

SDLC Model Selection Tool and Risk Incorporation

Prachal Agarwal

B.Tech, CSE

Bharati Vidyapeeth's College of
Engineering, New Delhi,
India

Aroshi Singhal

B.Tech, CSE

Bharati Vidyapeeth's College of
Engineering, New Delhi,
India

Archit Garg

B.Tech, CSE

Bharati Vidyapeeth's College of
Engineering, New Delhi,
India

ABSTRACT

Software Engineering is systematic method of development of software. SDLC lists all the processes and tasks for a developing a software. Choosing right model is very crucial as based on it development and testing processes are carried out. Also error has a cascading effect in SDLC. In this paper, a number of existing SDLC Models (waterfall, V shaped, prototype and RAD), used to develop softwares commercially are discussed and compared with each other. This comparison is used as basis to propose a tool that helps to identify the best suited model for the current project. Also the risk management activities are incorporated in the models itself as compared to a standalone activity. Thus, a resource efficient product is developed.

General Terms

SDLC Model, Model Selection, Risk

Keywords

SDLC, Waterfall, V-model, Prototype, RAD

1. INTRODUCTION

The development of software in an organised method by the use of application of engineering is called Software engineering. Software Engineering includes research, design, development, testing and maintenance of software for industrial, business, scientific, medical, military, aerospace and general computing applications.

Selection of the appropriate model for software product or application development is very crucial. Based on the model, the development and testing processes are carried out. Different companies based on the software application or product, select the type of development model whichever suits to their application.

2. LITERATURE REVIEW

M. Tuteja and G. Dubey, [5] in 2009, focused on the state of the art in testing techniques, as well as the latest techniques which represent the future direction of this area, hence concluded to apply testing in all the phases of SDLC rather than a particular stage and suggested testing technique types according to SDLC phase.

C. Banerjee and S. K. Pandey, [7] in 2009, suggested that for secure software, security of the software must begin at the requirement level and should be present during each phase of SDLC. They also proposed twenty one security rules along with validation results, hence showing their acceptability during SDLC.

PK.Ragunath et al., [9] in 2010, discussed the drawbacks of conventional SDLC models and proposed a hypothetical model to incorporate the user, owner and developer interaction by converting the conventional model into a three dimensional model, hence combating the issues such as change management, incident management etc.

M.Sharma, [11] in 2011, proposed an approach where selection of SDLC model is done according to different categories of project characteristics. Hence, project characteristics categories are classified and based on them SDLC models comparison tables are introduced.

V. Massey and K. J. Satao, [12] in 2012 developed a model to guarantee project development and delivery teams to have good co-ordination and collaboration which lead to enhanced productivity, efficiency, effectiveness and longer market life, hence incorporating concept of release management with basic SDLC phases.

S Balaji and M.S. Murugaiyan, [13] in 2012 discussed the various pros and cons of three SDLC models, waterfall, V model and Agile to summarize the steps, an organization would consider while making the selection of SDLC model.

N Kumar et al., [15] in 2013, in order to emphasize client satisfaction and to achieve the goal of high quality, cost effective product within schedule, proposed a new model so that client and developer can interact with each other to fulfill the requirements.

A.Mishra and D.Dubey, [16] in 2013 compared several SDLC models based on various features, according to which one can select correct SDLC model for a particular project, which is very important for a software industry for the software to be delivered with time limits with desired quality.

S. Patel, [17] in 2014 provided the concept of a check list which scans the analysis phase for the vulnerabilities present in a system, being designed. Also, the system suggests the ways to correct them, hence providing a less effort consuming, faster and secure system development.

3. SDLC MODELS

For the purpose of achieving different desired goals, many development life cycle models have been developed. The various stages for the same and their order of occurrence are specified by the models.

3.1 Waterfall Model

Waterfall is one of the simplest models to understand and use. This model is defined by a series of steps that must be accomplished in a linear or sequential order. Each phase is completed and verified before proceeding to next phase. Hence, the waterfall model is based on cascading the flow from one phase to another.

