

Rapid Revision Software Development Life Cycle Model (RR-SDLC Model 2016) Based on Concept of Reusability of Software

Mukesh Chauhan¹, Jawahar Thakur²

Department of Computer Science, HP University, Summer Hill Shimla (HP)^{1,2}

Abstract : Large open source Software applications based organization like mozilla , facebook, google, and AOL are migrating their product towards rapid revision based SDLC Model to release their product as update new revision. The same techniques can be applied to semi automation based software application or developing new application based on existing similar software application using re-usability concept. The newly developed application passes through three processes in parallel development mode to bring better product in shorter time at the lower cost as compared to following any traditional Software development life cycle (SDLC) model. This research investigates the changes in SDLC models after moving to Rapid Revision based software development life cycle model.

Keywords: RR- Rapid Revision, Software Testing, SDLC software development life cycle, AOL American online.

I. INTRODUCTION

A software development life cycle model has primary basic objective such as high quality product delivered in time that provides strong control on Quality, delivery and maximize the profit in terms of cutting cost on product development. It gives the process flow for software development to followed by software development organizations to organize their resource and work distribution in systematic manner from start to end of software project.

In order to meet these above objectives the various technical activities to be supported such as system definition, analysis, design, coding, testing, and performance reviews and developing project technical strategy to be adopted for software development life cycle model. Every SDLC model should use layered approach to doing tasks and creating output. Also it must organize the information such that the final product get developed, tested and released in time.

Every SDLC model should support various types of projects.

1. New software application Development
2. Upgrade the existing software product.
3. Package type conversion or compatibility
4. Conversion of existing system to new DBMS (portability of data from one database to another database)

The tradition SDLC model such as waterfall model, rapid application development (Prototype) -COCOMO model or Spiral model, Free flow model and recently newly invented models provides sequential or hierarchical development approach, which causes high cost and time in completion of project to release the final product as most of them suggest to start development at ground level of requirements to end of all the software requirements are

not meet. The parallel developed approach is not suggested not focused much on them.

II. EXISTING SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

2.1 SDLC Models

Every software projects undergoes defined stages regardless of whether small or large project which together are known as SDLC. There are various phases of SDLC

Identify Problem –Investigation Phase

Analysis – Solution to solve the problem

Design – Plan for solution

Implementation Phase - Maintenance or review for evaluating the results of the solution.

The basic concept similar to all SDLC model is output generated by each phase serves as the input to next phase.

The various SDLC Model [6] existing are as

1. Waterfall Model
2. Spiral Model
3. Incremental Model
4. RAD Model
5. Prototype Model
6. Extreme programming model
7. Free flow Model
8. Web Development Model

Only few of the above models have been explained to explain the concept of sequential SDLC Model.

2.1.1 Waterfall Model [1]

Waterfall model is the most used and well known software life cycle development model. It has a very simple layered

approach. Each next phase in this model must begin only after the previous phase is completed. The beginning of this model starts from requirements and planning. The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design (validation), Construction, Testing and maintenance. Small to medium database software projects are generally broken down into six stages:

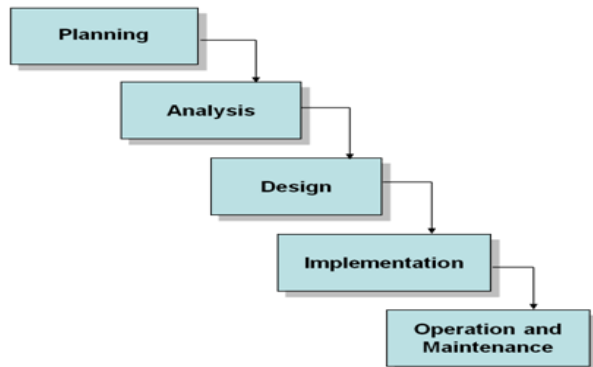


Figure 1.2.1

1.2.3 RAD Model [4] 1 Waterfall Model

RAD model combines the best available techniques and specifies the sequence of tasks that will make those techniques most effective. This model makes use of evolutionary prototypes that are eventually transformed into the final product. A set of CASE tools is used to support modeling, prototyping and code re usability, as well as automating many of the combinations of techniques.

It implements time boxed development that allows development teams to quickly build the core of the system and implement refinements in subsequent releases.

