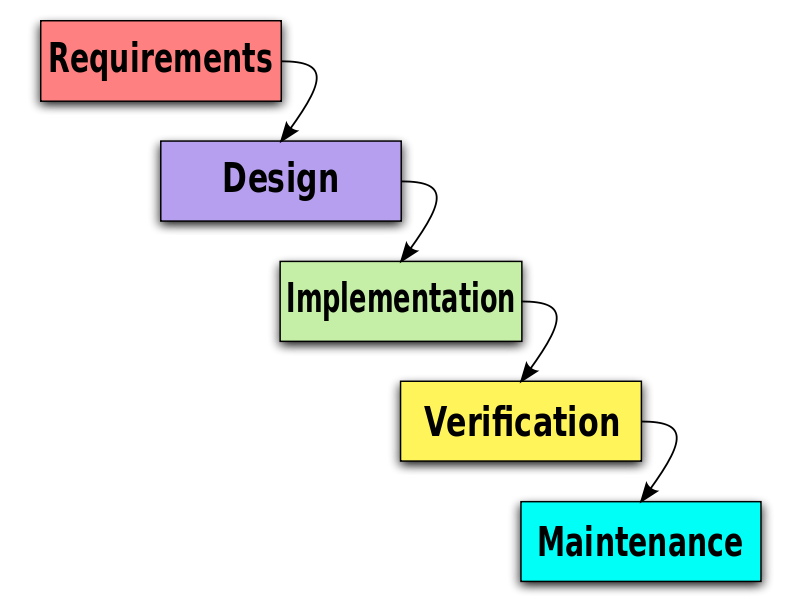
**Software Development Methodologies**

1. Waterfall Model
2. Give a description of each model
3. Give Pros and Cons of each model
4. Compare one model to another one
5. Spiral Model
6. Give a description of each model
7. Give Pros and Cons of each model
8. Compare one model to another one
9. Prototype Model
10. Give a description of each model
11. Give Pros and Cons of each model
12. Compare one model to another one
13. Incremental Model
14. Give a description of each model
15. Give Pros and Cons of each model
16. Compare one model to another one

**Design an application:**

* Requirements
* Algorithm
* Pseudo Code
* Flowchart
* Inputs
* Outputs
* Testing

1. **Waterfall Model**

Waterfall model is one of the most used software development methodologies, it is mostly used when developing a program that is dangerous or very important mostly made by major companies and government projects. Waterfall model is broken down into five important steps which are: Requirements, High Level Design, Coding, Testing and finally Maintenance. This structure is very helpful when working with an intense or difficult scenario which ensures maximum level of efficiency and productivity if the steps are followed thoroughly. The first step is requirements, in this step you go to your clients to collect feedback such as pros and cons of a current application or wants and needs from a completely new program they would want. With this step all the intel provided by the clients is put together, analysed, and documented for the next step. The second step is High Level Design which is a very crucial step in development because it is the whole foundation of the program. Once all the previous information is documented the initial start of development slowly begins as well as the planning and solving of problems encountered currently. In this part of development, algorithms are being looked at to select the most appropriate one for the current scenario. After the plan is accumulated the third step takes place which is coding which is simple as long as your plan is sawed after and carefully made. Penultimately after the code is written and there’s a rough prototype of a program the Testing begins to patch any bugs or glitches with updates. Ultimately after all the glitches are looked at a patched to a good enough extent Maintenance takes place ensuring that the program is operating in a stable enough condition which can be reliable for the client to use frequently.

With every design model there’s pros and cons to them. One advantage of using the waterfall model is that you take time to focus on the plan itself rather than going into a prototype straight away. Another advantage of using this way of planning can be further simplifying the task gave at hand which makes everything easier to digest and understand. On the other hand, there could be some disadvantages such as the cost, with this way of planning there is more costs of interviewing and more money spent on research and planning rather than the coding itself. Another disadvantage of using this model can be the fact that its can be harder to work on feedback after the final build is finished.