Waterfall model is not suitable for complex projects. It is not recommended for object-oriented software development, for long-term or ongoing projects, or for projects with unknown or variable requirements.

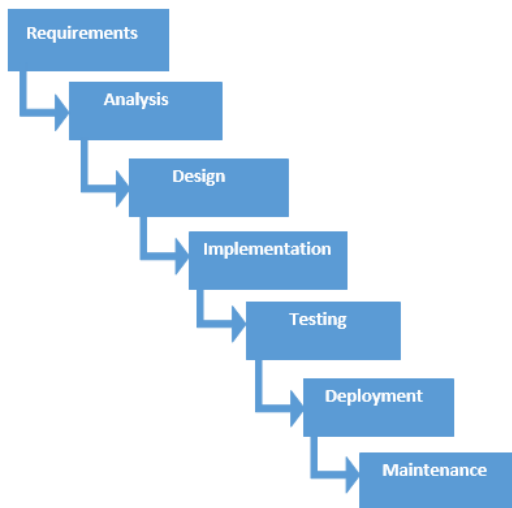


Fig 1: Waterfall Model [18]

3.2 V-Shaped Model

The V-Shaped life cycle model includes a procedural path of each phase that must be finished before moving on to the next phase. V-shaped model means Verification and Validation model. The initial test planning is the main difference between V-shaped development model and Waterfall development model. The development of the test plans (for each phase) takes place during each level of verification. After the completion of coding, testing is conducted as a part of validation.

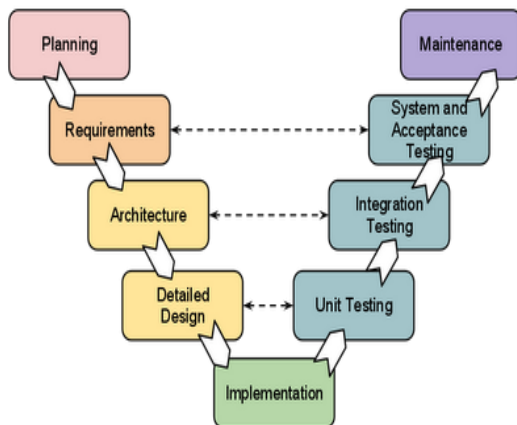


Fig 2: V-Shaped Model [19]

3.3 Prototype Model

Prototype model does not include finalizing the requirements before proceeding with the design or coding. Rather, to understand the requirements, a throwaway prototype is designed based on presently known requirements. It includes thorough interactions of clients with the prototypes to realize the requirements of the desired system in a better way. Complicated and large systems having no existing system for determining requirements are best suited for this prototyping concept.

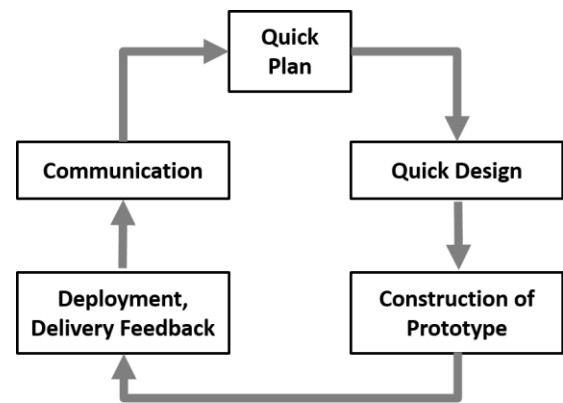


Fig 3: Prototype Model

3.4 RAD Model

Rapid Application Development (RAD) model is a methodology for software development that focuses on rapid prototyping hence favouring minimal planning. In this, the functional modules or the prototypes are developed in parallel, which are then integrated to fulfil the purpose of faster complete product delivery. Absence of preplanning makes the incorporation of any changes easier within the development process.

