

Empirical Study of Software Development Life Cycle and its Various Models

Mohd. Ehmer Khan

Lecturer

*Department of Information Technology
Al Musanna College of Technology
P.O. Box-191, PC-314, Sultanate of Oman*

ehmerkhan@gmail.com

S. G. M. Shadab

Lecturer

*Department of Information Technology
Al Musanna College of Technology
P.O. Box-191, PC-314, Sultanate of Oman*

sgmshadab@gmail.com

Farmeena Khan

*Department of Computer Science
EILLM University
Jorethang, Sikkim, India*

ehmerkhan@gmail.com

Abstract

A process used by the software industry to design, develop and test high quality software is called software development life cycle. The main aim of SDLC is to produce high quality software that meets customer expectation. We can also refer SDLC as Application Development Life Cycle. SDLC is not a methodology it is a description of various phases that are involved in software development starting from project definition to deployment and sustainment. These SDLC phases serve as a programmatic guide to project activity. In our paper we have explain various SDLC models (Waterfall, Spiral, V-Model, Iterative, Big Bang, Agile and Rapid Application Model).

Keywords: Waterfall, Spiral, V-Model, Iterative, Big Bang, Agile, Rapid Application Model.

1. INTRODUCTION

SDLC over the years has remained the reliable approach to software development. It is a framework that defines tasks performed at each step in the software development process that's why we also called SDLC as software development process. SDLC is a mechanism for project tracking and control, it increases visibility of project planning and enhance development speed [1] [2].

Stages of SDLC are:

- 1) Project Definition
- 2) Requirement
 - a) User requirement
 - b) System requirement
- 3) Analysis and Design
- 4) System Build
- 5) Testing and Implementing
- 6) Deployment
- 7) Sustainment

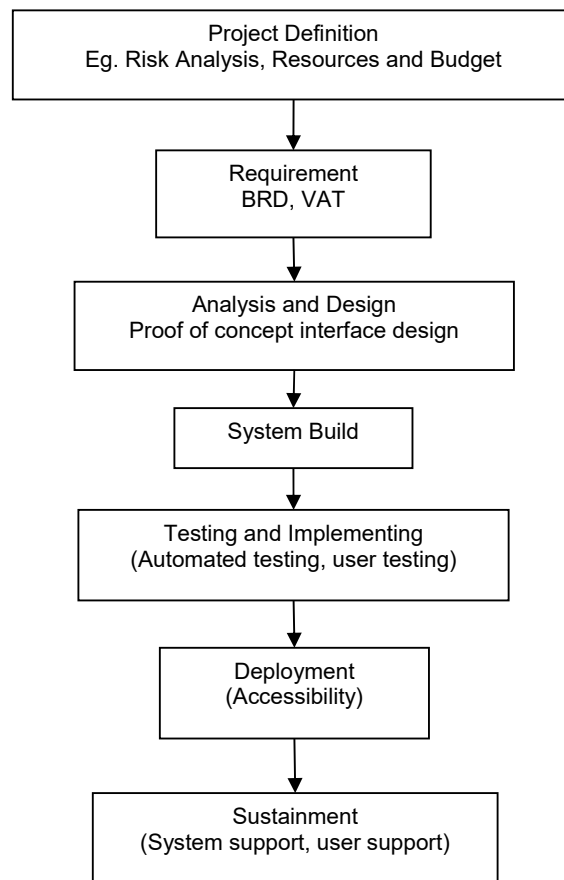


FIGURE 1: Represent the Various Stages of SDLC.

2. SDLC MODELS

To manage the level of complexity, a number of SDLC methodologies or models have been created. These models are created to ensure success in software development process [3].

Below are the SDLC models followed in the software industry: -

2.1 Waterfall Model

It is a non-iterative (linear sequential) design process. In a waterfall model each phase must be completed before the new phase can begin, that is, the progress is seen as flowing downwards through all the phases like system feasibility, requirements, analysis design, code and unit test, system integration, installation and maintenance.

