ISSN 2278-3091

Volume 3, No.5, September - October 2014

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://warse.org/pdfs/2014/ijatcse02352014.pdf

Review of Software Development Methodologies Used in Software Design

Er. Sheilly Padda[1], Er. Apoorva Arora[2], Er. Sonali Gupta[2], Er. Priya Sharma[2]

[1]Assistant Professor, CSE Deptt. CEC Landran Mohali Punjab, Email: cecm.cse.sheilly@gmail.com

[2]Assistant Professor, CSE Deptt. CEC Landran Mohali Punjab, Email: cecm.cse.apoorva@gmail.com

[2]Assistant Professor, CSE Deptt. CEC Landran Mohali Punjab, Email: cecm.cse.sonaligupta@gmail.com

[2]Assistant Professor, CSE Deptt. CEC Landran Mohali Punjab, Email: cecm.cse.priyasharma@gmail.com

ABSTRACT

Software development process is the process where activities are carried out related to the production of software. A development process model specifies some activities according to the model and the order in which they should be performed.

Keywords: Agile, Iterative, Prototype, Scrum, V-Model.

1. INTRODUCTION

Software development is branch of Software Engineering which deals with the study and application of engineering process to the design, development, and maintenance of software. The field of software development has been started in the early 1940s. Other terms such as software development and information technology are widely used instead.

There are various software development methods. Out of those some methods are discussed here with their different behavior and features.

The various methods of S/w development are:

- Waterfall model
- Prototype model
- Iterative model
- V-Model
- Scrum
- Cleanroom
- Agile development

2. WATERFALL MODEL

The waterfall model describes the simplest process model which is organized in linear way. It is also known as linear sequential model. This model consists of –analysis, design, coding and testing, implementation and maintenance [1].

The waterfall model includes requirement analysis where requirements are gathered from different sources resulting in product requirement document. These requirements are then designed to form a proper design for each module. The next stages are coding and testing where the designed module is subjected to different coding standards and treated under testing stage to remove errors so as to build the actual software as proposed [2].

After the coding and testing stage the software implementation is carried out by the users. Fig 1. shows the waterfall model of system development [3].

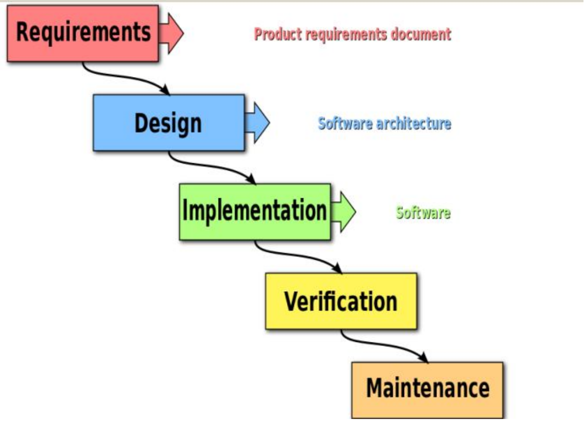

Figure 1: Waterfall model of system development

Modified Waterfall Model

The modified waterfall uses the same phases as the pure waterfall model, but does not allow discontinuous approach. This leads to overlapping of software process phases. The pure waterfall can also split into subprojects at an appropriate phase (such as after the architectural design or detailed design) [4].

Merits of waterfall model:

- Simple to use and understand.
- Rigid structure allows to manage it easily.
- Completion of phases one at a time.
- Benefits only small projects which requires less resources requirement.

Demerits of waterfall model:

- Once the testing phase starts it becomes very complicated to go back and modify.
- High amounts of risk and uncertainty.
- Unsuitable for projects which are complex and object-oriented.
- Deprived model for long and enduring projects.
- Not proper for the projects where requirements are reasonable changing.

3. PROTOTYPE MODEL

The prototype model includes creating an incomplete version of software program being developed. The basic idea of this model is that instead of freezing the requirements before any design or coding can proceed, a prototype is built. This prototype is built on the current requirements of the system. The prototype model [5] is depicted in Fig 2.