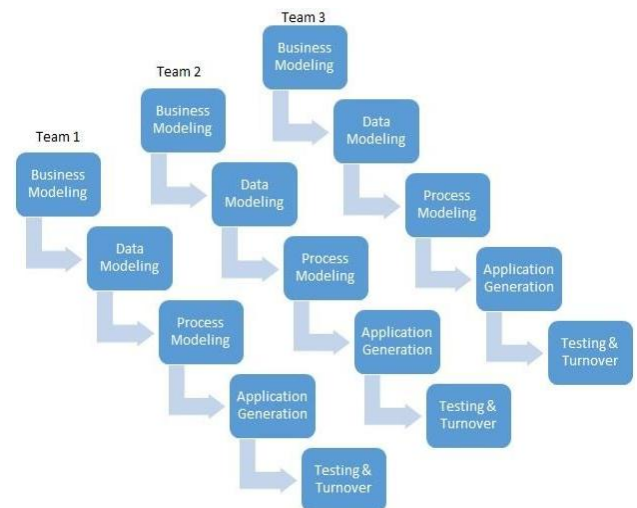


Fig 4: RAD Model

4. SDLC MODEL SELECTION TOOL

4.1 Overview

The SDLC Selection Tool will determine the best and most efficient SDLC model to be used on the basis of the project priorities entered and the present level capability of the organisation. The tool has two main components:

1. Comparison Metric Generation

Generate a comparison metric from the previous completed project data undertaken by the organization head.

2. SDLC Model Selection

Select a SDLC model to be used for the development of the project. This task is achieved by using the comparison metric generated and project priority.

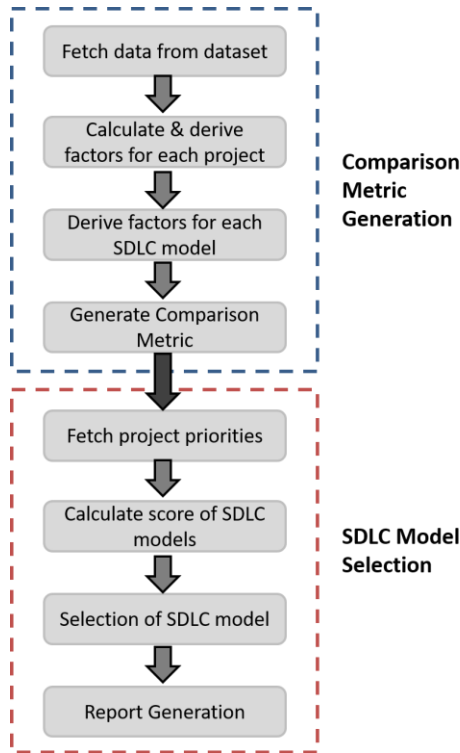


Fig 5: Flowchart of the Tool

4.2. Critical processes involved in the above mentioned components:

A. Factors derivation

The value of the factors for each project is derived from the project data according to the factor description.

B. Computing comparison metric

After derivation of factors with respect to each project undertaken, the next step is to generate the factors for each SDLC model. This is done by considering different projects using the same SDLC model and taking the mean value of the factors.

C. Score calculation

To select a SDLC model, the values of factors from the comparison metric are multiplied with the project priorities respectively, for each SDLC model. Among them the model with the highest aggregate score is suggested to be used by the tool.

D. Conflict resolution

A conflict may arise in selection of the SDLC model, if the aggregate score of more than one model comes out to be exactly same. In such case, the conflict will be resolved by selecting the model with the individual highest score of the factors compared in the order of the project priorities.

4.2.1 Factor Description

The factors used to formulate comparison metrics are described below:

a) Cost

Describes the cost incurred for the project per unit size. It can be simply derived by dividing the cost incurred by the size of the product.

b) Time

Describes the time taken for completion of the project per unit size. It can be simply derived by dividing the total time taken by the size of the product.

c) User involvement

Describes the level of involvement of the user during the development. It is the aggregation of the total time users were involved and dividing it by the product size.

d) Requirement Specification

Describes the nature of the requirements of the project. It can be calculated from the ratio of the number of requirements changed/added and the number of requirements stated beforehand.

e) Changes Incorporation

Describes the ease with which changes can be incorporated. It has 2 components: Effort and Money. They can be derived by monitoring the size of the changes along with the time and cost incurred to incorporate it. And then divide the two to get a per unit size estimate.

