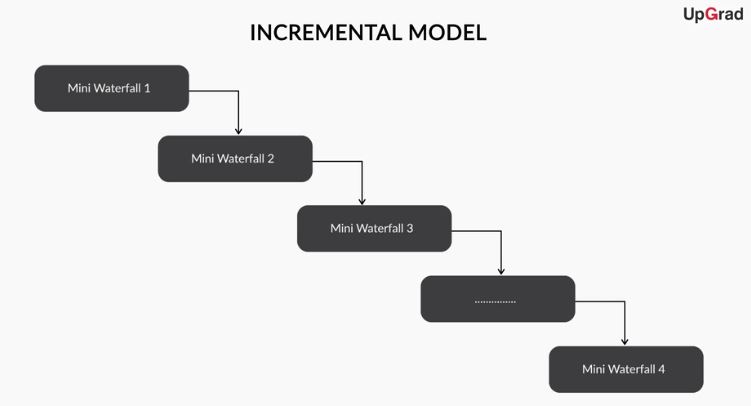
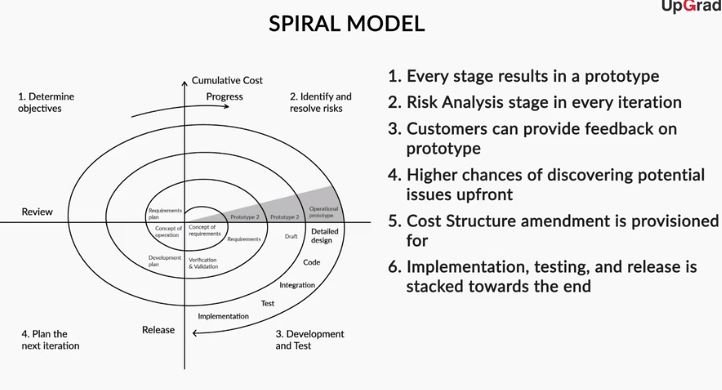
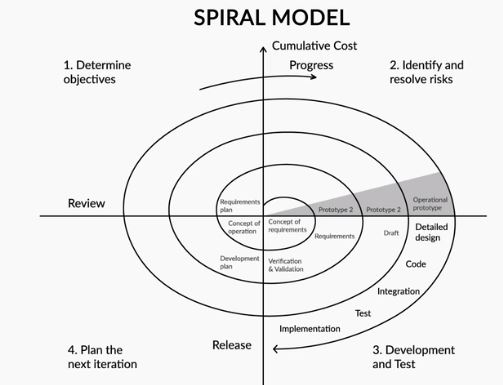
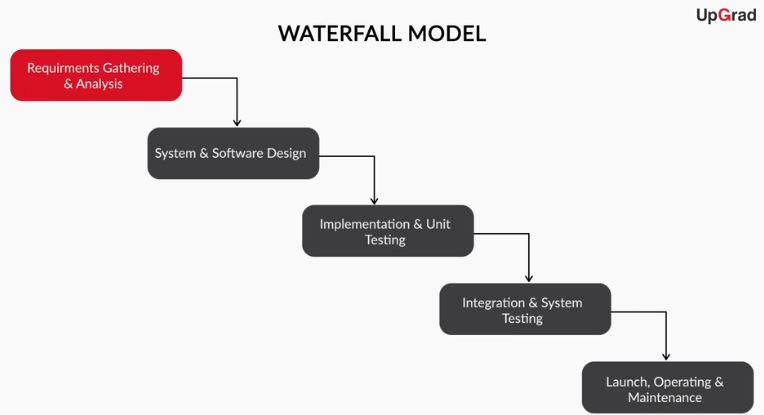
In this segment, Professor will discuss one of the widely used SDLC models Extreme Programming. Let’s see what it is





# Industry Demonstration

Each SDLC model answers a different question. Some suit a situation well with others may not fit into a particular situation. A particular scenario can be better handled by Waterfall model, while others may need some other model to work on them. How do we choose a model? How do we decide which model will suit a given situation? This is something that our industry expert Varun Sehgal tries to answer in the video below

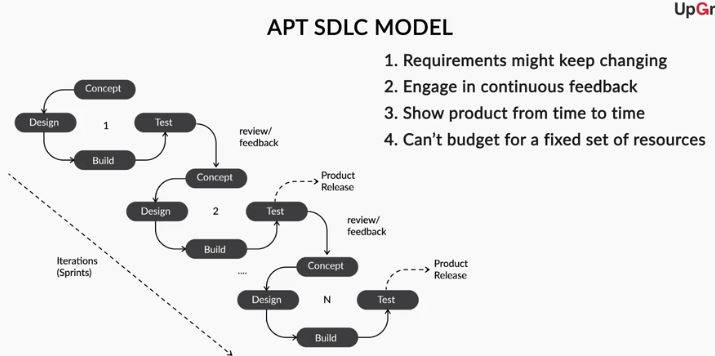


So, we made following changes to the GEM statement.

