

PRACTICAL-1

AIM: List & draw at least 7 various Software Development Life Cycle (SDLC) models and preparing the detailed case study on “JobHunt” that which SDLC model is suitable to develop the “JobHunt”?

❖ SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

➤ Here, are some of the most important models of Software Development Life Cycle (SDLC):

1. Waterfall Model
2. Incremental Model
3. Prototyping Model
4. Agile Model
5. Spiral Model
6. Iterative Model
7. RAD Model

1. Waterfall Model

- The waterfall is a widely accepted SDLC model. In this approach, the whole process of the software development is divided into various phases of SDLC. In this SDLC model, the outcome of one phase acts as the input for the next phase.
- This SDLC model is documentation-intensive, with earlier phases documenting what need to be performed in the subsequent phases.

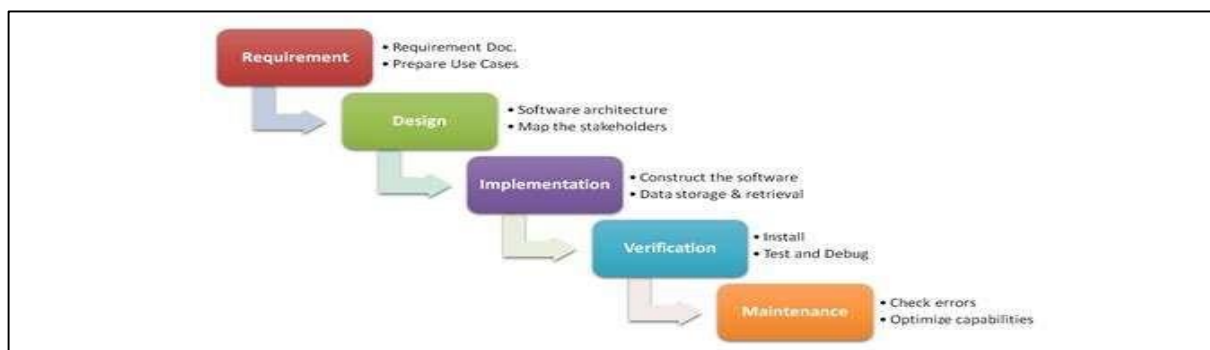


Figure 1.1 Waterfall Model

Advantages:

1. It uses a clear structure.
2. The progression of the waterfall model is intuitive.
3. The waterfall model determines the end goal early.
4. There are fewer financial surprises with the waterfall method.
5. It reinforces good testing habits.

Disadvantages:

1. The waterfall model doesn't support making changes.
2. It can invalidate the work you've previously accomplished.
3. This method excludes end-users and clients.
4. It delays testing until after the completion of the project.
6. It typically works better for large projects.

Applications:

1. Requirements are very well documented, clear and fixed.
2. Product definition is stable.
3. Technology is understood and is not dynamic.
4. There are no ambiguous requirements.

2. Incremental Model

- The incremental model is not a separate model. It is essentially a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software.
- The SDLC life cycle process is repeated, with each release adding more functionality until all requirements are met. In this method, every cycle act as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

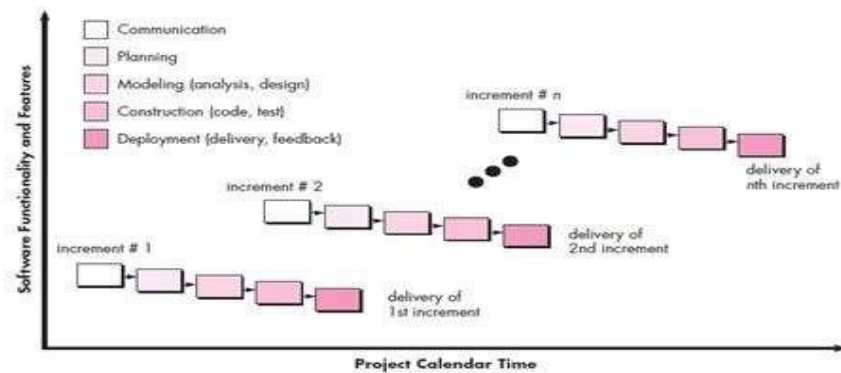


Figure 1.2 Incremental Process

Advantages:

1. Prepares the software fast.
2. Clients have a clear idea of the project.
3. Changes are easy to implement.
4. Provides risk handling support, because of its iterations.