f) Maintenance

Describes the ability of the maintenance. It has two components: Effort and Money. They can be derived by monitoring the size of maintenance along with the time and cost incurred to implement it. And then divide the two to get a per unit size estimate.

g) Flexibility

Describes the level upto which model is customized according to the project. It is the ratio of the number of the processes undertaken in a specific project and the total number of the processes existing in the SDLC model used for development.

h) Simplicity

Describes the simplicity of the processes of development. It can be calculated from the difference of the effort spent on various model specific processes and the implementation and testing phases. Then, obtaining a per unit size value of the factor.

i) Guarantee of Success

Describes the probability of an error free product within schedule. It has two components, first being the ratio of the actual completion time and the estimated schedule time for the project. The second factor deals with the effort taken in testing process.

j) Reusability

Describes the reusability achieved in the model. It has two components: Effort and Money. They can be derived by monitoring the size of the changes along with the time and cost saved by reusing components and then divide the two to get a per unit size estimate.

4.2.2. Comparison Metric

The following table shows the comparison of Waterfall, V-Shaped, Prototype and RAD model with respect to each other.

Table 1: Software Metric for Comparison

Features	SDLC Models			
	Waterfall	V Model	Prototype	RAD
Cost	4	1	2	3
Time Duration	1	2	3	4

User Involvement	1	2	4	3
Requirement Specification	4	3	1	2
Changes Incorporation	1	1	3	2
Maintenance	1	2	4	3
Guarantee of Success	1	4	2	3
Flexibility	1	2	4	3
Simplicity	3	1	2	2

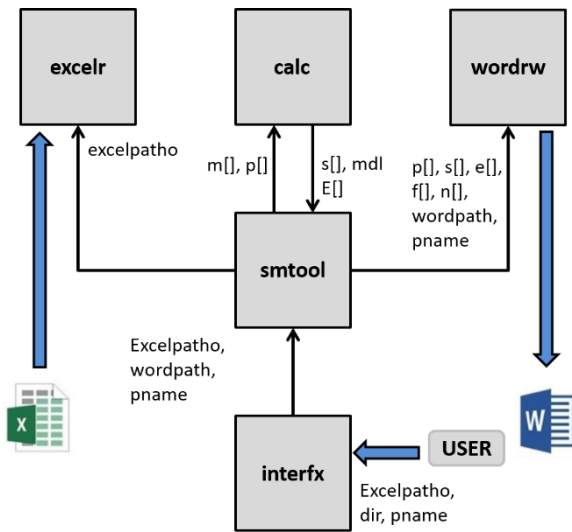


Fig 6: SDLC Selection Tool Block Diagram

5. RISK INCORPORATION

Risk is associated with each phase of the SDLC models. For this purpose, rather than a standalone activity, risk management activities are incorporated in the models itself. This ensures a robust SDLC model and efficient product. The following images depict some of the risks along with phases they can occur in for respective models.