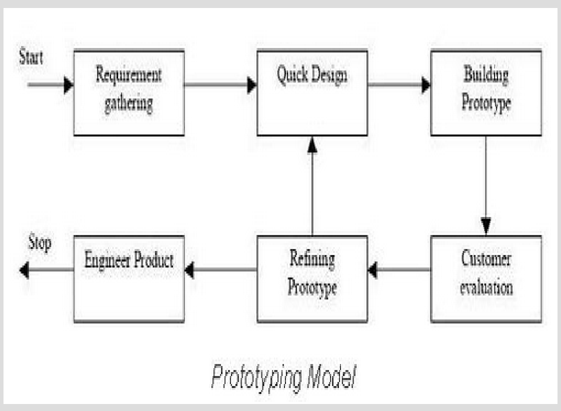

Figure 2: Prototype model

The prototype is developed so that the user can get that it is interacting with the actual system and this can enable the client to understand the requirement of the system in a better way. Prototyping process is a solution to complicated and large systems for which there is no manual process or existing system to help determining the requirements. These prototypes are not always complete and many of the details are not built in the prototype. The goal of prototype is to provide a overall functional system.

Merits of Prototype model:

- Requires active involvement of users.
- As a prototype is built means a working model of the system is provided, the users get a better understanding of the system being developed.
- Early error detection process.
- Better solutions can be formulated as it provides quicker user feedback.
- Leads to easy detection of any missing functionality of the system.
- Confusing or difficult functions can be identified..

Demerits of Prototype model:

- Involves implementing and then repairing way of building systems.
- If seen practically, this method may increase the complexity of the system.
- Incomplete application may cause application not to be used as the full system was designed.
- Incomplete or inadequate problem analysis.

4. ITERATIVE MODEL

Iterative method for developing software is suited for management of small to medium projects. Unlike waterfall model, which uses top-down approach for developing software, the iterative method integrates the design and prototypes throughout the life of the project.The development of software begins with defining requirements and developing a working prototype. This prototype is tested and feedback is gathered depending upon the tested results. The data collected from testing is further refined to create a new prototype. The designing, testing and enhancement continue until further changes are not needed and the project is considered to be complete [6].

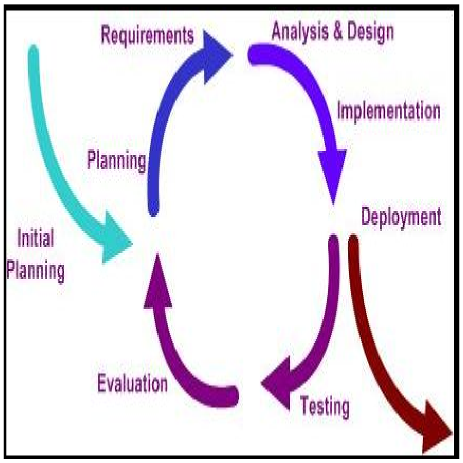

Figure 3: Iterative development model

Comparison with waterfall model

Waterfall development completes the project-wide work-products of each discipline in one step before moving on to the next discipline in the next step. Allows the delivery of fully developed product and that too at the very end of the project. Backtracking is not possible in an waterfall approach.

Merits of Iterative model:

- 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.
- 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 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.

Demerits of Iterative model:

- 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

5. V-MODEL

The V-model represents a software development process and this model can be applied for hardware development also, which may be considered an extension of the waterfall model. In case of waterfall model we move in linear direction, but in V-model instead of moving down in a linear way, the development steps are bent upwards after the coding phase, to form V shape. The V-Model describes the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represent time or project completeness (left-to-right) and level of abstraction. Fig 4 shows the V-Model below [7].

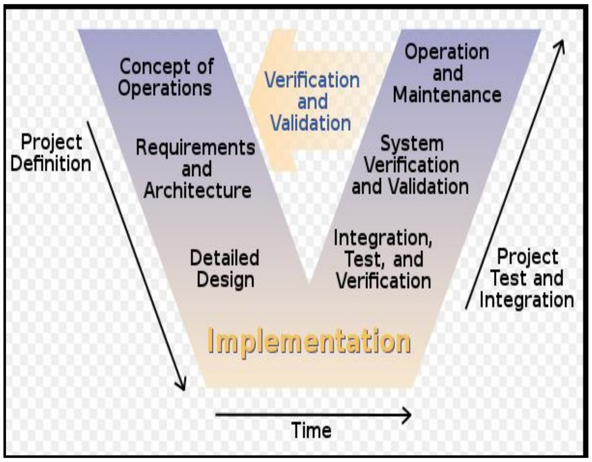
Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

Figure 4: The V-model

The requirement phase includes verification and product requirements. This process specifies what is to be performed. Usually the users are interviewed and requirements are gathered which results in the generation of requirement document. Next step includes system design and architecture design in which the developer analyses the user requirements and starts designing those requirements accordingly. The designing is carried out in two ways: Architecture design (high level design) and Module design (Low level design).The architecture design describes the functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology. In module design the designed system is broken into smaller units and each of them is coded by the programmer. This design contains detail logical functioning of the module. The unit test design is developed in this stage.