The difficulties which were previously encountered in software projects were eliminated by waterfall model and it ensures the success of the project. Typically, in waterfall model the outcome of first phase is the input of next phase [4] [5] [6].

Advantages

1. Requirements are clearly defined, that is they are simple and easy to understand.
2. Easy to manage.
3. Early identification of slippages.
4. Process and results are well documented.

Disadvantages

1. High amount of risk and uncertainty as customer requirement may change.

2. No working software is produced until late in the life cycle.
3. Phases cannot run concurrently.

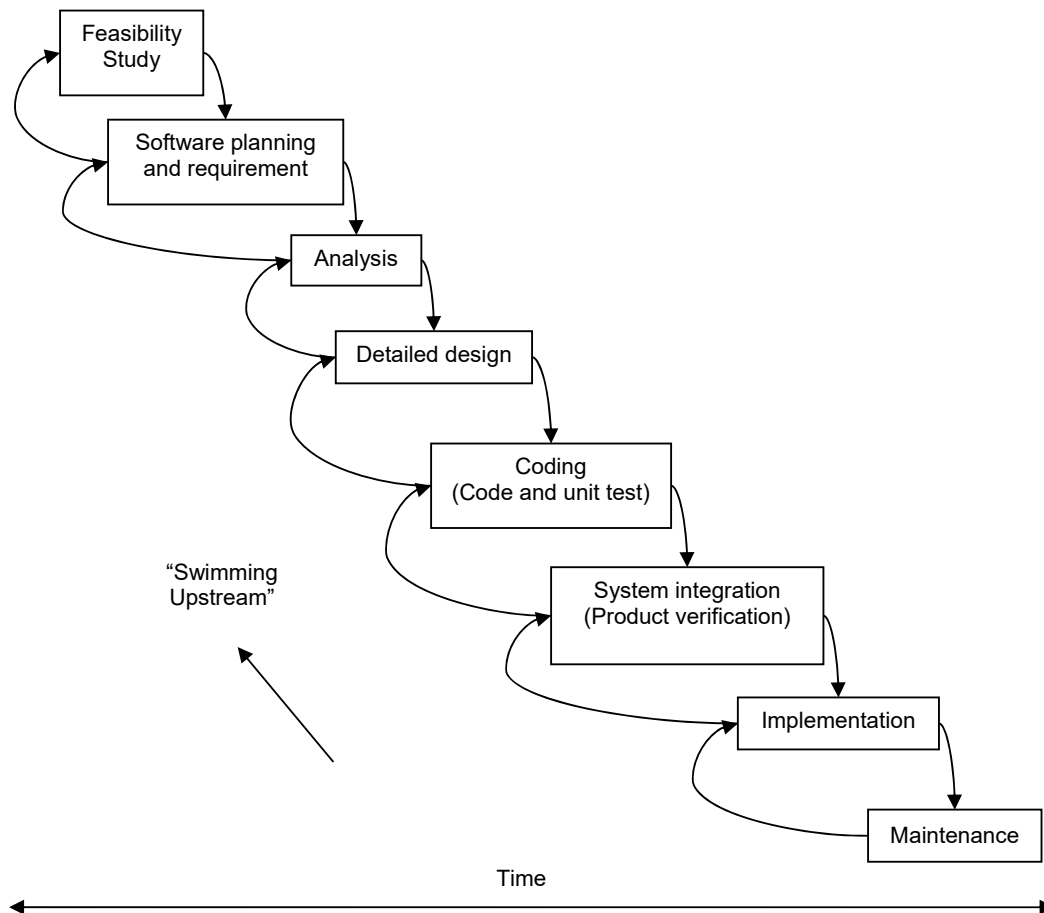


FIGURE 2.1: Represent the Waterfall Model.

2.2 Spiral Model

Spiral model is a risk-driven process model generator for software projects; it combines the idea of iterative development with the controlled and systematic aspects of waterfall model [7] [8] [9].

