Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 05, May-2016

Cost and Delivery Optimization using Rapid Revision Software Development Life Cycle (RR-SDLC March 2016)

Mukesh Chauhan
Computer Science Department
HP University Summer Hill Shimla (HP)

Jawahar Thakur
Department of Computer Science
HP University Summer Hill Shimla (HP)

Abstract :**The Biggest challenge for Software Development Firm's is to estimate the customer's application problem size and complexity followed by estimate cost and project delivery schedule with high percentage of accuracies prior to start of any software project. The same can be later verified with actual data found while developing and delivery of product. None of the firm's has unlimited resources and all the tools available to meet these challenges. This research firstly investigates the study of existing methods and tools used for software project development, size estimation and its limitation for meeting the software project size estimation in terms of accuracies of projected estimated cost and delivery. We have proposed an alternate technique of development called reusability of existing software based concept model called RR-SDLC model 2016 and estimation of cost and delivery based on this concept of development. Based on comparison with existing practices and method cost estimation and delivery it has been found that RR-SDLC Model based development not only enhances the accuracies in terms of estimation of cost and delivery but also optimize the both in context to software development project.**

Keywords: SDLC Software Development Life Cycle model, RR-SDLC Model 2016.

I INTRODUCTION

Software engineering has gone through various changes in the process of developing and testing software product. The software development imposed lots of challenges today due to high level of competition and customer awareness about the product. Every development firm has to deliver the project as early as possible to meet the customer's requirements due to change in technology imposing -challenge every day. Now the twenty years back theory is almost obsolete, which is study application and supply partial product followed by supply of other modules later as earlier attachment. It is proven fact that none of software project was completed and it become obsolete with new technology before completion. Now we are working with web enabled application type networking and working together through internet and require data from many servers together and many users working simultaneously. Big data, meta-data, cloud computing, software securities and data mining are also part of your software development application. So the complexity is added in software application as external with internal complexities as earlier, Size of software requirements estimation is a big challenge for software manager. Inaccuracies in Software estimation and complexities leads to product loss in terms of development cost and customer satisfaction. The futurist business is also suffered indirectly by this method. It should support various types of projects.

So the project cost and delivery optimization depends on the various available software project size estimation techniques are divided into two basic categories

1) Tradition methods known as Software Metrics to measure Size of application.

2) Recently developed method known as software estimation method based size estimation techniques.

In tradition methods includes size based metrics such as productivity measures and Measurements of Lines of Code and point based metrics such as Function Point or object point or test point etc.

The recently Developed methods are software size estimation such as Delphi Method, CoCoMo Based method or Work Breakdown structure (WBS). We have explored various size estimation methods and research studies and found that most of the methods are project and domain specific and none of them is generalized method to apply freely to any software projects. The few basic software project process model are discussed here only to clarify the reader about the concept of size estimation of software project.

II EXISTING SOFTWARE PROJECT PROCESS MODEL

2.1 SDLC Models

Every software project has defined stages such as
Problem Investigation Phase
Problem Analysis Phase
Design phase
Software Coding and Testing Phase
Implementation Phase
Maintenance and Review Phase

The basic concept behind every SDLC model is it a sequential process and one stage serve as input to next stage. The SDLC Model [6] existing are of two types

Traditional Model (TR-SDLC Models) Such as

1. Waterfall Model
2. Spiral Model
3. Incremental Model
4. RAD Model
5. Prototype Model

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 05, May-2016

Recently Developed Open Source Model are such as

1. Extreme programming model
2. Free Flow Model
3. Web Development Model

These Models have been developed to improve or overcome drawback of existing model in the new model. None of the models have been focused on product cost and delivery optimization techniques.

Traditional Model

The traditional model is based on sequential approach as when one phase is completed the project move to next phase and continue the process till the product is released as shown in the diagram below

Figure 1.1
Recently Developed SDLC Models

Due to competition in software development fields and advancement in computing methods the new application can be developed as latest techniques using remote development and off-time development method or free flow development models as shown in the figure below

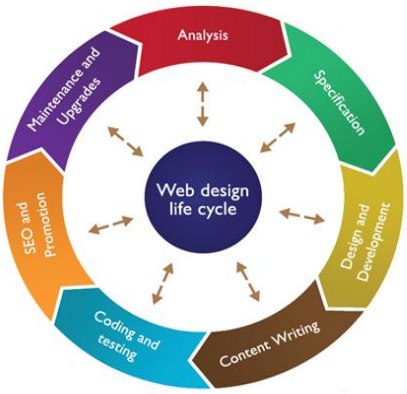

Figure1.2 emote or Web Development Model [18]

In this free flow development or remote development model analyst, Designer, coder and tester are connected by internet and reside anywhere in world and work on same product as web design life cycle model.