One model that is like this can be the spiral model they share the same amount of quality and starting steps. For example, things such as looking for the requirements and maintaining the final program. One thing that is different is that the waterfall model is more expensive than the spiral model.

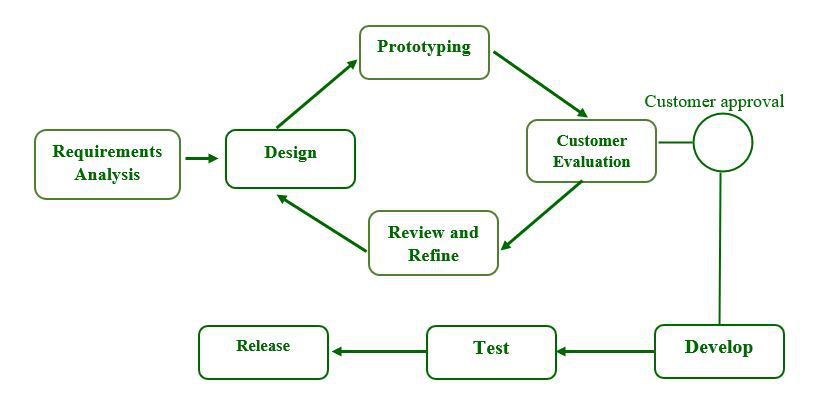
1. **Spiral Model**

The spiral model is a Software Development Methodology that combines techniques from other software development methodology that are top to bottom and vice versa. The spiral model consists of assessing risk and minimising it, this can be done by sectioning each part of the software into smaller segments to make it easier to digest and understand. The first phase of the spiral model is planning, this consists of getting a rough idea of what the software is going to be or what its intents to do. After that the second phase takes place which is the risk analysis which is when you ensure that there isn’t any dangers or risk when developing your program. After that the third phase takes place which is development which is when the program is has started it’s making process. Finally, the step to finish it is the evaluation which is when you take a step back to see what you have built and how you can move forward.

With each software development methodology there are pros and cons and with spiral model there are a few. For example, one disadvantage of the spiral model can be that it can be costly to go through all the steps with a good enough approach and focus on each thing especially when there’s more time spent researching and brainstorming instead of coding all the time. Another disadvantage can be that it is time consuming to do so when there’s a small deadline this approach shouldn’t be used. On the other hand, there could be advantages as well such as the fact that it oversimplifies projects that seem more difficult given at hand. Another advantage is how simple it is to implement spiral model into any project that involves developing and engineering.

One model that is like the spiral model can be the incremental model because you also break down the problem into smaller segments to further simplify matters. For example, when you are coding with the spiral model in mind, you’re likely to look at the current problem before you go to the next which is very unlike the prototype model. One thing that is different is that the spiral model uses plans to operate.

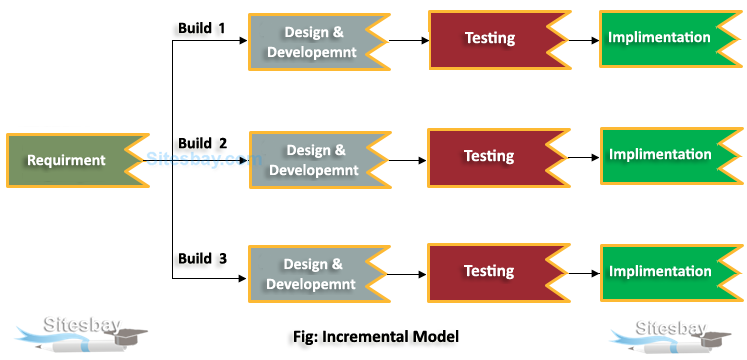
1. **Prototype model**

Prototype model is another software development methodology which is commonly used when developing software using a rough sketch and to get into coding straightaway excluding the need of a plan. The prototype model consists of starting coding as the first thing of a project, the issue with this can be that it’s harder to do this with more complicated software applications and is mostly applicable to simpler or easier software.