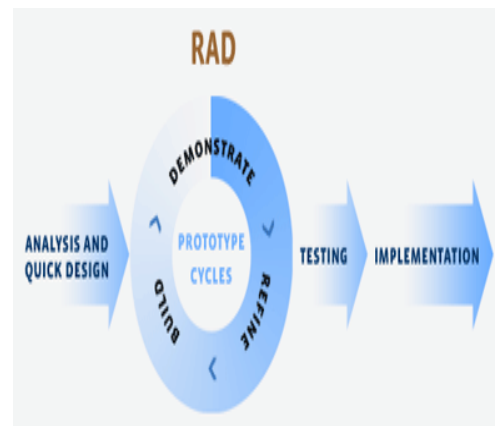
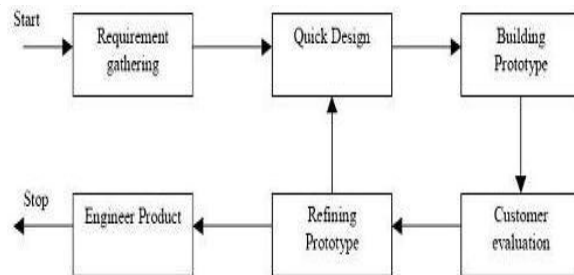


Figure1.2.2 RAD Model [4]

1.2.4 Prototype Model[5]

Instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. A prototype is a toy implementation of a system; usually exhibiting limited functional capabilities, low reliability, and inefficient performance.



Prototyping Model

Figure1.2.3 Prototype Model

1.3 Common Features-wise Comparison of various SDLC

1.3.1 Table of Common Features-wise Comparison of various SDLC Models[17]

Features	Spiral Model	RAD Model	Waterfall Model	Incremental Model	Pro to-type Model	Free flow Model	XP Model
Requirement Clarification	Understood In beginning	Time to time	Understood In beginning	Understood In beginning	Not Known fully	Not known fully	Fully Known
Schedule or Time Frame	Long	Short	Long	Very Long	Short	Long	Short
Cost	High	Low	Medium	Low	High	High	High
Manpower Utilization	Medium	Best	Poor	Medium	High	High	Medium
Re-usability	Yes	Very Less	Minimum	Yes	Yes	Very less	Not possible
Risk	Low	Low	High	Low	High	High	High
Incorporating changes	Easy	Easy	Difficult	Easy	Easy	Easy	Not Easy
Testing Time	Long	Long	Long	Long	Long	Long	Long
Complex city	Low	Medium	Low	Medium	High	High	High
Integration and Success	Good	Good	Low	High	Good	Low	Low

The above comparison gives us clear understanding of SDLC models and gives us the facts to state that there is need of more adaptable type SDLC model which can improve quality, re usability and rapid in releasing the software product.

III. RELATED WORK OR RESEARCH STUDY CONDUCTED

3.1 Research Papers published on “Evolving a New Model (SDLC Model-2010) For Software Development Life Cycle (SDLC)” [1] This paper clearly focused on the use of project control activities to be incorporated inside the software project in the form of owner, user and developer form and to enhance the product quality measured as no of testing error using this project life cycle.. It clearly implies that most of traditional SDLC use process flow that results in poor quality product and cost more money and time.

3.2 Research papers on “Open Incremental Model- An Open source Software Development Life cycle Model. (OSDLC)”[2] .This paper clearly explains the use of re-usability of open source software that leads to high quality product at shorter span of time, lower cost and also explains the possibility of development of software product using the study of similar existing product. The most important fact it gives us that involvement of more developer and expert can lead to develop better product as revise product.

3.3 Research Papers on “Dynamical Simulation Models for the Development Process of Open Source Software Projects”[3] The simulation results demonstrated the model’s ability to capture reported qualitative features of OSS and scope of future work should centre on Designing alternative OSS simulation models within the general framework, Conducting future case studies on OSS real-world projects with the purpose of collecting all the necessary data needed for accurately calibrating and validating OSS simulation models as useful for development of software projects.

3.4 Research papers on “Analysis and Tabular Comparison of Popular SDLC Models” [4] This paper examine the software flow as in every SDLC model suggest us that their exists defects of each SDLC model and gives us chances to vary that every SDLC is not best to every projects need.

3.5 Research Papers on “ Evolving a New Free-Flow Software Development Life Cycle model Integrating Concept on Kiazen” This research paper implies that if free flow of information is shared between user and developer to meet the requirement better way by satisfy the need of customer till they are satisfied with the product as free flow between start to end and end to start or top to bottom ot bottom to top of development work.

