

Research Article

Analysis of SDLC Models

Ritika Arora^{†*} and Neha Arora[‡]

[†]Department of Information Technology Panjab University Swami Sarvanand Giri Regional Centre, Hoshiarpur, Punjab India
[‡]Infosys Limited Pune, Maharashtra India

Accepted 05 Feb 2016, Available online 09 Feb 2016, Vol.6, No.1 (Feb 2016)

Abstract

Software Development Life Cycle (SDLC) describes the activities performed at each stage of software development. Software Development did with utmost care and vigilance is bound to have bugs, limitations and errors because there is always a difference between what the users wanted, how the analyst interpreted, how it actually worked. It is the requirement of the design that has to be best suited by a SDLC. It is the job of the analyst to choose the right SDLC Model and implement it.

Keywords: SDLC, Waterfall, Iterative, Spiral, RAD, Agile, Lean, Big Bang Model, Prototype

1. Introduction

Software Development Life Cycle (SDLC) is a process that describes the methods and strategies like how to develop design and maintain the software project ensuring that all the goals, objectives, functional and user requirement are met. SDLC should result in high quality software that meets client's expectation, reaches completion within deadline and cost estimates, and works effectively and efficiently (Kushwaha *et al*, 2006).

2. Software development Life Cycle models

Waterfall Development

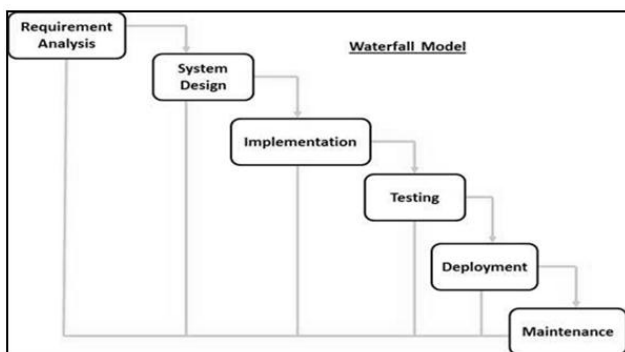


Fig 1: Waterfall Development

Waterfall development is mostly planned by using Gantt chart i.e. we complete one phase and then we

move to the next phase. We shall never revisit a phase once it is completed. The basic approach for waterfall method is that it sets requirement stability. Its requirements define needed information, function, behavior, performances, and interfaces. The design comprises of software architecture, interface representations and its implementations based upon the testing source.

The advantages of the waterfall approach are

- Simple and easy to use and understand.
- Phases do not overlap and hence management is very efficiently done.
- Works very well for small projects

The bottlenecks of the waterfall approach are

- We cannot seek approval from the user until the end.
- The project success is undetermined till the end.

When to use waterfall method

- This model is best suited when all the requirements are clear, fixed and very well unknown.
- Product definition and technology well understood.
- No ambiguous requirements.
- The project tenure is short.

Iterative Waterfall development

This approach carries less risk as compared to original waterfall method but is considered less efficient than

*Corresponding author Ritika Arora is working as Assistant Professor and Neha Arora is a Software Engineer

agile methodologies as sprint of work is delivered against a series of features.

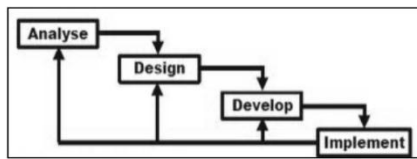


Fig 2: Iterative Waterfall Model

The advantages of the iterative waterfall method are

- We can track bugs at an early stage.
- We can avoid downward flow of defects.
- Less time spent on documenting
- More time is devoted to designing.

The bottlenecks of the iterative waterfall method are

- Costly system architecture
- Each phase is rigid with no overlaps.
- As each and every requirement is not gathered at the same time design issues may arise.

When to use iterative waterfall method

- When the requirements are clearly understood.
- The project is big enough.
- Minor details can evolve with time.

Agile Development

Agile is an iterative model where phases of a project run simultaneously i.e. in parallel. Requirements evolve at any stage at in any phase .There are many agile frameworks each having the same empirical structure (Jovanovich, D., Dogsa, T 2003).