Comparison of various SDLC On the Basis of Cost and Delivery Estimation.

When the comparison of various SDLC model is done based on the following parameter as shown in the table

Table2.1 of Comparison of various SDLC Models [17]

Features	-Models	Recent Developed Models
Size Estimation	Based on application to application.	Based on application modules.
Cost Estimation	Approximation method based on problem size	Approximation method based on module size
Re-usability	Impossible	Impossible
Risk	High	High
Complexity estimation at Initial stage	Difficult to Estimate at Initial stage.	Difficult to Estimate at Initial stage
Delivery Estimation	Based on Size Estimation pr similar project size.	Based on Size of each module Estimation,

The above comparison gives us clear understanding of failure to estimate the problem size and complexity using TR-SDLC models and gives us also 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 and can result in optimization of cost and delivery.

III Related Work or Research Study Conducted

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.

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. 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 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.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 05, May-2016

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 very that every SDLC is not best to every projects need.

5. Research Papers on " Evolving a New Free-Flow Software Development Life Cycle model Integrating Concept on Kiazen"[18] 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. 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.

IV Objectives Of My Research

1. To Study and compare the Traditional SDLC models and recently developed model on the basis of estimation of cost and delivery
2. To apply the estimation of cost and delivery on Proposed Rapid Revision based SDLC Model 2016 based on the concept of re usability of existing similar software.
3. To develop a case for further studies and performance evaluation of RR-SDLC Model 2016 for optimization of cost and delivery of software practically.

V Rapid Revision Based SDLC Model 2016

The proposed RR -SDLC Model 2016 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 the block diagram Figure

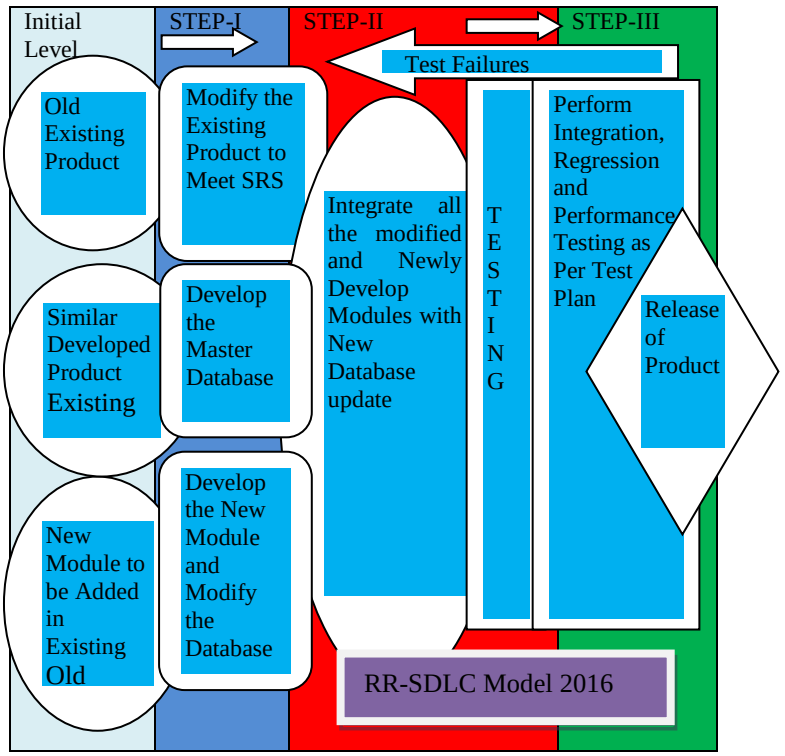

Figure 5.1 New Proposed RR -SDLC Model 2016

5.1 Features of New Proposed RR-SDLC

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 timely delivery of project and better customer satisfaction.

5.2 Implementation of Cost and Delivery Estimation of on Proposed RR-SDLC Model 2016

We apply CoCoMo model for estimation of cost and delivery to TR-models, recently developed model and RR-SDLC Model 2016 and perform the comparison between them validate the hypothesis.

VI. Data Results

When CoCoMo Model are used for comparing Traditional Model and proposed model.

Assume a project of three types
Small non complex Project= 1000 Lines of Codes with Less Complex in form.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 05, May-2016

Medium Range Complex Project (Medium Team Project) =125000 Lines Of Codes with more Complex form than small non complex project.