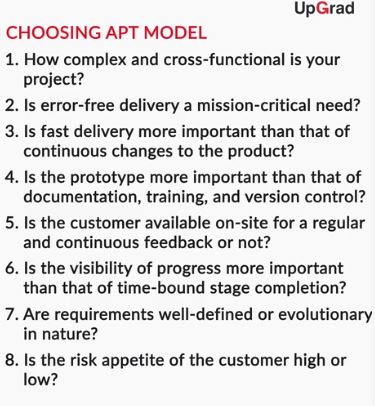
* Greater Emphasis on risk management
* The only Conceptual idea of the system is required
* The government wants to have a prototype that can be enhanced
* A change in the cost structure is no more a problem

In this case, Spiral model was used.

Again, let's make one more change to the GEM statement and see what happens



This time Agile was found to have answered our question. Hence, it can be concluded that a model can be chosen as per the problem statement. Try to attempt the question below to verify your understanding.



Q. **Choose the right model**

You’re the product manager of XYZ Corp. and you’re part of a team that’s looking to create an e-commerce App for company’s recent decision to foray into the groceries business. After a few rounds of discussions within the team and that with leadership, you have agreed on the stipulations around which the app needs to be built — a few of which are mentioned below:

1. The app should allow consumers to buy groceries and get them delivered to their doorstep.
2. Following categories of items need to be sold:
   * Fruits and Vegetables
   * Groceries and staples
   * Beverages
   * Home and Kitchen
   * Personal Care
   * Instant Food and Snacks
   * Furnishing and Household needs
   * Daily Needs
   * Meats and Seafood
3. Consumers should be able to search or navigate through category menu to find the items they want to buy.
4. There should be a mechanism for the consumer to return/raise a dispute in case the delivery is not up to the mark.
5. The customer care support should be integrated into the app at appropriate entry points.
6. To begin with, we can offer customers a pre-calculated fixed delivery date/time. At a later point in time, we would want them to avail of various delivery slots.
7. We want to restrict the ordering to two cities for now: Delhi NCR and Bangalore. Expansion into other cities to follow post first launch.
8. We will go ahead with a few categories, to begin with, and roll out the rest of the categories as we go along.
9. There’s no fixed end date in sight. The earlier we can launch, the better it is.

Based on the information above, can you identify which Software Development Life Cycle model should work best here? Also, state your reasons for making your choice.

Ans: **Suggested Answer**

Agile Methodology is the right choice. Points “6, 7, 8” — all point in the direction of a "build-as-you-go" strategy. Plus, point “9” tells us that the end date is negotiable — which means we will build iteratively until the product is ready to be launched and we will continue to add more functionality as we go, even post the first launch. This is a classic case of Agile development strategy.

In this segment, you learnt 12 Agile principles that guide developers to remain efficient and produce quality software. These 12 principles are —

1. Customer satisfaction
2. Welcoming change
3. Working software
4. Collaboration
5. Motivation
6. Face-to-face interactions
7. Measuring progress through working software
8. Constant pace
9. Continuous attention
10. Simplicity
11. Self-organised teams
12. Reviewing work regularly

**Measuring Progress in Agile**

Why is measuring progress based on the number of functionalities delivered in an iteration a better approach than measuring it based on the number of written lines of code?

Top of Form



**Progress measured based on the number of functionalities that are present in the working software is a more objective approach to measure how much work is done.**

**Feedback :**

*Progress measured based on the number of functionalities that are present in the working software provides more objective evidence of the work done than, say, other metrics such as the number of lines of code written.*

Bottom of Form

**Transparency in Agile**

Which of the principles in the Agile methodology take care of transparency issues between clients and developers?

Top of Form



**Working software and collaboration**

**Feedback :**

*The working software principle of the Agile methodology teaches you to always measure progress based on the working number of functionalities in the developed software. Earlier, when the progress was told in terms of the number of lines of code written, business people weren’t able to tell which stage software development had reached. Also, constant collaboration can promote better clarity and alignment between developers and business people. Thus leaving fewer things hidden and reduces project risks.*

Bottom of Form

# Values and Practices of Extreme Programming

In the previous segment, you learnt what extreme programming is and how it is related to Agile. In this segment, you will learn about the values and practices of extreme programming.



So, in this segment, you learnt about the —

1. Values of extreme programming
   * Communication
   * Simplicity
   * Feedback
   * Courage
   * Respect
2. Practices of extreme programming
   * User stories
   * Planning game
   * Short releases
   * Metaphor
   * Simple design
   * Testing
   * Refactoring
   * Pair programming
   * Collective ownership
   * Continuous integration
   * 40-hour week
   * On-site customer
   * Coding standard

**Welcoming Change**

Which of the following statements doesn’t support the extreme programming value?

Top of Form



**Code should be written for all the possible future functionalities that might come.**

**Feedback :**

*Extreme programming welcomes changes, and it encourages developers to write the minimum required code to fulfil needs.*

Bottom of Form