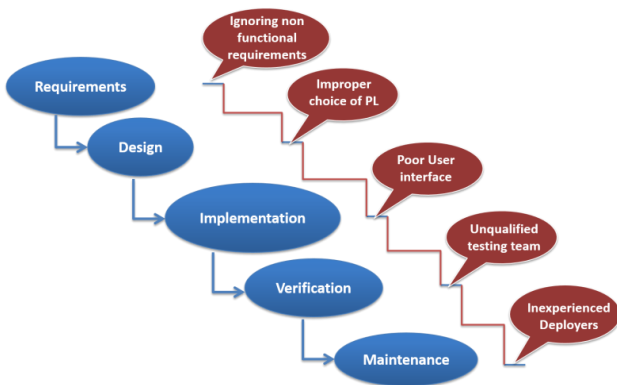


Fig 7: Waterfall Model with risk identification

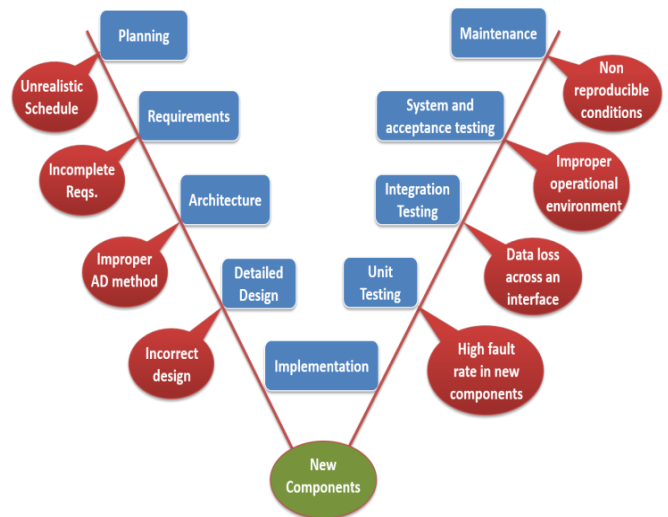


Fig 8: V Model with Risk Identification

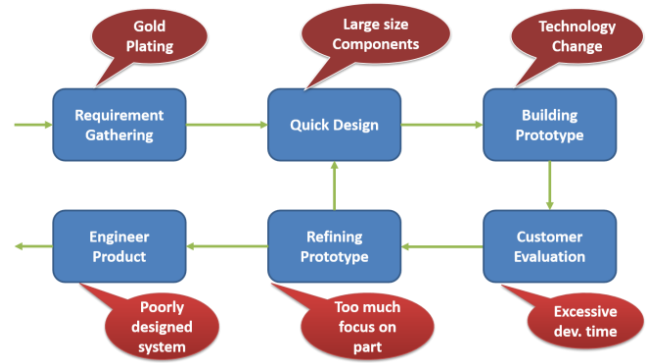


Fig 9: Prototype Model with Risk Identification

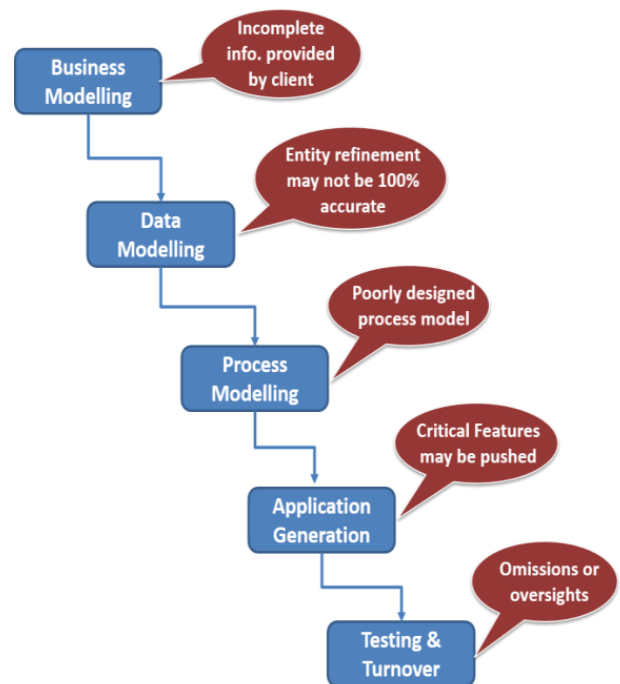


Fig 10: RAD Model with Risk Identification

6. CONCLUSION

The choice of SDLC model is crucial for a product's efficient development. In SDLC model, errors have a cascading effect hence, it is critical to identify and resolve the errors as early as possible in the development cycle.

The models used to develop software commercially like Waterfall, V-shaped, Prototype and RAD SDLC Models were discussed and compared. This comparison has been used as the basis to propose a tool that helps to select the best-suited model for the current project using the project priorities. Also, the risk management activities are incorporated in the models itself as compared to a standalone activity. This ensures a robust SDLC which in turn ensures that the product has been developed with an efficient use of resources.