Large Complex Project =More than 1000000 Lines of Codes and highly complex constraints.

Now when using traditional model the Standard lines of Codes remains same.

When using reusability of existing similar product the SLOC ret reduced based on following type of application Development using existing similar domain Application = SLOC reduced from 50% -80% as required, when taken average 75% i.e. ¼ SLOC for Development.
Non-Similar Domain= SLOC reduced from 20% -50% as required, when taken average 35% i.e. 2/3 SLOC for Development.
On the average from ¼ SLOC -2/3 SLOC is nearly equal to ½ SLOC for Development.
These data are projected based on open source project development applications.

a> Using Basic CoCoMo Model (Small non complex Project)[20]
According to the basic CoCoMo, the following can be computed as
Effort Applied (E) = ab(SLOC)bb [total man-months]
Development Time (D) = cb(Effort Applied)db [months]
Manpower Required (P) = Effort Applied / Development Time [count]

Software project	ab	bb	cb	db
Small non complex Project	2.4	1.05	2.5	0.38
Medium Complex Project	3.0	1.12	2.5	0.35
Large Complex Project	3.6	1.20	2.5	0.32

Where, SLOC is the estimated number of statement lines of code some time expressed as KSLOC in thousands of code lines for project. The table below shows the projected data

Table 6.1 Test Result data using Basic CoCoMo (Small non complex Project)

Sr. NO.	Parameters	Tradition SDLC Model Data	Proposed RR-SDLC Model Data
1	Efforts	100X	50X
2	Development Time	10Y	5Y
3	Cost	1000XY	250XY

X,Y is Multiplication Factor in terms of above table

Since as per CoCoMo Basic model efforts and Development time is directly proportional to SLOC. The cost is product multiple of both effort and development time.

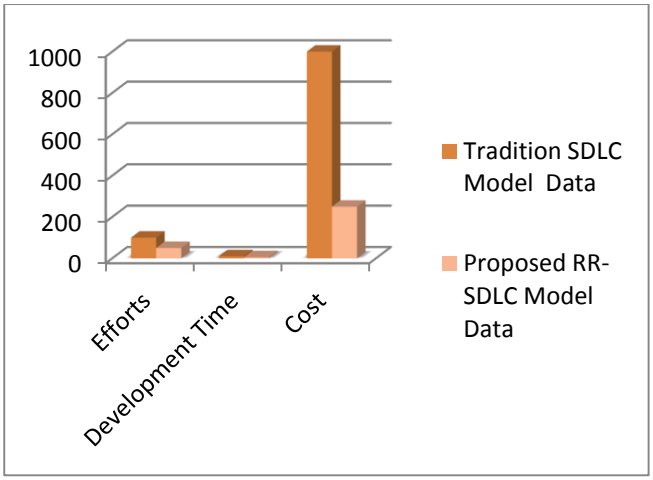

Figure 6.1: Graphical Representation of data using Basic CoCoMo(Small non complex Project)

b>Intermediate CoCoMo (Medium Range Complex Project) [20]

Intermediate CoCoMo estimate and computes software development effort as function of program size and other cost drivers. The product of all effort multipliers results in an effort adjustment factor (EAF). Typical values for EAF range from 0.9 to 1.4.
The Intermediate CoCoMo the effort can be calculated or computed as E=ai(SLoC)(bi).EAF where E is the effort applied in person-months,
SLOC is the estimated number of thousands of delivered lines of code for the project, and EAF is the factor calculated above.

The coefficient ai and the exponent bi are given as below

Software project	Ai	bi
Small non complex Project	3.2	1.05
Medium Complex Project	3.0	1.12
Large Complex Project	2.8	1.20

The Development time D calculation uses E in the same way as in the Basic CoCoMo.

Table 6.2 Test Result data using Intermediate CoCoMo (Medium range Semi-complex Project)

Sr. NO.	Parameters	Tradition SDLC Model Data	Proposed RR-SDLC Model Data
1	Efforts	500X	250X
2	Development Time	50Y	25Y
3	Cost	25000Y	12500XY

