SURVEY AND REVIEW ON SOFTWARE ENGINEERING MODELS

Shivani Ahuja[1], Pooja Bangar[2], Palak Keshwani[3]

[1][2]B.E. Student , CSE Dept, KITE ,Raipur

[3]Asst. Prpfessor , CSE Dept, KITE ,Raipur

E-mail : -*shivaniahuja.rpr16@gmail.com, palakeshwani@gmail.com*

ABSTRACT

Software engineering goal is to produce a quality software package which is delivered on time, within assigned budget, and by the demand & needs expected by customer but unluckily maximum times this aim is rarely achieved. Software engineering has a continuous growth in research in determining new methodology. The tools and methodologies that has a vast modernization in software maintenance and development to be much more dependable and effective. We will learn that the quality, cost, and schedule are primary forces that control the software project.

Keywords: *Waterfall model, Spiral model, RAD model, Prototype model.*

1. Introduction

Here we see the basic terms software, software engineering, software engineering models, their requirements, their designing, their testing and maintenance.

Software :

The term software in general is a set of instructions and programs instructing a computer for doing particular task or a project.

Software Engineering :

Software Engineering is elaborated study for engineering to designing, development and maintaining the software. The concept Software Engineering was introduced for addressing the issue of low quality software programs or project. Problems arise when a particular software generally exceeds timelines, budgets, and reduced levels of quality. It assure that the application is built constantly, correctly, on time and on budget and within under requirements.

Software Engineering Model :

The **software development or engineering models** are distinct processes or methodologies that are selected for the development of the project or application depending on the project's aims or goals. There are many development life cycle models that have always been developed in order to accomplish different needed objectives. The selection of model has very high impact on the testing that is carried out.

The phases of the software engineering models are the main criteria for developing any application project or program or model. They are as follows :

- Gathering the requirement for the particular type of model.
- Designing should be done in a type of blue print.
- Implementation process of design of the particular model.

- Testing process of that models are necessary.
- Maintenance is the main function after the testing of development model.

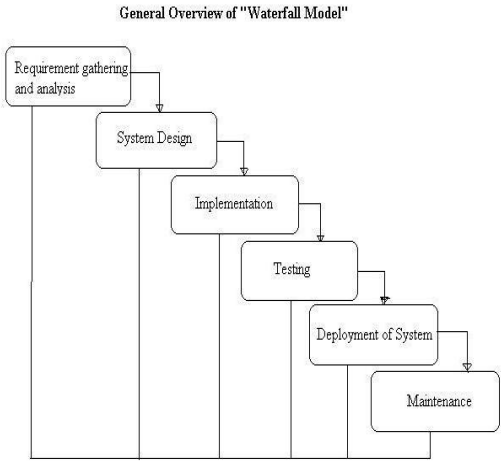
There are various Software development models or methodologies. They are as follows:

- ✓ Waterfall model
- ✓ RAD model
- ✓ Spiral model
- ✓ Prototype model

2.Description

2.1 Waterfall Model

The Waterfall Model was first Process Model. It reference as a **linear-sequential life cycle model**. It is very simple to understand and adopt. In waterfall model, each stage must be completed altogether before the next stage can begin. This type of software development model is generally used for the project which is small and there are no indistinct requirements. At the end of each stage, a review takes place to determine if the project or model is on the right path and whether or not to continue or discard the project. In this model software testing starts only after the development is complete or whole-hog. In **waterfall model stages** do not overlap or fold over.



Advantages of waterfall model:

- This model is simple and easy to adopt and use.
- It is easy to handle due to the rigidity of the model – each stage has specific deliverables and analysis process.
- In this model stages are processed and completed one at a time. Stages do not overlap or fold over.
- Waterfall model works well for smaller or minor projects or programs where fulfillments are very well understood.

Disadvantages of waterfall model:

If an application is in the testing stage, it is not possible 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.
- There is high amounts of risk and uncertainty.
- Poor models for long and ongoing projects.
- Not suitable for the projects where fulfillments are at a considerate to high risk of changing

When to use the waterfall model:

- This model is utilized when the requirements are well known , fixed and clear.
- Product definitions are stable.
- Technology is understandable.
- There should be no ambiguous requirements.
- The projects are short.