Disadvantages:

1. A good team and proper planned execution is required.
2. Because of its continuous iterations the cost increases.

Applications:

1. When the requirements are superior.
2. A project has a lengthy development schedule.
3. When Software team are not very well skilled or trained.
4. When the customer demands a quick release of the product.
5. You can develop prioritized requirements first.

3. Prototyping Model

- The Prototyping Model is one of the most popularly used Software Development Life Cycle Models .
- This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved

which forms the basis for developing the final product. In this process model, the system is partially implemented before or during the analysis phase thereby giving the customers an opportunity to see the product early in the life cycle. The process starts by interviewing the customers and developing the incomplete high-level paper model. This document is used to build the initial prototype supporting only the basic functionality as desired by the customer.

- Once the customer figures out the problems, the prototype is further refined to eliminate them. The process continues until the user approves the prototype and finds the working model to be satisfactory.

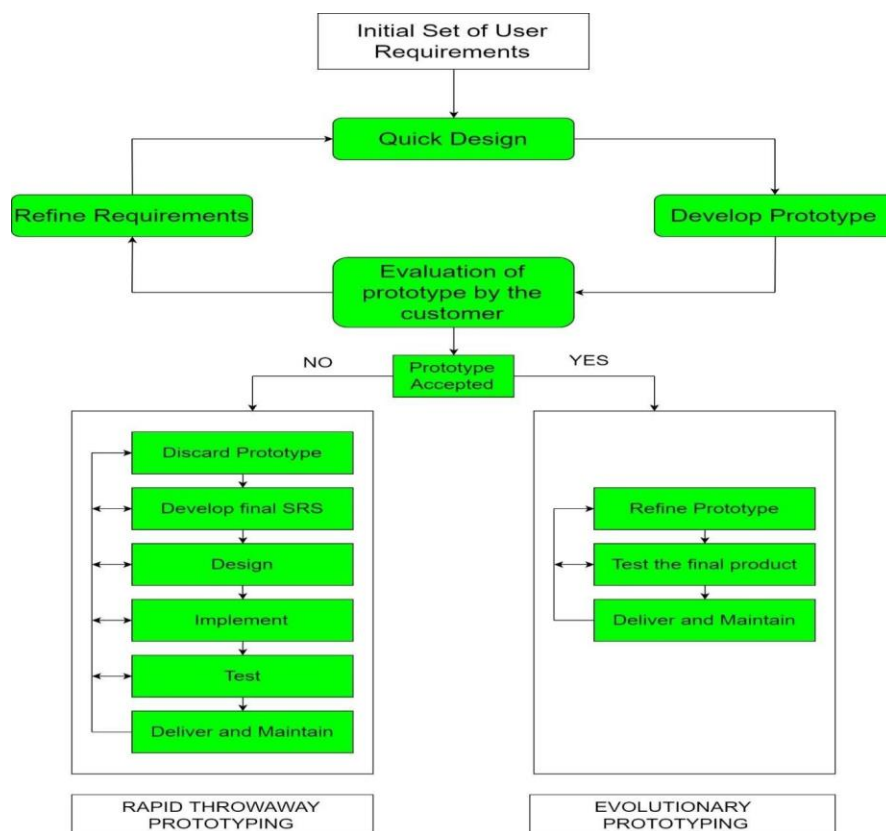


Figure 1.3 Prototype Process

Advantages:

1. The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
2. New requirements can be easily accommodated as there is scope for refinement.
3. Missing functionalities can be easily figured out.
4. Flexibility in design.

Disadvantages:

1. Costly w.r.t time as well as money.
2. There may be too much variation in requirements each time the prototype is evaluated by the customer.
3. Poor Documentation due to continuously changing customer requirements.
4. It is very difficult for developers to accommodate all the changes demanded by the customer.