In future, more SDLC models can be compared and an enhanced set of features/priorities can be considered. The tool can be deployed on a cloud or local network and made to be compatible with different file types in order to broaden its utility.

7. REFERENCES

- [1] Boehm, B.W. 1988. A Spiral Model for Software Development and Enhancement. IEEE Computer Society. vol 21, issue 5, (May 1988), 61 – 72.
- [2] Royce, W.W. 1970. Managing the Development of Large Software Systems: Concepts and Techniques. IEEE Computer Society. (August 1970), 1-9.
- [3] Davis, .A. M., Bersoff, H. E, and Comer, R. A Strategy for Comparing Alternative Software Development Life Cycle Models. IEEE Trans. on Software Eng. Vol. 14, Issue 10, (1988).
- [4] Jovanovich, D., and Dogsa, T. Comparison of software development models. Proceedings of the 7th International Conf, 11-13 June 2003, ConTEL 2003, pp. 587-592.
- [5] Tuteja, M., and Dubey, G. 2009. A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models. Int. J. of Soft Computing and Engineering (IJSCE). (July 2009).
- [6] Laura, C., Martinez, R., Mora M, and Alvarez, F.J. 2009. A Descriptive/Comparative Study of the Evolution of Process Models of Software Development Life Cycles. Proceedings of the 2009 Mexican Int. Conf. on Computer Science IEEE Computer Society. Washington, DC, USA. (2009).
- [7] Banerjee, C., and Pandey, S.K. 2009 .Software Security Rules: SDLC Perspective. IJCSIS Int. J. of Computer Science and Information Security. Vol. 6, No.1. (2009)
- [8] Kumar, D., and Hiranwal, S. 2010. Performance Enhancement of Software Process Models. 2nd Int. Conf. on Software Technology and Engineering (ICSTE). (2010).
- [9] Ragunath, P.K. et al. 2010. Evolving A New Model (SDLC Model-2010) For Software Development Life Cycle (SDLC). IJCSNS Int. J. of Computer Science and Network Security, Vol .10 No.1. (Jan 2010).
- [10] Wakchaure, A.M., and Joshi, S.D. 2011. A Framework to Detect and Analyze Software Vulnerabilities: Analysis Phase in SDLC. Int. J. of Advanced Computer Engineering and Architecture. Vol. 1, No. 1. (June 2011).
- [11] Sharma, M. 2011. A Survey of project scenario impact in SDLC models selection process. Int. J. of Scientific & Engineering Research. Vol 2, Issue 7. (July 2011).
- [12] Massey, V., and Satao, K.J., 2012. Evolving a New Software Development Life Cycle Model (SDLC) incorporated with Release Management. Int. J. of Engineering and Advanced Technology (IJEAT). Vol 1, Issue 4. (April 2012). 2249 – 8958.
- [13] Balaji, S., and Murugaiyan, M.S. 2012. Waterfall Vs V-Model Vs Agile: A Comparative Study On SDLC. Int. J. of Information Technology and Business Management. Vol. 2, No. 1. (June 2012).
- [14] Sabale, R.G., and Dani, A.R. 2012. Comparative Study of Prototype Model For Software Engineering With System Development Life Cycle. IOSR J. of Engineering (IOSRJEN). Vol 2, Issue 7. (July 2012). 2250-3021. 21-24.
- [15] Kumar, N. et al. 2013. Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction. Int. J. of Soft Computing and Engineering (IJSCE). Vol.3, Issue.1. (March 2013). 2231-2307.
- [16] A. Mishra and Dubey, "A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios", International Journal of Advance Research in Computer Science and Management Studies, Volume 1, Issue 5, October 2013
- [17] Patel, S. 2014. Security Integration in SDLC Analysis Phase", IJCSNS Int. J. of Computer Science and Network Security. Vol.14 No.12, (December 2014).
- [18] Massey, V., and Satao, K.J., Comparing Various SDLC Models And The New Proposed Model On The Basis Of Available Methodology.
- [19] Wikipedia.org/wiki/Waterfall_model
- [20] Tutorialspoint.com/sdlc/sdlc_v_model