Advantages

1. Software are created and handled in a strategic way and project monitoring is easy and effective.
2. Users can see the system early i.e. software is produced early.
3. Changes are implemented faster and can also be implemented later in the life cycle.
4. Documentation control and strong approval.
5. We can develop highly customized product by using spiral model.

Disadvantages

1. Not suitable for smaller projects and low risk project, as it is costly for smaller projects.
2. Development process is very complex due to amount of documentation required.
3. Complex management due to amount of documentation required in intermediate stage.
4. Risk analysis requires high expertise.
5. Risk of not meeting the desired schedule.

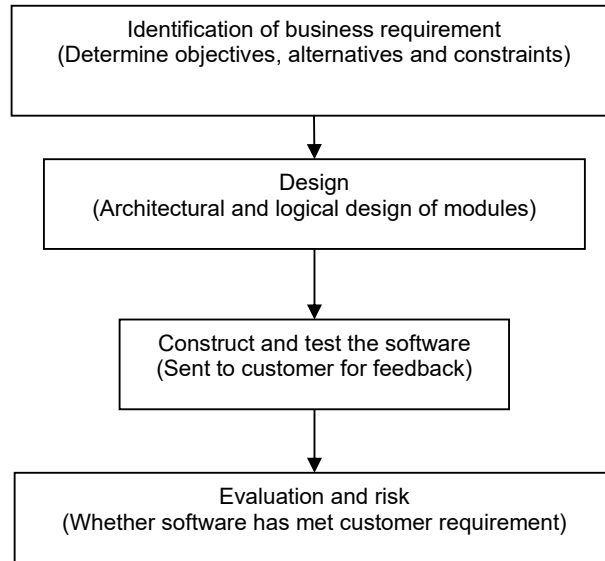


FIGURE 2.2(A): Represent the Steps Involved in Spiral Model.

