**Assignment 1**

**Software Engineering Concepts :**

**Section A:**

**Questions**

**Note : Explain with examples**

1. What are the reasons of a successful and unsuccessful software project?
2. What types of problems may arise if a software project is developed on ad hoc basis?
3. Provide three examples of software projects that would be amenable to the waterfall model. Be specific.
4. Provide three examples of software projects that would be amenable to the prototyping model. Be specific.
5. What process adaptations are required if the prototype will evolve into a delivery system or product?
6. Provide three examples of software projects that would be amenable to the incremental model. Be specific.
7. As you move outward along the spiral process flow, what can you say about the software that is being developed or maintained?
8. What is a Product & Service based company?
9. What is a Process framework & framework activities
10. What are principles of software engineering?

**Assignment 1**

**Section B:**

**Aim : To study Software Development Life Cycle ( SDLC) Models**

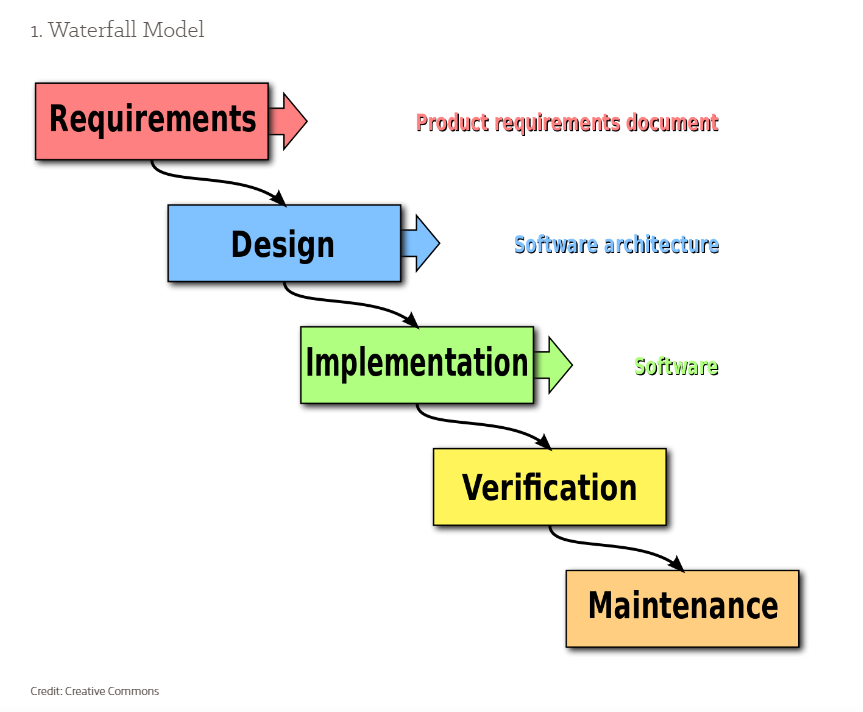
**1. Team 9 KH - CodeAstra**

**2.** **Team Member’s Name**

* 082-Prashant Chaudhary
* 083-Prathamesh Patidar
* 084-Pratik Pawar
* 085-Pratik Raut
* 086-Pratiksha Barge
* 087-Priyadarshini Meher
* 088-Priyanka Pawar
* 089-Priyanka Mahajan
* 090-Priyanka Aher
* 091-Rahul Darade
* 092-Rahul Somvanshi

**3. Model**

**3.1 Waterfall Model**

* The [**Waterfall Model**](http://www.umsl.edu/~hugheyd/is6840/waterfall.html) is a linear sequential flow, often used with projects that have a defined set of requirements.
* The model’s process flows steadily downwards through the phases of software implementation.
* This model relies on the completion of the previous phase for the following to begin.
* The Waterfall Model should be used with projects that do not anticipate unforeseen changes mid-development.
* This approach was one of the earliest models used for software development.

**Advantages**:

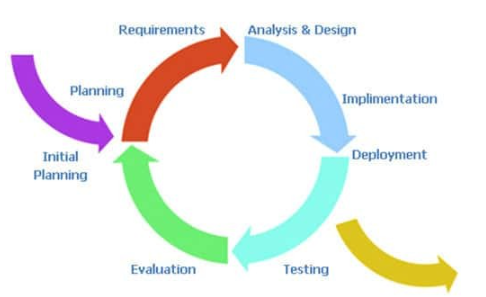
* Simple and understandable, the Waterfall Model is a manageable method ideal for lifecycle management of smaller projects where the requirements are established and finalized upfront.

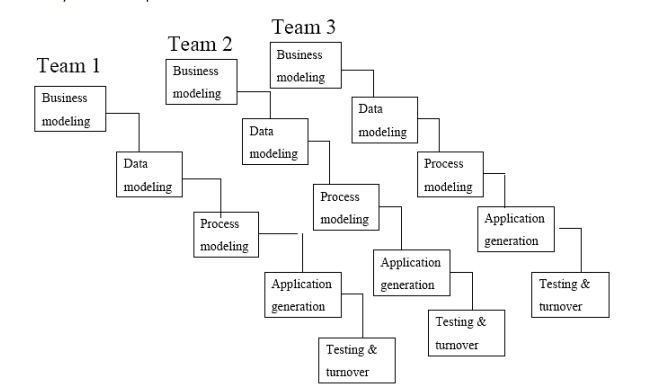
**Disadvantages**:

* Because of its rigid structure, the Waterfall Model does not work well for complex projects where there is a chance of a change in requirements and/or significant impromptu testing throughout the software development stage.