The various Research papers, case studies and reference books above clearly focus on selecting proper SDLC

model for developing software projects and also they states that every model of SDLC is not suitable to every project. One most common factor among all study also is releasing the product in time, reduced the cost and providing more satisfaction to acquirer of product. Every Software project need to be tested against the requirements and size of project against SDLC Model used. The study of every component SDLC model gives us understanding that which component is useful or best fitted in project various stages to maximize the utilization of resources. Following any pre developed SDLC model is not solution of every software project. The Focus is on the acquisition of quantitative information and use of this to control the process methods which can help us to identify the problem present in existing models.

Objectives Of My Research

- 1.To Study and compare the existing SDLC models and find the Common weaknesses of each SDLC Model.
2. To Develop the Frame work for Proposed Rapid Revision based SDLC Model on the concept of re usability of existing similar software.
3. To develop a case for further studies and performance evaluation of proposed RR-SDLC Model Practically.

IV. PROPOSED RAPID REVISION BASED SDLC MODEL

The Proposed RR -SDLC Model emphasizes on Developing product from some similar old product or add new components in it as to meet the requirement of dissimilarity by modifying the existing similar product or adding new component. It is a parallel three step approach as explained in figure below

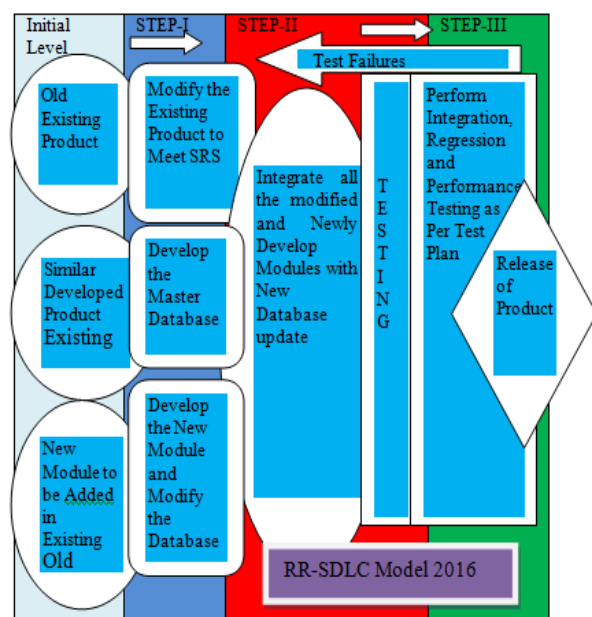


Figure 3: New Proposed RR -SDLC Model

V. FEATURES OF PROPOSED RR-SDLC MODEL 2016

- a) The new proposed hypothetical model issues various

common weakness of traditional sequential model such as parallel development approach.

b) The issue of developer understanding existing system as user like involvement in it and modify the concept as per requirement of existing and new product.

c) Development and testing can start simultaneously as parallel process.

d) Portability of old and new data is one of the major advantages as old volume data helps in testing the product performance better and improve the overall product quality.

e) It is kind of light weight technology helps in reuse the existing similar product as involvement of more professional results in better concept and implementation.

d) Apart from this it involves project level attributes inside embedding as control flow, release management and user/owner and developer in much free manner.

e) The project estimate is much better as workload is prior estimated helps in time delivery of project and better customer satisfaction.

VI. IMPLEMENTATION OF PROPOSED RR-SDLC MODEL 2016

The Implementation is done as parallel comparison between existing research done on SDLC models and with best data drawn from that research and projected data based on my proposed model. The tested failure results analysis has been done on two software one on Automotive spare parts cost optimization software and another academic Manager HR Software done by STQC Centre of reliability Chennai India . Based on their test data helps in parallel support and validation of my newly Proposed RR- SDLC model.

VII. RESULT AND DISCUSSION

New Proposed RR-SDLC has been parallel reviewed with existing released model SDLC 2010[19] and applied review for same validated data and projected data of RR-Model for same software application

- a) Spare parts cost optimization
- b) Academic –HR Manager

The RR-SDLC has been compared with Traditional SDLC Models, SDLC Model 2010 and the below table explains the merits of comparison between all the models and RR-Model on Test Cases Failure.

The chances of error due to error handler, configuration failure, Infrastructure and user awareness also get reduced as we are updating the existing product to meet the requirement of software. But the sequential models is a kind of theoretical approach not to project specific

approach in reality every programmer and manager uses his prior experience he/she is having i.e reusability in the form of indirect system of software projects, the same concept if systematically performed as proposed in my RR-Model that will not only enhance our product reliability but also focus on our old exiting valuable data of many years. None of the software model explains porting of existing application data into new software developed.