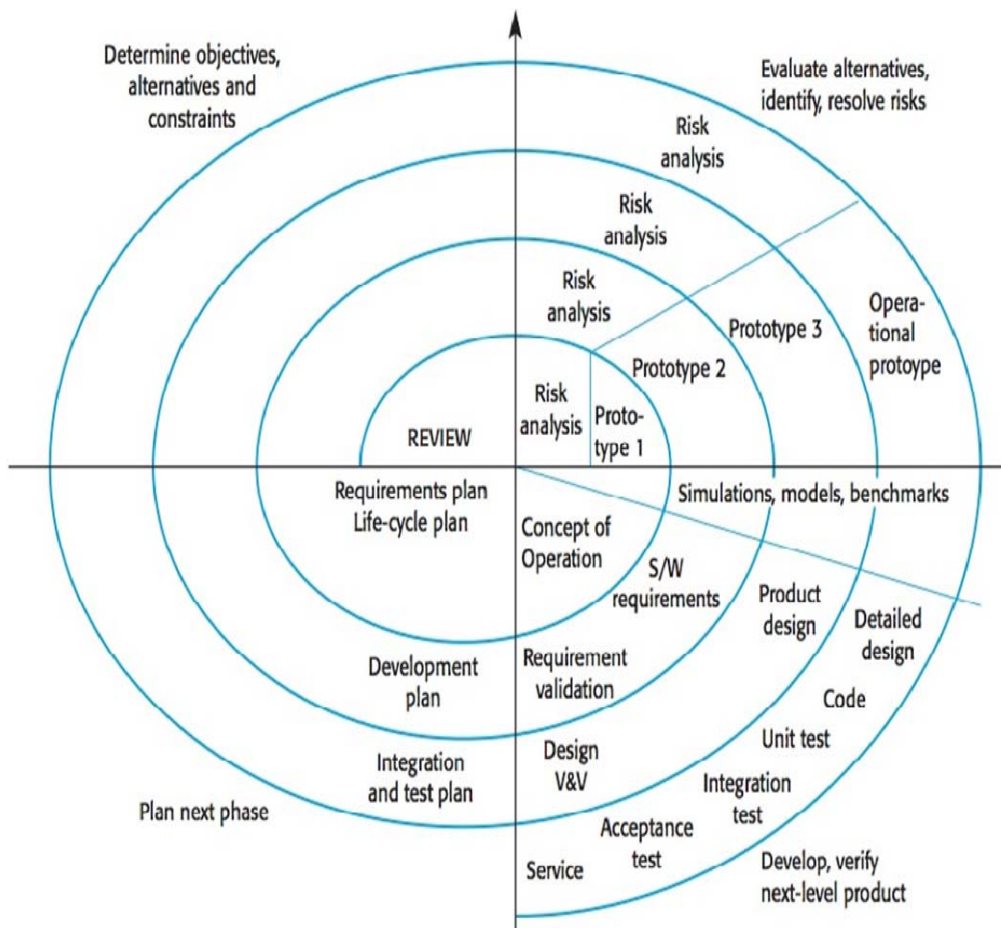


FIGURE 2.2(B): Represent the Spiral Model of a Software Process.

2.3 V-Model

V-model is the software development process where execution of process happens in a sequential manner in a V-form. V-shape represents the relationship between each developmental life cycle that is associated with the phase of testing. V-model is also known as verification and validation model. In this model before the beginning of next phase the previous stage should be completed [10] [11] [12].

Phases of V-model:

1. Verification Phase
 - a. Business requirement and analysis
 - b. System requirement analysis
 - c. Architecture engineering
 - d. Design
 - e. Detailed specification
2. Coding Phase: - Coding is performed which are based on the coding guidelines and standards.
3. Validation Phase
 - a. Unit testing
 - b. Component testing
 - c. System integration testing
 - d. System testing
 - e. Acceptance testing

Advantages

1. Very simple and easy to use as each phase has well defined objectives and goals.
2. Development and progress is very organized and systematic and phases are completed one at a time.
3. Small type projects are mostly worked well in V-model as requirements are easily understood.
4. Bugs are discovered at the earlier stage of the product development.
5. Higher chance of success over the waterfall model due to early development during the software development life cycle.

Disadvantages

1. Not suitable for complex and object oriented project as it is difficult to go back and change functionality.
2. Less chance of meeting the customer expectation as no prototype are produced.
3. Uncertainty and risks are there as there is no provision of doing risk analysis.
4. It lacks coherence and precision.
5. All project documentation should be updated if some modifications are made during software development process.