With every model there are advantages and disadvantages. For example, one advantage of using the prototype model can be that it is very time efficient meaning that you can come up with a prototype program for testing very quickly due to how fast it can be to code software. Another advantage of using prototype model can be cost, since all you’re doing is coding there’s less money spent on research and planning which means all the money can just go into coding which can be good for small organisations, On the other hand there could be some disadvantages to this way of modelling. For example, this way of development can’t be used on more complicated projects or software that is more important than others, so it needs to work. Another disadvantage of using this can be that it’s not stable to use and is mostly used to just to see how the concept works and you will most likely code something similar again.

One model that is like the prototype model can be the waterfall model because it just about uses everything that the waterfall model does but more in a rough sketch version to get things done quicker. For example, things such as requirements, testing and all the other tools used in the waterfall model. One thing that is different is that there isn’t as much risk analysis when using the prototype model compared to the waterfall model.

1. **Incremental Model**

Incremental model is a software development methodology that performs like the waterfall model. The basic requirement for this is just the core product and a couple features. Incremental model uses things like planning, risk analysis, coding, testing and maintenance. This is mostly used for important software and when used it is broken down to several sections to complete the task set one by one. When using the incremental model, you make several builds to create a software build that is stable enough to use in the final product

With every model there are advantages and disadvantages. For example, one advantage of using the incremental model can be that it can be very fast to code software since there’s not a lot of planning done but just coding instantly. Another advantage can be the fact that you come up with multiple builds to which you can choose which one is best or take elements from each build to make the best build. One disadvantage is cost, this model is the most expensive model to use and is even more expensive than the waterfall model which makes it difficult to use this type of model. Another disadvantage can be that there needs to be a good plan and a deep understanding of the topic since everything needs to be understood before a plan is written.

One model that is like the incremental model can be the waterfall model because they’re both reusable and can use the same code for another program if you change it up a bit, you’ll be able to use it again. One thing that is different from these two models are that the incremental model is flexible to change to whereas the waterfall model can’t be flexible.

**Requirements:**

1. Number of Players
2. How many in a group
3. Placement in Each event
4. Team Name
5. Number of Points
6. Type of event participated
7. Individual or Group
8. Rules

**Algorithm:**

1. The application will start with a welcome message to greet the user. The message could be “Good luck in Odysseas Quiz”
2. The application will inform the user what options and possibilities exist such as:
   1. This application can accept up to 20 individuals to join the competition
   2. Or 4 tams with 5 members can join the competition
   3. The application will allocate different point for individuals or teams such as:
      1. For individual/s the highest grade will be….
      2. For the next place (i.e., second place will be 10 minus from the highest grade)
   4. The application will show the possible events that are available such as:
      1. Film
      2. History
      3. Gaming
   5. The application will then ask the user if they are in an individual or in a team
   6. Then the application will ask the users that are in a team how many members are there in the team
      1. If they are in a team the application will not let there be more than 5 members.
   7. The application will then ask the team what their team’s name is
   8. Then the application will ask what the names of each team member is
   9. The application will ask the user how many points they have allocated throughout the tournament
3. 1st Place will need to have 95+/100
4. 2nd Place will need to have 85+/100

**Pseudo Code:**

Output: Welcome

Output: Introduction

Output: Types of Events

Input: Type of event to participate in

Input: Group or Individual

Output: Available Spaces

Input: Number of Group Members

Output: Validation (Error or Continue)

Input: Group Member Names

Input: Number of Points Allocated

Output: Validation

Output: Placement in the tournament

Output: Thank you Message

**Inputs:**

Input: Type of event to participate in

Input: Group or Individual

Input: Number of Group Members

Input: Group Member Names

Input: Number of Points Allocated

**Outputs:**

Output: Welcome

Output: Introduction

Output: Types of Events

Output: Available Spaces

Output: Validation (Error or Continue)

Output: Validation

Output: Placement in the tournament

Output: Thank you Message

**Flowchart:**

Welcome/ Intro

End Message

Number of Points Allocated

Available Spaces

Choose Event

Types of Events

Number of Members

Member Names

Group or Individual

Ranking

Individual Name