



ANJUMAN-I-ISLAM'S KALSEKAR TECHNICAL CAMPUS School of Engineering & Technology

Affiliated to : University of Mumbai, Recognised by : DTE (Maharashtra) & Approved by : AICTE (New Delhi)

CourseCode:

Course Name:

Class:

Batch:

Roll no :

Name :

Experiment : 01

Aim: Prepare detailed statement of problem for the allotted miniproject and identify the suitable process module for the same with justification.

Theory :

Explain the following process modules with advantages and disadvantages.

- Waterfall Model
- Prototyping Model
- Incremental Model
- Spiral Model
- Agile

Reference:

1. Smart India Hackathon 2019 - <https://www.sih.gov.in/sih2019>
2. Software Engineering Process Models - <https://bit.ly/2OknVZI>

Conclusion:

#Theory

1. Waterfall model

It's useful when the requirements are clear, or following a very structured process as in critical systems which needs a detailed, precise, and accurate documents describes the system to be produced. Not good when requirements are ambiguous, and doesn't support frequent interaction with the customers for feedback and proposing changes. It's not suitable for large projects that might take long time to be developed and delivered.

Advantages

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.

Disadvantages

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.

2. Prototype

Again, it's an early sample, or release of a product built to test a concept or to act as a thing to be replicated or learned from. This is very useful when requirements aren't clear, and the interactions with the customer and experimenting an initial version of the software results in high satisfaction and a clearance of what to be implemented. Its downsides are, good tools need to be acquired for quick development (like coding) in order to complete a prototype. In addition, the costs for training the development team on prototyping may be high.

Advantages

- Users are actively involved in the development
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.
- Quicker user feedback is available leading to better solutions.

Disadvantages

- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Incomplete application may cause application not to be used as the full system was designed Incomplete or inadequate problem analysis.

3. Incremental & Iterative

They're suited for large projects, less expensive to the change of requirements as they support customer interactions with each increment. Initial versions of the software are produced early, which facilitates customer evaluation and feedback. They don't fit into small projects, or projects that waterfall are best suited for; A structured process with a detailed, and accurate description of the system. In incremental we deploy system module wise whereas in iterative whole system is developed and deployed portion wise.

Advantages (Incremental)

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.

Advantages (Iterative)

- In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In iterative model less time is spent on documenting and more time is given for designing.
- In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.

Disadvantages (Incremental)

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than waterfall.

Disadvantages (Iterative)

- Each phase of an iteration is rigid with no overlaps
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

4. Spiral

It's good for high risky or large projects where the requirements are ambiguous. The risks might be due to cost, schedule, performance, user interfaces, etc. Risk analysis requires highly specific expertise, and project's success is highly dependent on the risk analysis phase. It doesn't work well for smaller projects.

Advantages

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date.

Disadvantages

- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project's success is highly dependent on the risk analysis phase.
- Doesn't work well for smaller projects.

5. Agile

It suits small-medium size project, with rapidly changes in the requirements as customer is involved during each phase. Very limited planning is required to get started with the project. It helps the company in saving time and money (as result of customer physical interaction in each phase). The daily meetings make it possible to Difficult to scale up to large projects where documentation is essential. A highly skilled team is also needed. If team members aren't committed, the project will either never complete or fail. And there's always a limitation in time, like in increments, meetings, etc.

Advantages

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.

Disadvantages

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

#Problem Statement

Aspirants/Students can be assessed on different fields using this website. Fields can be Reasoning, Aptitude, Technical MCQ, Coding etc. This will be an online platform with questions at different levels (Difficulty). Candidate would start at assessment with a medium difficulty question and depending on its response, the platform should decide the next level of questions to be shown (level would increase or decrease as per response). Each question would be assigned a weight age and time duration

#Process Model

Incremental Model - The product comprises of developing an assessment system such as MCQ and based on users' responses the difficulty of the assessment increases.

So incremental model would be suitable here as deploying system modules by modules and then creating the system as a whole.