Applications:

1. The Prototyping Model should be used when the requirements of the product are not clearly understood or are unstable.
2. It can also be used if requirements are changing quickly.
3. This model can be successfully used for developing user interfaces, high technology softwareintensive systems, and systems with complex algorithms and interfaces.
4. It is also a very good choice to demonstrate the technical feasibility of the product.

4. Agile Model

- Agile methodology is a practice which promotes continue interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.



Figure 1.4 Agile Process

Advantages:

1. Customer satisfaction by rapid, continuous delivery of useful software.
2. Working software is delivered frequently (weeks rather than months).
3. Face-to-face conversation is the best form of communication.
4. Even late changes in requirements are welcomed.

Disadvantages:

1. 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.
2. There is lack of emphasis on necessary designing and documentation.
3. The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
4. 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.

Application:

1. When frequent changes are required.
2. Face-to-Face Communication with clients.
3. Efficient design and fulfils the business requirement.

5. Spiral Model

- The spiral model is a risk-driven process model. This SDLC testing model helps the team to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc.
- This model adopts the best features of the prototyping model and the waterfall model. The spiral methodology is a combination of rapid prototyping and concurrency in design and development activities.

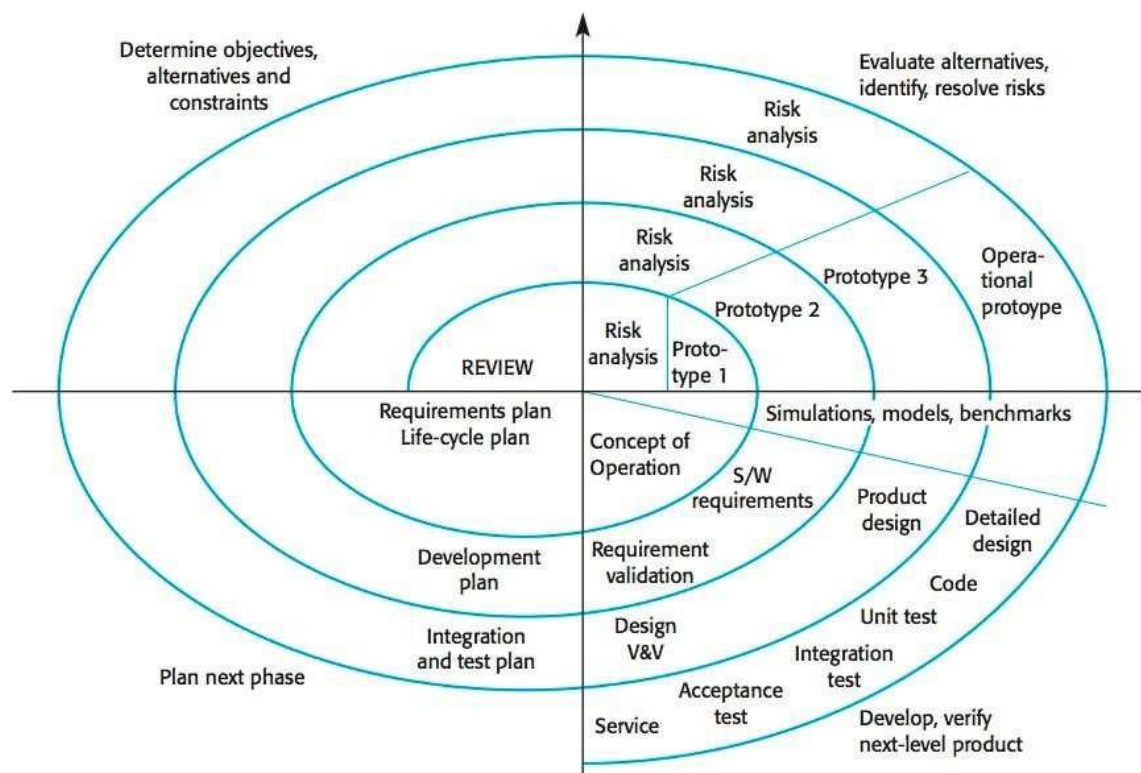


Figure 1.5 Spiral Process

Advantages:

1. Risk handling
2. Good for large projects
3. Flexibility in requirements
4. Customer satisfaction

Disadvantages:

1. Complex
2. Expensive
3. Too much dependability on risk analysis
4. Difficulty in time management

Applications:

1. When there is a budget constraint and risk evaluation is important.
2. For medium to high-risk projects.
3. Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
4. Customer is not sure of their requirements which is usually the case.

6. Iterative Model

- In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.
- An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.



Figure 1.6 Iterative Process

Advantages:

1. Some working functionality can be developed quickly and early in the life cycle.
2. Results are obtained early and periodically.
3. Parallel development can be planned.
4. Progress can be measured.
5. Less costly to change the scope/requirements.
6. Testing and debugging during smaller iteration is easy.

Disadvantages:

1. More resources may be required.
2. Although cost of change is lesser, but it is not very suitable for changing requirements.
3. More management attention is required.
4. System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle.
5. Defining increments may require definition of the complete system.
6. Not suitable for smaller projects.
7. Management complexity is more.

7. Rapid Application Development(RAD) Model

- RAD or Rapid Application Development process is an adoption of the waterfall model; it targets developing software in a short period. The RAD model is based on the concept that a better system can be developed in lesser time by using focus groups to gather system requirements.
- Business Modeling
- Data Modeling
- Process Modeling
- Application Generation

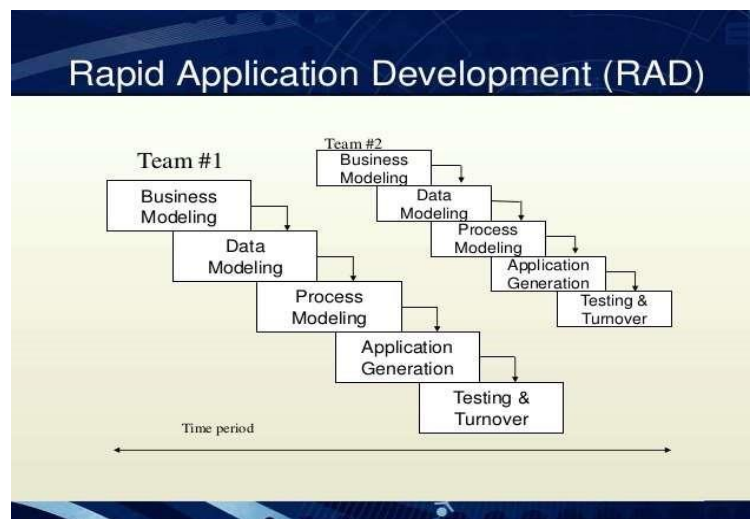


Fig 1.7 RAD Process

Advantages:

1. The use of reusable components helps to reduce the cycle time of the project.
2. Feedback from the customer is available at the initial stages.
3. Reduced costs as fewer developers are required.
4. The use of powerful development tools results in better quality products in comparatively shorter time spans.

Disadvantages:

1. The use of powerful and efficient tools requires highly skilled professionals.
2. The absence of reusable components can lead to the failure of the project.
3. The systems which cannot be modularized suitably cannot use this model.
4. It is not meant for small-scale projects as in such cases, the cost of using automated tools and techniques may exceed the entire budget of the project.

Applications:

1. This model should be used for a system with known requirements and requiring a short development time.
2. The model can also be used when already existing system components can be used in developing a new system with minimum changes.