6. SCRUM

Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal". It challenges assumptions of the "traditional, sequential approach" to product development. Scrum enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members and daily face-to-face communication among all team members and disciplines in the project.

A key principle of Scrum is its identification that during a project the customers can change their minds about what they want and need (often called requirements churn), and that unexpected challenges cannot be easily addressed in a traditional predictive or planned manner. As such, Scrum adopts an empirical approach—accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly and respond to emerging requirements.

The authors described a new approach to commercial product development that would increase speed and flexibility, based on case studies from manufacturing firms in the automotive, photocopier and printer industries. They called this the *holistic or rugby approach*, as the whole process is performed by one cross-functional team across multiple overlapping phases, where the team "tries to go the distance as a unit, passing the ball back and forth"

7. CLEANROOM DEVELOPMENT

Clean room software development is a software development philosophy that is based on avoiding software defects by using formal methods of development and a rigorous inspection process. The name ‘Clean room’ was derived by analogy with semiconductor fabrication units. In these units (clean rooms) defects are avoided by manufacturing in an ultra-clean atmosphere. The objective of this approach to software development is zero-defect software.

The Cleanroom approach to software development is based on five key strategies:

- *Formal specification:* The software to be developed is formally specified. A state-transition model which shows system responses to stimuli is used to express the specification.
- *Incremental development:* The software is partitioned into increments which are developed and validated separately using the Clean room process. These increments are specified, with customer input, at an early stage in the process.
- *Structured programming*: Only a limited number of control and data abstraction constructs are used. The program development process is a process of stepwise refinement of the specification. A limited number of constructs are used and the aim is to apply correctness-preserving transformations to the specification to create the program code.
- *Static verification:* The developed software is statically verified using rigorous software inspections. There is no unit or module testing process for code components.
- *Statistical testing of the system:* The integrated software increment is tested statistically to determine its reliability. These statistical tests are based on an operational profile which is developed in parallel with the system specification.

A model of the Cleanroom process, adapted from the description given by Linger (Linger, 1994), is shown below in Fig 5. This shows how these essential strategies are integrated [8].

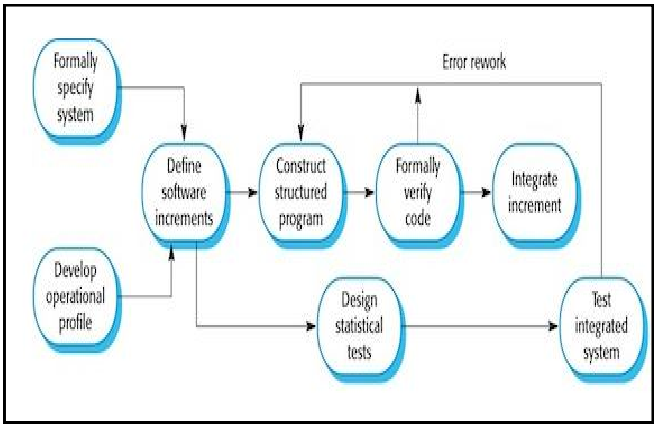

Figure 5: The Clean room process

Advantages of cleanroom development:

- Team orientated incremental pipelining approach allows for many components to be worked on concurrently therefore increasing productivity

- Continuous team meetings/reviews along with statistical testing and correctness verification produce a far better quality code then unit testing and debugging which may lead to further errors
- Rigorous formal design and specification methods and refinement lead to reduction in code size critical for embedded situations
- Due to the improvement in software performance and the time and money saved due to significantly less testing revenue on software will readily increase
- Finally, Cleanroom development certifies a software's reliability and presents the user with a near zero defect product .