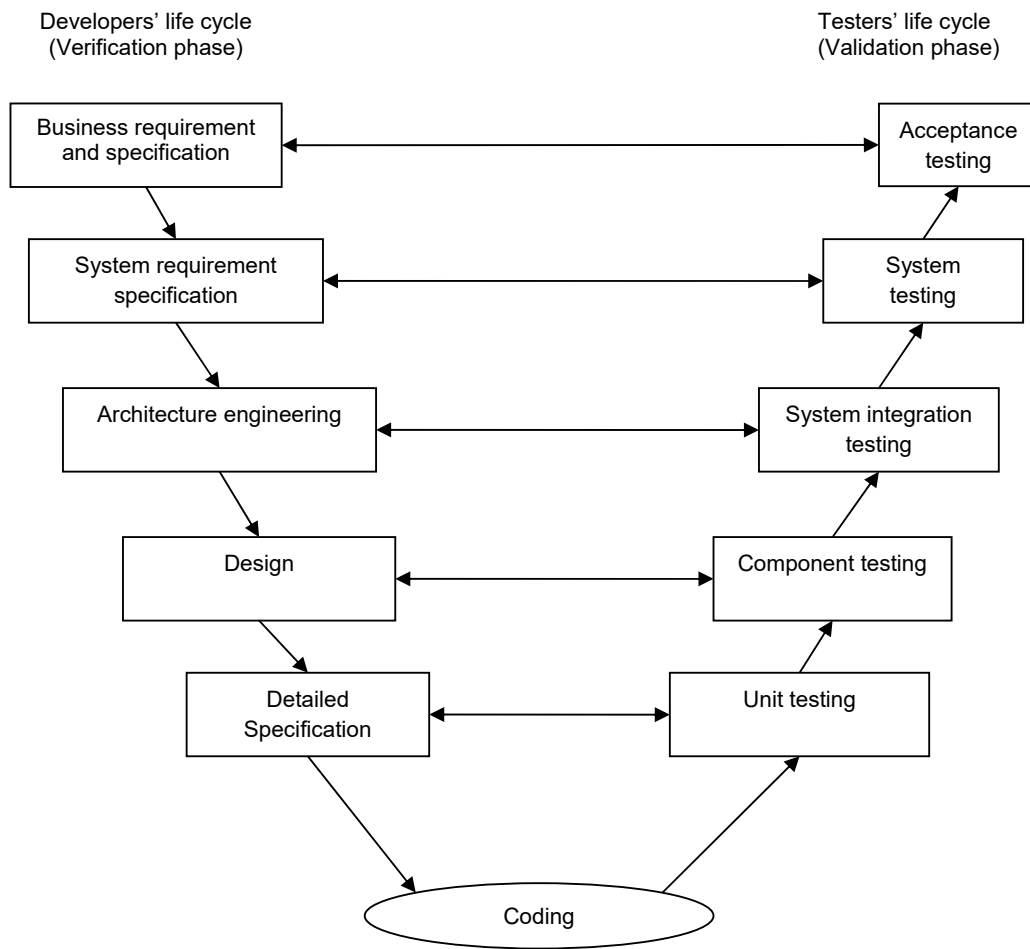


FIGURE 2.3: Represent the V-model.

2.4 Iterative Model

An iterative model does not attempt to start with a full specification of requirement instead starts with a simple implementation of a subset of the software requirements and after that interactively enhances the evolving versions until the system is implemented and ready to deployed. Iterative model is used when the requirements of the complete system are clearly understood and defined [13].

During software development process more than one iteration of the software development cycle may be in progress at the same time (as shown in Figure 2.4.) and it is called incremental approach.

Eg. of iterative approach.

2
 2 + 2
 2 + 2 =
 2 + 2 = 4

Advantages

1. Suitable for large projects, working software is generated quickly and early during the software development life cycle.
2. Less costly to change scope and requirement.
3. Testing and debugging are easier during smaller iteration.
4. Risk analysis is better- as risky process is identified during its iteration and it is easy to manage risks.

5. Lowers initial delivery cost.
6. Less time spent on documenting and more time is given for designing.

Disadvantages

1. Not suitable for smaller projects as management complexity is more.
2. Not all requirements are gathered up front for the entire software development life cycle.
3. Needs good planning and design i.e. more management attention is required.
4. Correction of a problem in one unit requires correction in all the units which consumes lots of time.

