SDLC MODEL DOCUMENT

CS GROUP NO -02

UNRAVEL

<Version 1.0>

Author - Bhavana Kurra

Table of Contents -

0. Purpose

1. CLASSICAL WATERFALL MODEL

  1.1 INTRODUCTION:

  1.2 CHOOSING CRITERIA:

2. PROTOTYPE MODEL

  2.1 INTRODUCTION:

  2.2 REJECTION CRITERIA FOR PROTOTYPE MODEL:

3. ITERATIVE WATERFALL MODEL

  3.1 INTRODUCTION:

  3.2 REJECTION CRITERIA FOR ITERATIVE WATERFALL MODEL:

4. INCREMENTAL MODEL

4.1 INTRODUCTION:

  4.2 REJECTION CRITERIA FOR INCREMENTAL MODEL

5. SPIRAL MODEL

  5.1 INTRODUCTION:

  5.2 REJECTION CRITERIA FOR SPIRAL MODEL:

6. AGILE MODEL

  6.1 INTRODUCTION:

  6.2 REJECTION CRITERIA FOR AGILE MODEL:

7. REFERENCES

Purpose

Purpose of this document is to understand different SDLC model and based on that decide which models has to be followed in current ongoing project.

1. CLASSICAL WATERFALL MODEL

1.1 INTRODUCTION:

- Waterfall model is an example of Sequential model. In this model, the software development activity is divided into different phases and each phase consists of series of tasks and has different objectives.
- The Waterfall model is the classical model and it is the oldest and widely used by the government organization. In fact it was the first model which was widely used in the software industry .The number of step in the waterfall model is not defined.
- The waterfall is strictly sequential and uses a top-to-down approach, the earlier phases deal with requirements followed by design before coding or any other formal implementation is started consequently, the phases deal with test followed by production and maintenance of a product. Waterfall model is totally document focused model, this means each phase has a certain document or set of documents as its output, the documents of earlier phase serve as input of the next phase and the of the life cycle, we find an output, i.e product.
- An important approach of the waterfall model is that one phase has to be completed before the next phase can be started.

1.2 CHOOSING CRITERIA:

If there are less number of employees to work on the project, Classical Waterfall model is very useful to complete the project before the deadline. In case if the core product is well-defined and understood and the team has clarity about requirements then this model can be used.We know all the requirements but it takes a-bit more time to complete those requirements.But we have the clarity how the end product should be and there are no updates in the requirements unless and until we want to extend our project.As of know we are confining this project only to CS people. So, we have the clarity about the requirements and product.Hence,we used classical waterfall model.

2. PROTOTYPE MODEL

2.1 INTRODUCTION:

- Software prototyping refers to the activity of creating prototypes of software applications, i.e. incomplete versions of the software program is being developed. A prototype typically simulates only a few aspects of, and may be completely different from, the final product.
- The prototype does not always hold the exact logic, used in the actual software application and is an extra effort to be considered under effort estimation.
- Prototype model is generally used when the requirements are not clear. It is used when the customer is not clear about the details of the input, process and the output needs of the software.

2.2 REJECTION CRITERIA FOR PROTOTYPE MODEL:

- ❖ Users may get confused in the prototypes and actual systems.
- ❖ Risk of insufficient requirements analysis owing to too much dependency on prototype.

3. ITERATIVE WATERFALL MODEL

3.1 INTRODUCTION:

- Each release of Iterative Model is developed in a specific, fixed time period called iteration. Each iteration focuses on a certain set of requirements. Each cycle ends with a usable system, i.e. each iteration results in an executable release. Mini waterfall model is followed in each iteration.
- All high priority risks are taken in the first iteration so that risk at the end of the project is minimal. Since every iteration results in executable release, it enables early user feedback. Another advantage is that major risks are resolved in initial iterations.
- In 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 lifecycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just one part of the software, which is then reviewed in order 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.

3.2 **REJECTION CRITERIA FOR ITERATIVE WATERFALL MODEL**:

- ❖ Once an application comes at the testing phase, 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.

4. INCREMENTAL MODEL

4.1 **INTRODUCTION**:

Incremental model in software engineering is the one which combines the elements of waterfall model, which are then applied in an iterative manner. It basically delivers a series of releases, called increments which provide progressively more functionality for the client as each increment is delivered.

- In Incremental model of software engineering, waterfall model is repeatedly applied in each increment. The incremental model applies linear sequences in a required pattern as calendar time passes. Each linear sequence produces an increment in the work.
- Incremental model plays a significant role when the demand for early release of product arises or model more in use of web applications associated concerns.

4.2 **REJECTION CRITERIA FOR INCREMENTAL MODEL**

- Requirements of good planning and design

- Initial cost is lower but the it may exceed at the end

- Precise and complete definition of the system is required

- Rigid iteration which do not overlap each other

5. SPIRAL MODEL

5.1 INTRODUCTION:

Spiral model is a combination of iterative development process model and waterfall model with very high emphasis on risk analysis.
Spiral has four phases.
1. Identification: Identify the objective of product. This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals all the unit and system requirements are done.
2. Risk Analysis: It includes identifying and managing risks such as schedule slippage and cost overrun. A process is undertaken to identify risk and alternate solutions. If any risk is found during the risk analysis then alternate solutions are suggested and implemented.
3. Development: In the baseline spiral when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback. Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software is produced. Testing is done at the end of phase.
4. Evaluation phase: This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

5.2 REJECTION CRITERIA FOR SPIRAL MODEL:

- In this model project’s success is highly dependent on the risk analysis phase so it requires highly specific expertise. That we don’t have in our group.
- As our project is smaller one so we can not use this model for our project because it can be more costly for smaller projects.
- As in each spiral we have to do documentation so there will be excessive documentation in this model and management is also more complex.
- If there are some changes every time then there is a risk of running the spiral in indefinite loop.

6. AGILE MODEL

6.1 **INTRODUCTION**:

- Agile SDLC model is a combination of iterative and incremental process models with focus on customer satisfaction by rapid delivery of working software product. This is done by small incremental releases of software. Agile is adaptive approach and others like waterfall model is predictive approach.

- Each release is thoroughly tested to ensure software quality is maintained. Each release is incremental in terms of features and the final release holds all the features required by the customer. Each iteration involves planning, requirement gathering, design, coding, unit testing and acceptance testing.

- In Agile model, according to adaptive approach, there is no detailed planning.There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

- Customer interaction and minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration.

6.2 **REJECTION CRITERIA FOR AGILE MODEL**:

- ❖ There is very high individual dependency due to lack of documentation. But we are supposed to do a lot of documentation.

- ❖ Transfer of technology to new team members may be quite challenging due to lack of documentation. We don’t want that because we all are new.

- ❖ There is lack of emphasis on necessary designing and documentation.

- As we are supposed to give a proper product to customer but it depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

7. REFERENCES

1. https://www.tutorialspoint.com/sdlc/
2. http://istqbexamcertification.com/
3. http://www.technotrice.com/incremental-model-in-software-engineering/
4. https://melsatar.wordpress.com/2012/03/21/choosing-the-right-software-development-life-cycle-model/