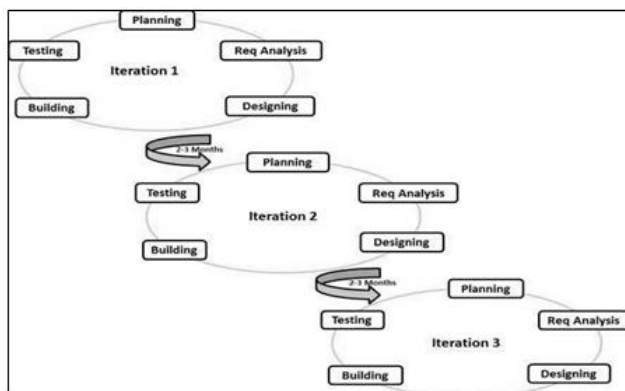


Fig 3: Agile Development

The advantages of the agile approach are

- Changes can be incorporated very easily.
- Without major re-planning costs changes can be handled well.
- Constant team involvement.

The disadvantages of the agile approach are

- More often there is problem in team structure rather than planning problems.
- Dubious planning date

When to use agile approach

- When there is continuous need to include changes.
- When both the developers and stakeholders have same freedom to suggest changes.

Scrum Development

Scrum development is combination of iterative and incremental agile software development methodology for managing product development. The focus is to deliver independent and valuable small features. Each feature is independent from the other if any one feature goes wrong the other features are left unaffected (Shikha maheshwari, Dinesh Ch. Jain et al, 2012).

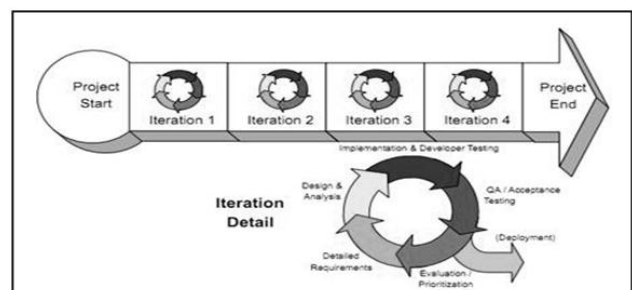


Fig 4: Scrum Development

The advantages of Scrum development approach are

- Makes the project economical and enables efficient use of time.
- Due to constant feedback changes can be incorporated easily.

The disadvantages of Scrum development approach are

- Continuous feedback keeps demanding new functionality.
- Project quality management is hard to implement.
- Works well with small team only.

When to use Scrum development approach

- When the request is of high priority and it requires high turnaround.
- When the process is cyclic i.e. planning for the next sprint overlaps with the work on the current topic.

Lean Development

In Lean development approach we focus individual feature and each individual feature consists of group of

features. In this development model we plan, build, test, review, deploy and again repeat the same steps for every next feature. We aim to eradicate whatever we do not require at every step (Ranjeet Kumar et al, 2013)

The advantages of the approach are

- Time to time elimination of waste which leads to overall efficiency
- Team can deliver more functionality in a shorter period of time.
- Empowered development team.

The disadvantages of the approach are

- There is great flexibility hence; the resultant product may be very different original framework.
- Tough decisions have to be made at early stages.

When to use lean development approach

- When we need insights into how to solve problems.
- When we need to focus on fewer, smaller, things in our development pipeline

Spiral development

Combining the features of both iterative and any one of the SDLC model a new framework called spiral is developed. The model starts with determining objectives and constraints of the software.

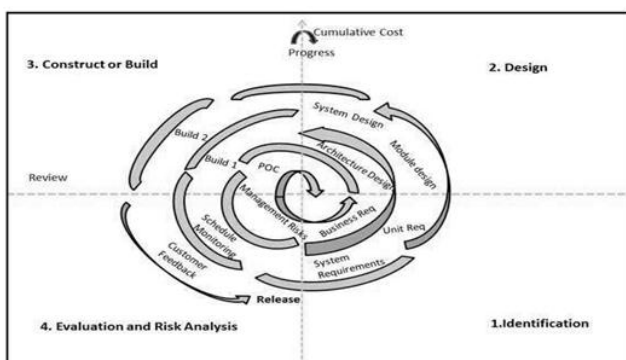


Fig 5: Spiral Development

The advantages of Spiral development are

- Extremely efficient for large and mission-critical projects.
- Software produced early in the development life cycle.