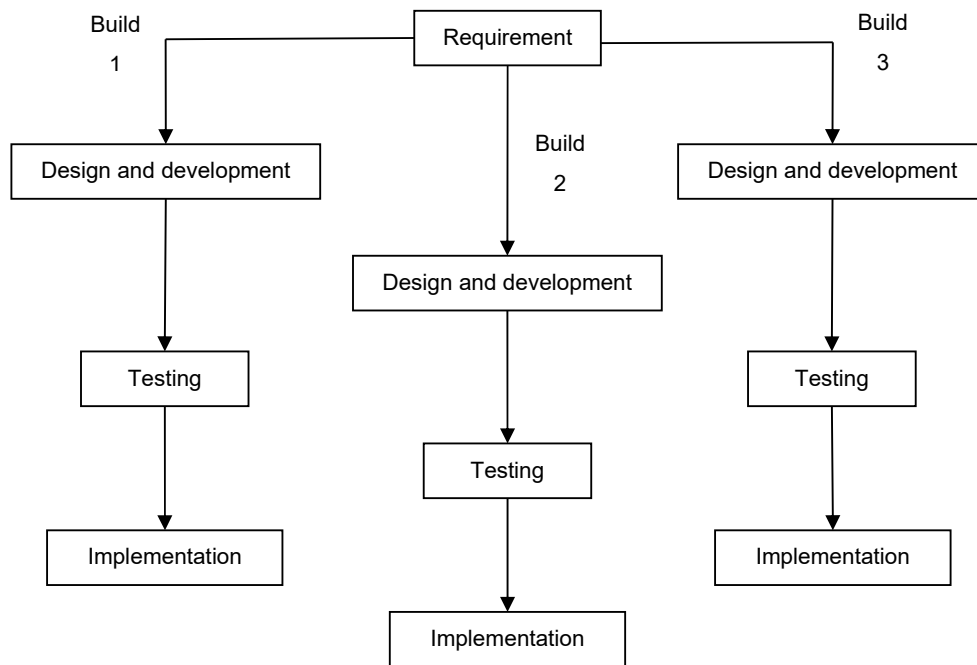


FIGURE 2.4: Represent the Iterative Model.

2.5 Big Bang Model

No specific process is followed in big bang model i.e. there is only little formal development process and very little planning is required. The requirements are implemented as they come. No need to revamp the complete software when any changes are required. Big Bang model is a very high risk model as misunderstood requirements may lead to failure. In this software development life cycle methodology only one or two software engineers are required and it is suitable for smaller projects [14].

Advantages

1. This is very simple model as there is very little or no planning is required.
2. It is easy manageable as very few sources are required.
3. It is a good learning aid for new comers as it allows them to get straight into the code and focus on development aspects.

Disadvantages

1. Not an ideal model for complex and object oriented projects.
2. Without proper planning, formal leadership and procedures it is extremely risky and uncertain.
3. When the project starts the big bang model is an inexpensive model but it turns out to be expensive if the requirements are misunderstood or changes dramatically without notice.

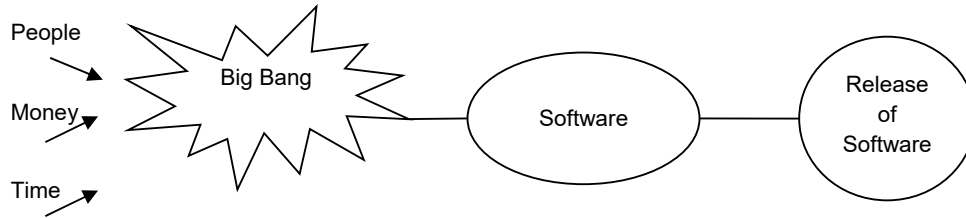


FIGURE 2.5: Represent the Big Bang Model.

2.6 Agile Model

Agile model is a combination of iterative and incremental process models with main focus is on process adaptability and customer satisfaction. Software is developed in incremental rapid cycle. It believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the requirement of the project. The agile model effectively gives the customer the finished system they want; as agile model assumes the user's requirements are changing in IT world. Some of the agile software development methods include agile modeling, agile unified process (AGP), crystal clear, dynamic systems development method (DSDM), extreme programming (XP), feature driven development (FDD), GSD, scrum, velocity tracking and team software development [15].

Advantages

1. It is legalistic approach to software development as agile model makes a project less cumbersome from communication point of view.
2. It is pay-as-you go system thus makes project finances very easy to manage from expense point of view.
3. It is easily manageable as little or no planning is required and thus gives flexibility to developers.
4. In agile model design are simple which makes design testable and thus therefore reliable system.