X,Y is Multiplication Factor in terms of above table

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 05, May-2016

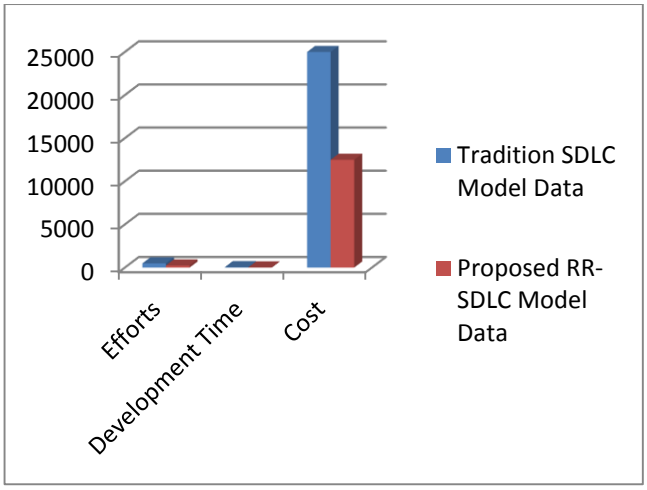

Figure 6.2 Graphical Representation of data using Intermediate CoCoMo((Medium range Semi-complex Project)

Since as per CoCoMo Intermediate model efforts applied and Development time is directly proportional to SLOC. The cost is computed as in Basic CoCoMo Model and is product multiple of both effort applied and development time.

c> Detailed CoCoMo [20] (On Large complex Project)
Detailed CoCoMo can take into consideration of all the characteristics of the intermediate CoCOMo version with its impact of as cost driver's on each step step of the software development process. The detailed model uses cost estimating relationship or project as in the form:
$PM_{NS}=A \times Size^{E} \times \pi_{i=1}^{n} EM_i$
where "Size" is one the main additive drivers in which one is the SLOC, and EM represents one of multiplicative effort multipliers.

Table 6.3 Test Result data using Detailed CoCoMo (On Complex project)

Sr. NO.	Parameters	Tradition SDLC MODEL Data	Proposed RR-SDLC Model Data
1	Size	100000XY	500000X
2	Estimated Cost	1000000XYZ	500000XYZ

X,Y,Z is Multiplication Factor in terms of above Size Additive Drivers , Effort Multiple Drivers, Project Related Driver.

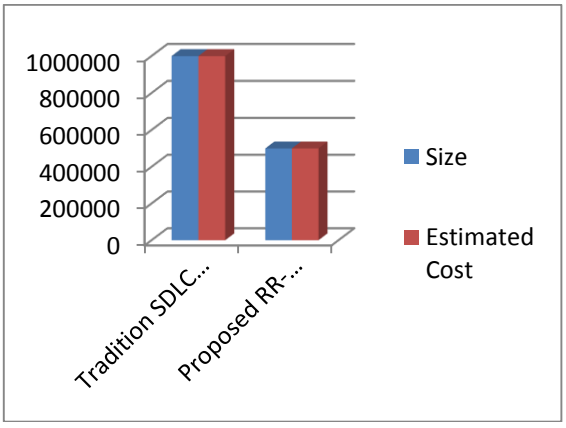

Figure 6.3 Graphical Representation of data using Detailed CoCoMo(On Large complex Project)

When Applied the Data to Basic CoCoMo for delivery time i.e. Development time which is also ratio of cost is to size which will be proportionally reduced to half as compared to Traditional Models.

VII. Discussions

Based on above cases results it clearly validated my hypothesis of cost and delivery optimization using RR-SDLC model 2016. The result data above shows that developing applications using Rapid Revision based concept of reusability not only gives better product but also reduced cost and delivery of release of product. The major concept of RR-SDLC states that development is an ongoing and consistent process along with the successive releases of revision modules of the whole software.. If the same software is to be developed by traditional model say waterfall model then due to its architecture the software will be released then only the client will be able to know the limitations of the software and if change needed then again the whole process of SDLC will have to be followed. Due to architecture of RR-SDLC model 2016, the client will be immediately informed about the development and the needs or changes will be informed back to the developer. Thus repetition of SDLC phases is minimized leading to decrease in the total number of LOC (lines of codes). The basic objective of this research was to develop a standardized methodology for software development in the very unique industry and culture to meet the project requirements. The reusability allows the development team to deliver for revise the existing product as intermediate-level based upon clearly defined customer requirements.

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 2016 in order to schedule and optimize software project estimate of cost and delivery 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 his all 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 thanks to all my colleagues of M.Tech. (CS) for devoting their time in

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 05, May-2016

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 Agiculkture 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 Engineeering theory and practice Fourth Edition by Shari Lawrence Pfleeger and Joannne M. Atle Pearson Publishing
[7] Principal of Distributed database system , third edition by M. Tamer Ozsu , Pattrick 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. Velmourougan
[20] Software cost Estimation by Nancy Maelo Schett and Dr. Martin Sinz University of Zurich Swiss.