Table7.1 Test Failure Comparison between other existing SDLC Models and RR-Model as Best Case and Worst case

S. No	Name of Modules or Cell	Projected No of Failures using Tradition SDLC Model	No Of failure Using SDLC 2010 Model	Projected no of failure using RR-SDLC Model	Re marks
1	Security Module	4-8	0	0-1	
2	File Module	6-12	0	0	
3	S/w STD.	6-12	1	0	
4	Database	4-8	0	0	
5	Reports	3-6	0	0	
6	Error Handling	7-14	1	0	
7	Repetitive Failure	13-26	0	0	
8	Configuration Failure	23-46	2	0	
9	Version Failure	3-6	0	0	
10	Infrastructure Failure	7-14	1	0	
11	Awareness Failure	3-6	1	0	
	Total No. Of failures	79-148	6	0-1	

VIII. CONCLUSION

The above research study gives clear knowledge and understanding about advancement in SDLC models that are used for developing software, which can give better results as compared to sequential SDLC model.

The proposed SDLC Model can be considered as an approach for creation of new SDLC model that is more practical and project specific that will helps software manger to manage his software project better manner to fulfil the objectives of his organization. It can also be summarised as advancement in software engineering concept for next generation academicians for study and further explore RR-SDLC Model in order to schedule and estimate software project more precisely and accurately.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my advisor **Jawahar Thakur** Associate Professor, Computer Science Department HP University for his guidance and support. His extreme knowledge, creativity and excellent skills have always been a constant source of motivation for me.

He is a great person and one of the best mentors, I always will be thankful to him. I would also like to thanks to Chairman HP University Department of Computer Science and other faculty members of his team for guiding us time to time and provide us ample opportunities and flexibility to imparting and sharing knowledge with us. I would also like to thanksS to all my colleagues of M Tech. (CS) for devoting their time in discussing ideas with me and giving their invaluable feedback. I would like to dedicate this Paper to my amazingly loving and supportive parents who have always been with me, no matter where I am.

REFERENCES

- [1] S. Bhatacharjee , “ Software Development Life Cycles (SDLC)”, MOF, College of Agiculture RBI 2009
- [2] IEEE Std 1047-1988 IEEE standard for Developing Software Life Cycle Processes IEEE Computer Society 1989
- [3] Adrian Bachmann and Abraham Bernstein, Data Retrieval, Processing and Linking for Software Process Data Analysis. Technical report, University of Zurich, Department of Informatics, 2009.
- [4] Pankaj Jalote. An Integrated Approach to Software Engineering. Third Edition, Narosa Publishing House, 2005.
- [5] Bennat P. Lientz AND E. Buton Swanson. Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations. Addison-Wesley, Reading, MA, 1980.
- [6] Software Engineering theory and practice Fourth Edition by Shari Lawrence Pfleeger and Joanne M. Atle Pearson Publishing
- [7] Principal of Distributed database system , third edition by M. Tamer Ozsü , Patrick Valduriez, Springer Publishing.
- [8]<https://www.google.co.in/search?q=waterfall+lifecycle+model&biw=1024https>
- [9]<https://www.google.co.in/search?q=spiral+model+in+software+engineering&biw=1024&bih>
- [10]<https://www.google.co.in/search?q=incremental+model+in+software+engineering&biw=1024&bih=696&tbm>
- [11]<https://www.google.co.in/search?q=rad+model+in+software+engineering&biw=1024&bih=696&tbm>
- [12]<https://www.google.co.in/search?q=prototyping+model+in+software+engineering&biw=1024&bih>
- [13]<https://www.google.co.in/search?q=extreme+programming+model+diagram+in+software+engineering&biw=1024&bih=696>
- [14]<https://www.google.co.in/search?q=web+designmodel+diagram+in+software+engineering&biw=1024&bih=696>
- [15]<https://www.google.co.in/search?q=Free+Flow+model+diagram+in+software+engineering&biw=1024&bih=695>
- [16] www.ipajournals.com
- [17] www.BenderRBT.com
- [18] Evolving New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction, International Journal of soft Computing and engineering (IJSCE) ISSN:2231-2307, Volume-3, Issue -1, March 2013 ,Naresh Kumar , Prof. A. S. Zagaonkar
- [19] Evolving New Model (SDLC-2010) for Software Development Life Cycle (SDLC), International Journal of soft Computer Science and Network Society (IJCSNS) Volume-10, No 1 Jan 2010 ,PK RaguNath, Prof. S. Velmourugan