**3.2 Rapid Application Development**

* Short for Rapid Application Development, the [RAD Model](http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/) is a modification of the Incremental Model.
* When implementing this model, several components are developed simultaneously as if they were smaller, individual projects.
* The different components are then assembled into working prototypes.

****

****

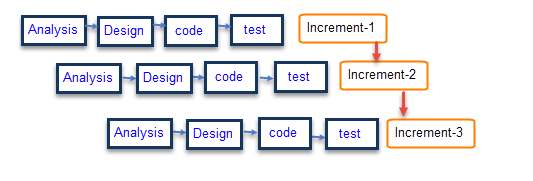
**Advantages**:

* The RAD Model allows for reduced development time and allows for more customer feedback throughout the software development.

**Disadvantages**:

* The applicability of the RAD Model is limited, as the project needs to be easily modularized into several increments.
* It also requires highly-versed developers as well as excellent modeling and planning skills.
* Issues with the final assembly of components could result in unforeseen setbacks and the redeveloping of components to properly fit the rest.

**3.3 Incremental Model**

* Built to mediate the shortcomings of the Waterfall Model, the [Incremental Model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/).
* Consists of iterative and incremental development stages.
* The Incremental Model is essentially comprised of several mini-Waterfall cycles.
* ****This model divides the development into small sections/stages and it can allow the software developers to take advantage of learnings and insights gleaned from earlier development stages.

**Advantages**:

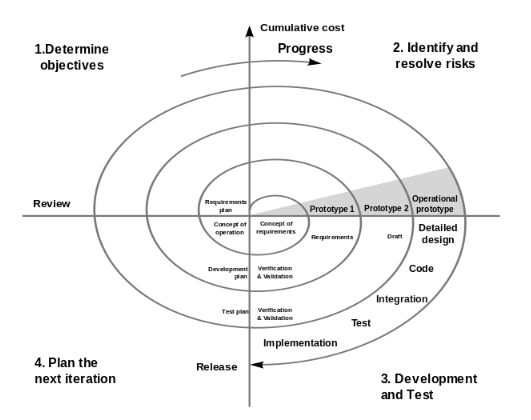
* The Incremental Model is a great solution for projects that need accommodation for some change requests between increments.
* This model also yields the benefit of being able to detect problems earlier in the software development for better lifecycle management planning.

**Disadvantages**:

* A potential disadvantage to the Incremental Model is the need for strategic planning and documentation.
* This method also tends to require more resources, staff and monetary, behind the project.
* This model isn’t ideal for ongoing development as the next sequence cannot begin until the previous stage has fully completed.

**3.4 Spiral Model**

* The [Spiral Model](http://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm) combines elements of both the Iterative and Waterfall development models, in efforts to combine advantages of top-down and bottom-up production.
* The Spiral Model has four phases – Identification, Design, Construct/Build, Evaluation and Risk Analysis.
* The software project repeatedly passes through these phases in iterations called spirals.

****

**Advantages:**

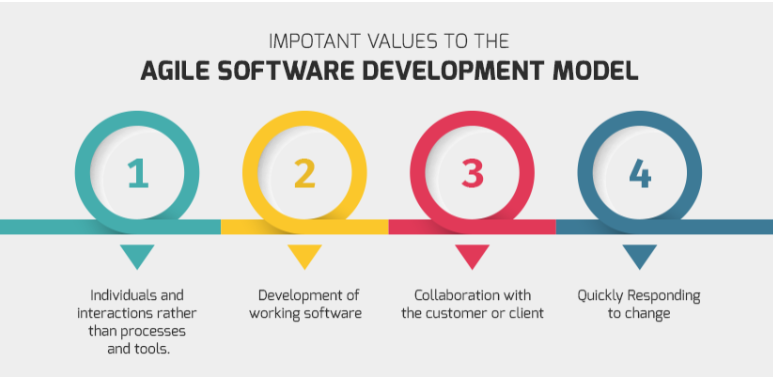
* The Spiral Model can be advantageous as it manages risks and divides development into phases.
* It also helps with more accurate estimates for budget and schedule as roadblocks are discovered earlier.

**Disadvantages:**

* Since this model is highly customized, repurposing the process can be confusing.
* It also requires team members that are well-versed in risk evaluation.

**3.5 Agile Model**

* This model is rooted in process adaptability and user engagement with rapid delivery of functioning software components.
* It has both Iterative and Incremental features, as it breaks the product development into small incremental builds that are then provided in iterations.
* The [Agile Model](http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm) relies on heavy collaboration between cross-functional teams and the customer/user.