- **Software Development Life Cycle Diagram:**

5 Core Stages of Software Development

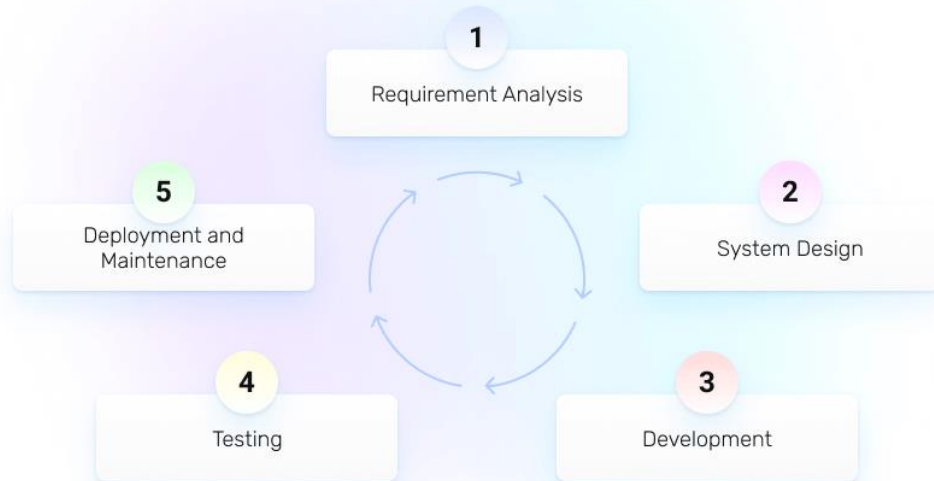


Fig: 1.1 – Software Development Life Cycle

CASE STUDY: JOBHUNT

Overview:

It is an online job portal website named “JobHunt”. Start-up companies who are looking for employees can register themselves to our website, can specify their requirements and describe their way of hiring and other details. On the other hand, those who are looking for a job can post their resume and approach companies for job. It is open source website and also platform independent website. This website make more easy to hiring process. The structure of the website is not so much large but the features of the site are much more important than the structure.

Requirements:

1. Ragistration for company and sign up and sign in for job seeker
2. Profiles of both the parties i.e. job seeker and job provider
3. Sorting candidates by profile
4. Chat with HR
5. Field wise filtration
6. Coding round facility also available
7. 100% secure video interviews
8. Direct and quick notifications
9. Verification process
10. Privacy protection
11. Quick hire
12. Placement satistics
13. Resume maker for freshers/resume building sessions
14. Location wise filtering

Suitable model for development of JobHunt:

Selecting accurate model for developing of the software invention or request is very significant. Founded on the model the expansion and testing processes are accepted out. As Waterfall model is more traditional and easy to gather requirements and analyzing system, so we choose the Waterfall model according to the JobHunt website . This technique works well for big projects that may take numerous to progress.

Here in case of JobHunt website definition is stable and also requirements are very well documented, clear and fixed. The project is short. Ample resources with required expertise are available to support the product. Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

So in case of JobHunt website development first of all we are gathering requirements and analysis our requirements and which requirements are more important so we give priority to this requirements. After that we move on of System Design which helps in specifying hardware and system requirements and also helps in defining overall system architecture. After that in implementation part With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. Whenever implementation part is over we are move on testing and debugging part. In this phase we are finding failure and some bugs of our system. Once the functional and non-functional testing done the product is deployed in the customer environment or released into the market. There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment. So all the point of view in our case waterfall model is used to develop JobHunt type website.

Advantages:

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Disadvantages:

- 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.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.
So risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing Requirements.

Conclusion: From the above theory and case study, Waterfall Model suits best for the development of “JobHunt” website .

References:

- [1] Waterfall Model: <https://labs.sogeti.com/sometimes-its-ok-to-waterfall>
- [2] Incremental Process Model: <http://softwareengineeringmca.blogspot.com/2016/10/>
- [3] Prototyping Model: <https://media.geeksforgeeks.org/wp>
- [4] Agile Model: <https://miro.medium.com/max/>
- [5] Spiral Model: <https://cdn-images-1.medium.com/max>

[6] Iterative Model: <https://bit.ly/3aGEWNC>

[7] Rapid application development model: <http://4.bp.blogspot.com/-JAp7lbWATs4/Va84VdiZWZI/AAAAAAAAAFk/>

Remarks :_____.

Marks :_____.

Signature:_____.