Disadvantages

1. As there are more risks of sustainability, maintainability and extensibility agile model is not suitable for complex dependencies.
2. Difficult to access the effort required at the beginning of the software development process.
3. Requirements are documented in much less detail due to which the transfer of technology to new team members may be quit challenging.
4. If the client is unclear about his requirement, team can be driven the wrong direction as agile software development process depends heavily on customer interaction.

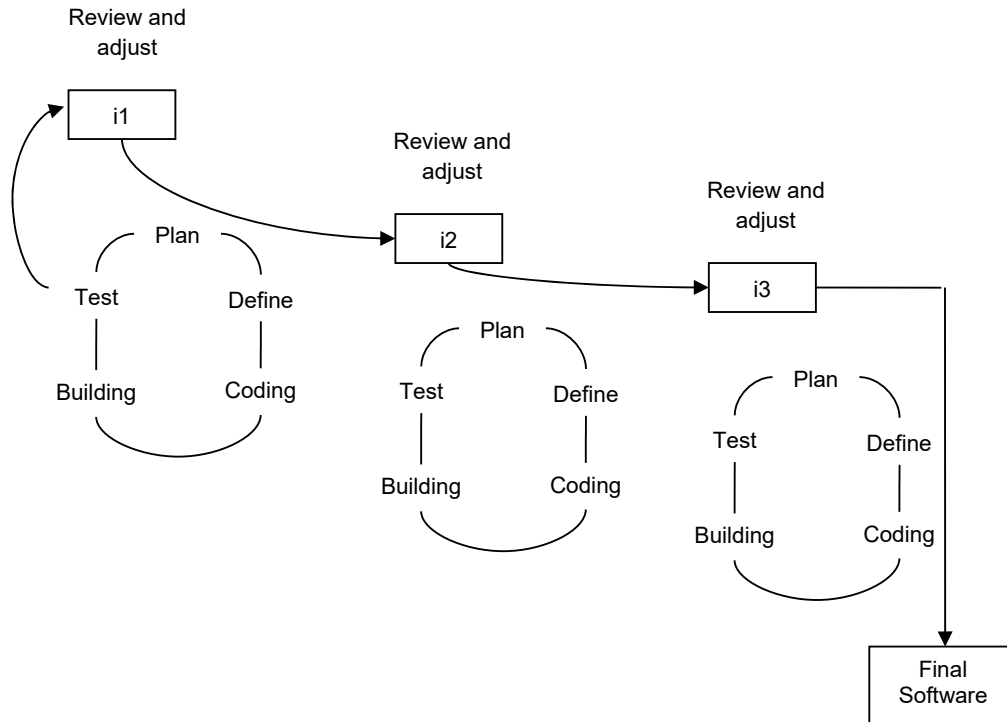


FIGURE 2.6: Represent the Agile Model.

2.7 RAD (Rapid Application Development Model)

In rapid application development method, no specific planning is involved and it is based on prototyping and iterative development i.e. if the project is large it is divided into small projects. The RAD model focuses on the reuse of code, processes, templates and tools and thus makes it successful. The main phases of RAD model are business modeling, data modeling, process modeling, application generation and testing and turnover. RAD usually embraces object oriented programming methodology such as C++ and Java (they are most popular OOP's language) [16] [17] [18].

Advantages

1. With the use of powerful RAD tools, the iteration can be reduced.
2. Users see the RAD product quickly as it usually incorporates short software development cycle.
3. Constant integration from beginning solves a lot of integration issues and encourages customer feedback.
4. Increases reusability of components and thus reduce development time.
5. There is a reduction of manual coding because of code generation and code reuse.

Disadvantages

1. Product may lose its competitive edge because of insufficient core functionality and thus exhibit overall poor quality.
2. As the cost of automated code generation is very high it is inapplicable to cheaper project.
3. There is no documentation to demonstrate what has been done. Progress and problems accustomed are hard to track.
4. Not all applications are compatible to RAD. It is only suitable for component based and scalable application.