Very less customer intercommunication is involved during the development of the product. Once the product is ready then after it can be demoed to the end users or customers. If the product is developed and if any failure arise then the cost of fixing such issues are high rated, since we need to update everywhere from document till the logics implemented.

2.2 Spiral Model

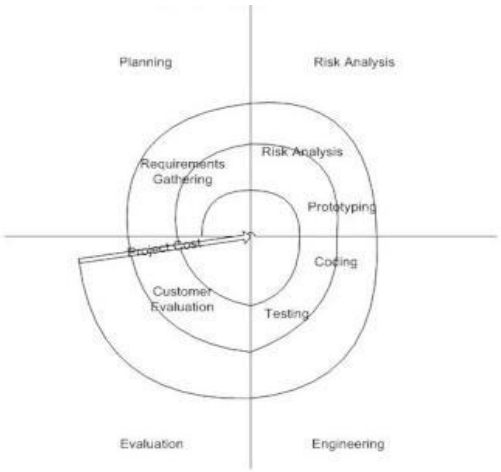
The spiral model is nearly similar to the incremental model, with more intensity placed on risk analysis. Spiral model has four phases or stages: Planning, Risk Analysis, Engineering and Evaluation. A software project again and again passes through these stages in iterations (called Spirals). The baseline spiral, starting or origin in the planning stage, fulfillments are gathered and risk is appraise. Every subsequent spirals builds on the baseline spiral. Its one of the software development models as Waterfall, V-Model.

Planning Stage: Requirements are accumulated during the planning stage. Requirements like 'BRS' i.e. 'Business Requirement Specifications' and 'SRS' i.e. 'System Requirement specifications'.

Risk Analysis: In the **risk analysis stage**, a process is undertaken to identify risk and alternate the solutions. A prototype is produced at end of risk analysis stage. If any risk is found during the risk analysis then the alternate solutions are suggested and implemented.

Engineering Stage: In this stage software is **developed**, along with testing at the end of stage. Hence in this stage the testing and development is done.

Evolution Stage: This stage grant the customer to assess the output of project to date before the project continues to next spiral.



Advantages of Spiral model:

- High amount of risk dissection hence, avoidance of the Risk is enhanced.
- Good for the large and mission-critical projects.
- Strong concurrence and documentation control.
- Additional Functionality can be added on a later date.
- Software's are produced early in the software life cycle.

Disadvantages of Spiral model:

- It can be costly model to use.
- Risk analysis requires highly explicit expertise mastery.
- Project's success is highly dependent on the risk analysis stage.
- Doesn't work well for the smaller or minor projects.

When to use Spiral model:

- When costs and risk evaluation is important it is useful
- For the medium to high-risk projects
- Long-term project commitment unwise because of potential changes to economic preferences
- Users are unsure of their needs and requirements
- Requirements or fulfillments are complex
- New product line are implied

2.3 RAD Model

RAD model is a Rapid Application Development model. It's a type of incremental model. In RAD model the functions are developed in parallel as if they were minor projects. The developments are time boxed, delivered and then massed into a working prototype. This can quickly give the customer something to use and to provide feedback in regard to the delivery and their requirements or fulfillments.

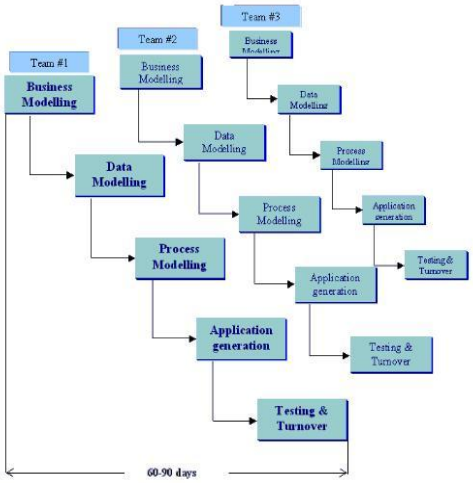


Figure 1.5 – RAD Model

The stages in rapid application development (RAD) model are:

Business modeling: The information flow is identified b/w many business functions.

Data modeling: Information massed from business modeling to define a data object that are needed for business.