The disadvantages of Spiral development are

- Not suitable for small projects.
- Costly for short term.

When to use Spiral development approach

- Requirements are complex.
- When research and exploration are involved.

V-Model

In this model we test software at each stage in reverse manner. Test plans and test cases are created in every stage to verify and validate the product according to the requirement of every stage. Hence, verification and validation go in parallel.

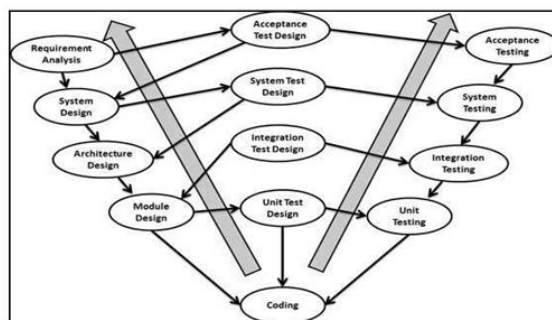


Fig 6: V Model

The advantages of the V-Model are

- Simple and easy to use.
- Planning, designing happens well before coding.

The disadvantages of the V-Model are

- High rigidity and low on flexibility.
- No early prototypes are produced.

When to use V-Model approach

- When ample technical resources are available with needed technical expertise.
- Requirements clearly defined and understood.

Big Bang Development

This model requires little programming, lots of programming and funds. This model is conceptualized around the big bang theory of the universe. Very small amount of planning is required. It does not follow any process and input requirements are arbitrary.

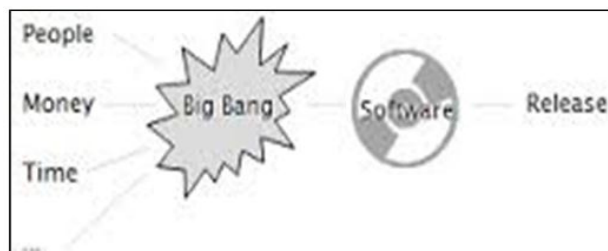


Fig 7: Big Bang Model

The advantages of the Big Bang development are

- Very few resources are employed.
- Very simple and easy to implement.
- Ideal for repetitive or small projects.

The disadvantages of the Big Bang development are

- Very high risk and uncertainty.
- Misunderstood requirements may lead to total scrapping of the model.
- Not suitable for complex and object oriented model.

When to use the Big Bang development approach

- High risk can be taken
- No deadline is to be met.

Rapid Application Development(RAD)

RAD has the key features of incremental model. In RAD model parallel development of functionalities and features take place as if they are independent projects. The development has strict deadlines. Each development is delivered and then assembled onto a working prototype (Laura C. Rodriguez Martinez, Manuel Mora, Francisco, J. Alvarez, 2009).

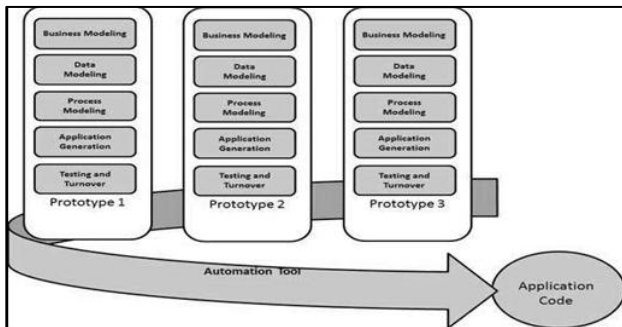


Fig 8: RAD Model

The advantages of the RAD model are

- Initial reviews occur quickly.
- Substantial reduction in development time.
- Integration from the beginning solves key issues.
- Customer feedback is encouraged.

The disadvantages of the RAD model are

- Highly dependent on modeling skills.
- Modular systems can be developed.
- Cost of modeling and automated code generation is high.

When to use the RAD model

- When we are creating a system that can be modularized in 2-3 months of time.

- The budget is high enough to employ large number of designers.
- Where good business analyst and advisors are involved.

Prototype Model

A prototype is built first to understand the requirements. This prototype is developed based on the currently known requirement, which is not a complete system as many details are not built in the prototype. The client gets the real feeling since; interactions with the prototype enable better understanding (Boehm, B. W et al, 1988).