****

**Advantages**:

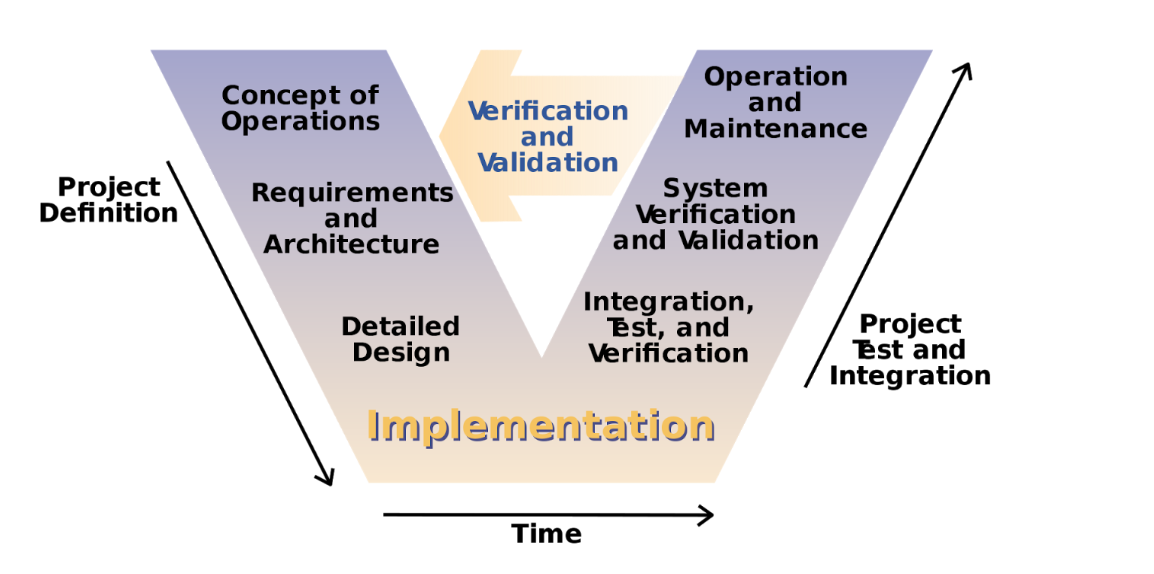
* The Agile Model decreases the amount of time to yield individual system features.
* It also calls for a lot of communication and continuous feedback from the customer/user that can provide clear direction for the project.

**Disadvantages:**

* The Agile method can potentially veer-off track as it relies on end-user interaction that may or may not be clearly expressed.
* Documentation is also minimal for an Agile software development strategy and requires a well-versed, cross-functional team.

**3.6 ‘V’ shaped model**

* An extension of the Waterfall Model, the [V-Model](http://www.tutorialspoint.com/sdlc/sdlc_v_model.htm) also functions as a sequential flow.
* However, instead of only moving linearly downward, the software development lifecycle bends upwards after the coding (for each of the testing phases).
* For every phase in the downward sequence, there is a corresponding testing phase in the following upwards sequence.
* This model is used for projects where software requirements and tools are known upfront.

****

**Advantages**:

* The V-Model is a simple process that’s great for smaller projects.
* Using the V-Model can yield a higher chance of success due to the test plans of the development stage and regularly schedule updates throughout its lifecycle.

**Disadvantages**:

* Similar to the Waterfall Model, the V-Model is very rigid in nature so it isn’t ideal for applications or systems software that may require unforeseen changes/updates throughout the software lifecycle.

**4. Comparison between the models**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Properties of Model | Water-Fall Model | Incremental Model | Spiral Model | Rad Model |
| Planning in early stage | Yes | Yes | Yes | No |
| Returning to an earlier phase | No | Yes | Yes | Yes |
| Handle Large-Project | Not Appropriate | Not Appropriate | Appropriate | Not Appropriate |
| Detailed Documentation | Necessary | Yes but not much | Yes | Limited |
| Cost | Low | Low | Expensive | Low |
| Requirement Specifications | Beginning | Beginning | Beginning | Time boxed release |
| Flexibility to change | Difficult | Easy | Easy | Easy |
| User Involvement | Only at beginning | Intermediate | High | Only at the beginning |
| Maintenance | Least | Promotes Maintainability | Typical | Easily Maintained |
| Duration | Long | Very long | Long | Short |
| Risk Involvement | High | Low | Medium to high risk | Low |
| Framework Type | Linear | Linear + Iterative | Linear + Iterative | Linear |
| Testing | After completion of coding phase | After every iteration | At the end of the engineering phase | After completion of coding |
| Overlapping Phases | No | Yes (As parallel development is there) | No | Yes |
| Maintenance | Least Maintainable | Maintainable | Yes | Easily Maintainable |
| Re-usability | Least possible | To some extent | To some extent | Yes |
| Time-Frame | Very Long | Long | Long | Short |
| Working software availability | At the end of the life-cycle | At the end of every iteration | At the end of every iteration | At the end of the life cycle |
| Objective | High Assurance | Rapid Development | High Assurance | Rapid development |
| Team size | Large Team | Not Large Team | Large Team | Small Team |
| Customer control over administrator | Very Low | Yes | Yes | Yes |