3. ACHIEVED THROUGH THIS STUDY

Our study explains how various models help us to navigate through complex process of software development and also the stakeholder expectation largely depends upon the chosen model. Therefore, we have to choose right model according to the requirement of the project to ensure its success.

4. CONCLUSION

SDLC consists of a detail plan which explains how to build, plan and maintain specific software. It is a framework defining tasks performed at each steps in the software development process. There are various models design during the software development process such as (Waterfall, Spiral, V-Model, Iterative, Big Bang, Agile and Rapid Application Model). We also referred these models as software development models and to ensure success these models have to follow a series of steps unique to its type.

5. FUTURE RESEARCH DIRECTION

Existing software development models guides us in the right direction and helps us to achieve desired result. But in future certain features of software development models are needed to be regulated by developers such as using of graphical symbols that can easily be understood by users and also describes the problem which is to be solved. In coming future, we also have to establish new software development models due to changing customer needs and also because of the increased pressure of innovation.

6. REFERENCES

- [1] Mark E. McMurtrey. (2013, Dec 19) "A Case Study of the Application of the Systems Development Life Cycle (SDLC) in 21st Century Health Care: Something Old, Something New?" *The Journal of the Southern Association for Information Systems*: [On-line]. Vol. 1: Iss. 1, Article 2. PP. 14-25, Available: <https://aisel.aisnet.org/jsais/vol1/iss1/2> [July, 2018].
- [2] Cadle and Yeates. "Slide 6.1. System development life cycle by, Project Management for Information Systems", *5th Edition*, © Pearson Education Limited 2008 [Aug, 2018].
- [3] Tutorials Point. "Software Development Life Cycle [SDLC]" Internet: https://www.tutorialspoint.com/sdlc/sdlc_tutorial.pdf, Copyright 2017 [Aug, 2018].
- [4] Wikipedia, "Waterfall model" Internet: https://en.wikipedia.org/wiki/Waterfall_model, [Oct, 2018].
- [5] Sunnyday classnotes, "Waterfall model" Internet: <http://sunnyday.mit.edu/16.355/classnotes-process.pdf>, [Oct, 2018].
- [6] Testing brain tutorial, "Waterfall model" Internet: <http://www.testingbrain/tutorials/waterfall-model-tetsing.html>, [Oct, 2018].
- [7] Wikipedia, "Spiral model" Internet: https://en.wikipedia.org/wiki/Spiral_model, [Dec, 2018].
- [8] Andrew Powell-Morse, (2016, September 29) "Spiral Model: Software Development for Critical Projects in SDLC" Internet: <https://airbrake.io/blog/sdlc/spiral-model>, [Dec, 2018].
- [9] Barry W. Boehm, TRW Defense Systems Group, (May, 1988) "A Spiral Model of Software Development and Enhancement" *ieeexplore*; [On-line]. PP. 61-72, Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=59>, [Jan, 2019].
- [10] Tutorials Point. "SDLC-V-Model" Internet: https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm, [Feb, 2019].

- [11] Testing brain tutorial, "V-Model Software Development" Internet: <http://www.testingbrain.com/tutorials/v-model-tutorial.html>, [Feb, 2019].
- [12] TRY QA, "V-model" Internet: <http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/>, [April, 2019].
- [13] Tutorials Point. "SDLC-Iterative Model" Internet: https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm, [May, 2019].
- [14] Tutorials Point. "SDLC-Big Bang Model" Internet: https://www.tutorialspoint.com/sdlc/sdlc_bigbang_model.htm, [Aug, 2019].
- [15] Tutorials Point. "SDLC-Agile Model" Internet: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm, [Sept, 2019].
- [16] Tutorials Point. "SDLC-RAD Model" Internet: https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm, [Oct, 2019].
- [17] TRY QA, "RAD-Model" Internet: <http://tryqa.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/>, [Nov, 2019].
- [18] Wikipedia, "Rapid application development" Internet: https://en.wikipedia.org/wiki/Rapid_application_development, [Dec, 2019].