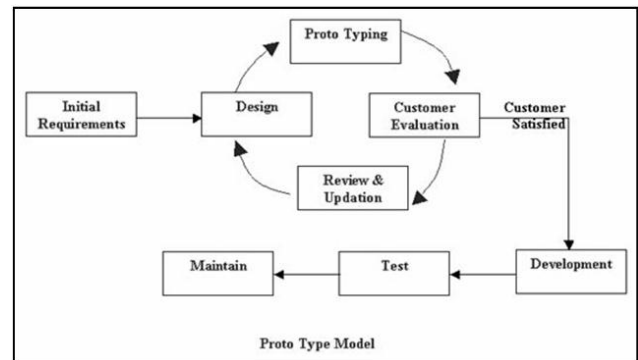


Fig 9: Prototype Model

The advantages of the prototype model are

- Users are actively involved in the development.
- Missing functionalities are identified early.
- Errors are detected much earlier.

The disadvantages of the prototype model are

- The complexity of the system increases as system may expand beyond original plans.
- Inadequate problem analysis.
- Incomplete application may cause application used differently from full system framework.

When to use prototype model

- When desired software needs extensive end user interaction
- When developing complete interface.

Conclusion

In this paper, we concluded that there are many existing models for software development life cycle based on clients' requirements and the size of projects. Some models are preferred over the others due to their characteristic properties and how they match the clients' needs. The waterfall model, spiral model, and incremental model may have some shared properties, but they still have different advantages and disadvantages for the development of systems, hence

each model tries to eliminate the disadvantages of the previous model. We have discussed the decision criteria to choose the right SDLC model by taking into account the features of each model in either the linear approach or agile approach and the characteristics of the software project. As all the models have their usability and limitations, so where it can be applicable is decided by its usability. This paper is useful for the developer to choose the SDLC model according to their requirements. In the future work, we are planning to extend this research to add other models and some models might be simulated using some tools.

Acknowledgment

I owe a great many thanks to a great many people who helped and supported me during writing of this paper. My deepest thanks to my family who guided and supported me in every phase of writing this paper. I am grateful to my parents who are inspirational in their understanding patience and constant encouragement.

About Authors



Mrs. Ritika Arora received her B.Tech. degree in Computer Science Engineering from Kurukshetra University, in 2009, then M.Tech. degree in Computer Science Engineering from Shaheed Bhagat Singh State Technical Campus, Ferozepur, in 2012 and is currently working as Assistant

Professor in Panjab University Regional Centre Hoshiarpur with an experience of 3 Years. Her research interests include Network Security. She has paper Published in various National and International Journals and Conferences.



Ms. Neha Arora received her B.Tech degree in Computer Science Engineering from Punjabi University, Patiala and is currently working as Software Engineer in Infosys Pune. She has won prizes in various National Talent Search Contest. Her research Area includes in Network security.

References

- Kushwaha *et al* (July 2006), Software Development Process and Associated Metrics-A Framework, *IEEE CNF*, Page(s):255 – 260.
- Ranjeet Kumar, Research methodology A step by step guide for beginners, pp 5-15, *SAGE Publications Asia Pacific Ltd*, ISBN 178-1-84920-300-5, 3rd Edition.
- Shikha maheshwari, Dinesh Ch. Jain (May 2012), A Comparative Analysis of Different types of Models in Software Development Life Cycle, published in *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, Volume 2, Issue 5, pp 285-290, ISSN: 2277 128X
- Jovanovich, D., Dogsa, T. (2003), Comparison of software development models, *Proceedings of the 7th International Conference on*, 11-13 June 2003, ConTEL, pp. 587-592.
- Laura C. Rodriguez Martinez, Manuel Mora, Francisco, J. Alvarez, (2009) A Descriptive/Comparative Study of the Evolution of Process Models of Software Development Life Cycles, *Proceedings of the Mexican International Conference on Computer Science IEEE Computer Society Washington, DC*
- Boehm, B. W. (May 1988) A Spiral Model of Software Development and Enhancement, ISSN: 0018-9162, Volume: 21, Issue: 5, on page(s): 61-72.