Cleanroom developed Software:

- IBM COBOL/SF restructuring tool
- IBM AOExpert/MVS™ system outage analyzer
- Ericsson Telecom OS32 operating system
- IBM mass storage control unit adapters
- SEL at NASA Goddard Space Flight Center
- U.S. Army Picatinny Arsenal

8. AGILE DEVELOPMENT

The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability [9].

Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications. Extreme Programming (XP) is currently one of the most well known agile development life cycle model.

Agile development provides opportunities to assess the direction throughout the development lifecycle. This is achieved through regular flow of work, known as Sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.” In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development — requirements, design, etc. — is continually revisited. When a team stops and re-evaluates the direction of a project every two weeks, there's time to steer it in another direction. Fig 6 shows the agile development methodology [10].

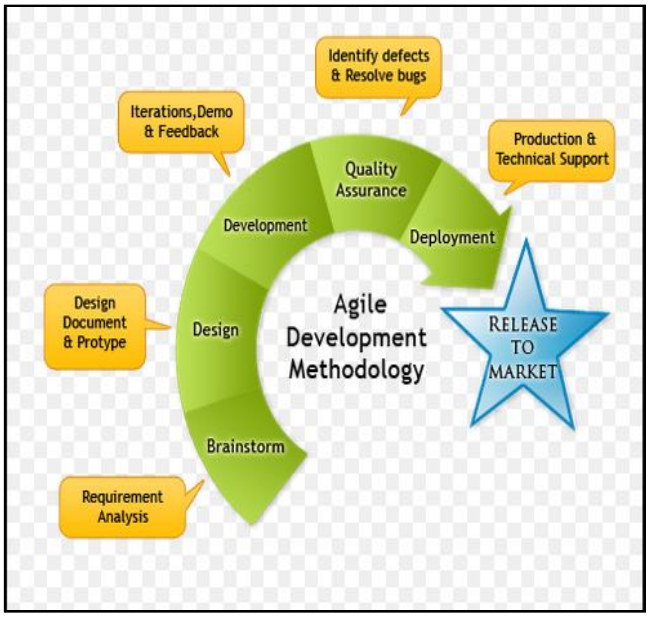
Identify defects & Resolve bugs
Iterations,Demo & Feedback
Production & Technical Support
Quality Assurance
Development
Deployment
Design Document & Protype
Design
Agile Development Methodology
RELEASE TO MARKET
Brainstorm
Requirement Analysis

Figure 6: Agile Development Methodology

This “inspect-and-adapt” approach to development greatly reduces development costs and time to market. Because teams can develop software at the same time they are gathering requirements, “analysis paralysis” is less likely to impede a team from making progress. And because a team's work cycle is limited to two weeks, stakeholders have recurring opportunities to calibrate releases for success in the real world. Agile development helps companies build the right product. Instead of committing to market a piece of software that hasn't been written yet, agile empowers teams to continuously replan their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Agile development preserves a product's critical market relevance and ensures a team's work doesn't wind up on a shelf, never released.

Advantages of Agile model:

- 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.
- Close daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

Disadvantages of Agile model:

- 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.

9. CONCLUSION

In this paper, different ways of developing a software product are discussed. Each method of developing software is different from one another in some way or the other. These methods provide us a framework for developing our system in proper manner.

REFERENCES

1. Andrew Kumiega Ph.D. and Ben Van Vliet, "**Article on A Software Development Methodology for Research and Prototyping in Financial Markets**" 3 Mar 2008.

2. Royce, Winston (1970), "**Managing the Development of Large Software Systems**", *Proceedings of IEEE WESCON.*

3. Paul smith June2010

4. Nabil Mohammed Ali Munassar and A. Govardhan , "**A Comparison Between Five Models Of Software Engineering**", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010).

5. http://istqbexamcertification.com.

6. George F. Sullivan III, "**The Iterative Design Process In Research and Development**", Department of Electrical Engineering Junior First Co-op Work Term National Aeronautics and Space Administration, Spring, 2013.

7. *Clarus Concept of Operations.* Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005.

8. http://ifs.host.cs.standrews.ac.uk/Books/SE9/Web/Cleanroom.

9. http://agilemethodology.org/

10. How Agile methodology in mobile app development can be benefitted,© 2013, Prismetric Technologies Pvt. Ltd.