Process modeling: Data object defining in data modeling are reformed for achieving the business information flow to achieve some specific business objective. Descriptions are identified and are created for CRUD of data objects.

Application generation: Automated tools are used for replacing process models into code and actual system.

Testing and turnover: Test the new components and all their interface.

Advantages of the RAD model:

- Reduce the development time.
- Increase the reusability of components
- Quick and initial reviews occur
- Encourages the customers feedback
- Integration from beginning solves a integration issues

Disadvantages of RAD model:

- Depend on strong team and an individual performances for identifying business requirements.
- Only system which can be modularized can be built using RAD
- Requires highly skilled developers and designers.
- High dependency on the modeling skills
- Inapplicable to the cheaper projects as the cost of modeling and automated code generation is high.

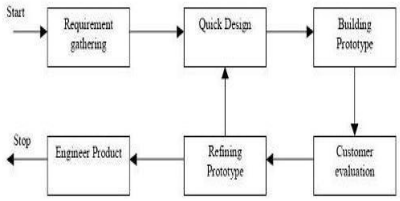
When to use RAD model:

- RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
- It should be used if there's high availability of developers for modeling and the budget is high enough to afford their cost.
- RAD SDLC model should be chosen only when the resources are with high business knowledge are available, there is a need to produce the system in a short span/term of time.

2.3 Prototype Model :

The basic information in **Prototype model** is that instead of freezing the requirements before coding and design can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the recently known requirements. Prototype model is a software development model. By using prototype, the client can get "actual feel" of system, since the interactions with prototype can enable the client to better understand the requirements of desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system for determining the requirements.

The prototype are not usually complete systems and many of the details are not built in prototype. The aim is to provide a system with overall functionality.



Prototyping Model

Advantages of Prototype model:

- Users are actively involved throughout development
- Since in this methodology is a working model of the system which is provided, the users gets a better understanding of the system which is being developed.
- Errors can be detected previously.
- Quick user feedback is available leading to a better solutions.
- Missing functions can be identified easily
- Confusing functions can be identified, Requirements validation, Quick implementation.

Disadvantages of Prototype model:

- Leads to implementations and then repairing way of building the systems.
- This methodology can increase the complexity of system as scope of the system may expand beyond the original plans.
- Incomplete application can cause application not to use as full system was designed inadequate problem analysis.

When to use Prototype model:

- Prototype model should be used when the desired system wants to have a lot of interaction with the customer.
- Online systems, web interfaces have a high amount of interaction with customers, are best for Prototype model.
- Prototyping ensures that customer constantly work with system and provide a feedback which is incorporated in the prototype to result in useable system. They are excellent in designing good human computer interface systems.

3. Conclusion

In this paper, different methodologies are examined in the software development life cycle by categorizing them in traditional and modern methodologies. Advantages and shortcomings of all these models have been described. Traditional methods are used for extremely analytical projects where requirements do not change often, with limited requirements, limited features and large number of developers. Modern methodologies are used for somewhat low analytical projects where requirements changes often with flexibility of design ,improvement in quality, iterative and incremental delivery, increased performance, easy fault detection capabilities. Hence, it is important for a development team to select a software development model that best suits the project .

4. Acknowledgement

The success of our research is never limited to the individual undertaking the paper. It is the cooperative effort of the people around an individual that spell success. For all efforts, behind this successful research, we are highly intended to all those personalities without whom this research would ever be completed. We find no words to express our gratitude towards those who were constantly involved with us throughout our work.

5. References

[1]. Ian Somerville, "Software Engineering", Addison Wesley, 7th edition, 2004.

[2]. Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.

[3]. Vinay Tiwari, “Software Engineering Issues in Development Models of Open Source Software”, International Journal of Computer Science and Technology, Vol. 2, Issue 2, pp. 38-44, June 2011.

[4]. K. K. Aggarwal, Yogesh Singh, Software Engineering, 3rd edition, New Age International Publishers, 2007.

[5]. Barry Boehm, “Anchoring the Software Process”, IEEE Transaction on Software Engineering, pp. 79-82, July 1996.

[6]. U. A. Khan, “The evolution of component based software engineering from the traditional appro

