An Analysis of Software Process Model for Minimize the Software-Development Issues

Mohamed Azharudheen.A[1], Natarajan.S[2]

[1,2] Assistant Professor, Department of Computer Applications,
Thanthai Hans Roever College, Perambalur, Tamil Nadu, India

Abstract: ***The term "Software Engineering" means the application of an experimental, closely controlled and logical approach to the development and maintenance of software. Any company when going to develop software according to our software engineering definition, they would follow someone of the process model. In this paper, we will take two process models waterfall model and RAD model. These two models have the following issues the first model has a long time to finish development, and the second model has a high development cost and bugs these issues will make huge economic losses to the company. This paper aims to review these two process models and make some changes in that model for minimizing the above issues.***

Keywords: Software Process Model, Software Process Model Issues, Minimize the Software development Issues, New Skill process model, Develop good software.

1. Introduction

The software engineering is a sequence of tasks. The software community has continually attempted to develop technologies that will be quicker, less expensive and make it easier to construct and continue high-quality software's. In the last 50 years, the computer software has many changes in Computing Architecture, Memory and Storage capacity and so on. However, they can also pose huge problems for those who must build complex systems. To avoid these difficulties the software developers have needed some models before they are going to develop software's. That model is called a software process model. The software process model provides interaction between users and designers, between users and developing tools, and between designers and developing tools. The software process model as a framework for the tasks that are required to build high-quality software. The use of framework activity is to meet the development goals. In this paper, we have taken two process models that are waterfall model and RAD model these two process models have some advantages and disadvantages also. So we collaborate these two models and form one new process model for minimizing our software-development process issues.

2. Software Development Process Model

Set your Software's are developed or engineered it is not manufactured. Because software's are instructions that when executed to provide desired features, functions and performance. We said software's are engineered so all the engineering works are starting with the designs or blue print for our sequence of tasks. Like that when develops software's also we would create a model that would call software process model. When we build up any process model, we have following three major issues.

2.1 Issues in software development process

1. To take so long to get software finished.
2. Development cost so high.
3. Cannot find all errors before we deliver the software to customers.
4. Why this issue has happened.

2.2 Time delay

When software development has taken a long time it will not satisfy the customer. The software developers take long time because of the following reasons.

- Not having enough developers.
- Some team members waiting for other team members to completed a dependent task.
- "Blocking States" in which the developments done after one step is over.

2.3 High Cost

All the customers are needed to finish their project in a couple of months, so the company needs more human resources to complete this task and also the team of developers work 24 hours then only they achieve their goals. However, when the company has more human resources, they have not a sufficient work environment in this situation, they split their teams for daytime worker and night time workers, and also they need many resource persons this factor is increasing our development cost.

2.3 Bugs or Errors

When developing software the debugging is a very vital role. In some cases, we cannot find all errors before we deliver the software to the customer. This will happen for the following reasons.

- The software community doesn't understand their clients need.
- The product scope is poorly defined.
- The chosen technology changes.
- Managers keep away from top practices and training scholarly.

3. Waterfall Model and RAD Model

Now we are going to see about advantages and disadvantages of waterfall model and RAD model and how can we solve our three issues.

3.1 Waterfall Model

The figure.1 represents the waterfall process model.

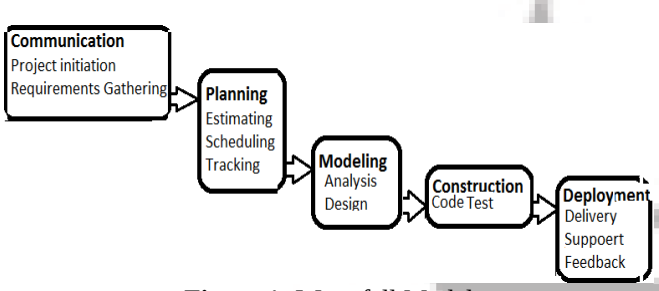

Figure 1: Waterfall Model

The Waterfall model is a systematic and sequential approach to software development that begins with customer specification of requirements that is an advantage of the waterfall model because the developments are done after one step is over so the customer requirements are achieved because of this when using the waterfall process model; we get some little bugs only. The waterfall model leads to "blocking states" in section 2.2 we have already seen about this for this reason the waterfall model took a long time to develop software.

3.2 RAD Model

The RAD model (Rapid Application Development) is a short development cycle. It is a "high speed" adaptation of the waterfall model; it is done by using a component based construction approach. Like a waterfall model, the RAD model also has drawbacks, the RAD requires sufficient human resources to create the right number of RAD teams, and if we have not been sufficient developers in the RAD team, then the project will be failed. The development cost is high in RAD model according to our section 2.2. The figure.2 shows the RAD process model.

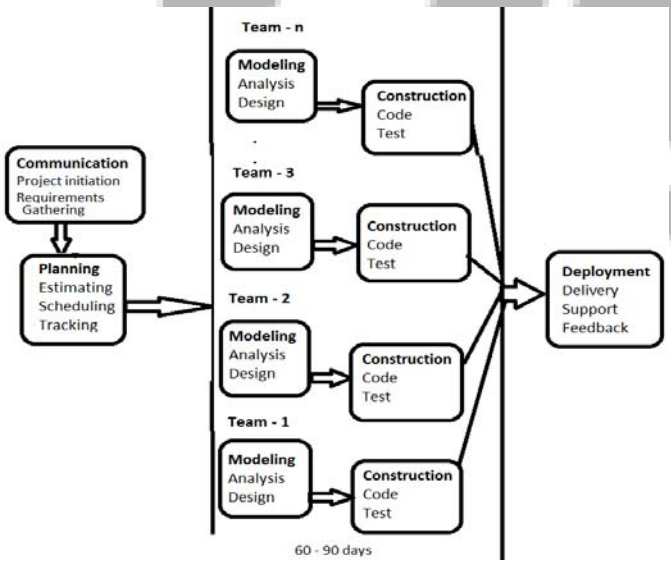

Figure 2: RAD Model

4. New Skill Process Model for minimizing the issues

From the above two models, we have some advantages and disadvantages also the waterfall model will satisfy the customers from that we get only small bugs only because it has a "blocking state" but it takes long time approximately 360 days compare with RAD. In the RAD model, we can complete our tasks in a very short period within 60-90 days, but the development cost is high, so we collaborate these two models and introduce some newer techniques in that and create a new process model.

4.1 Collaboration of Waterfall Model and RAD Model

To collaborate waterfall model and the RAD model, we can reduce the development time and bugs. Because the component of the RAD model will reduce the time delay, and the waterfall model will reduce the bugs using the handiness of complete every step.

4.2 New Skill for minimizing cost

Look at the section2.3 it will show why the development cost is high, the companies pay out most of the money for the developers and the office maintenance, if we reduce these expenses, we can minimize our development cost, so we replace the team of developers by student teams that is our new skill idea for reducing our costs, the following diagram shows our collaboration, and new software process model.

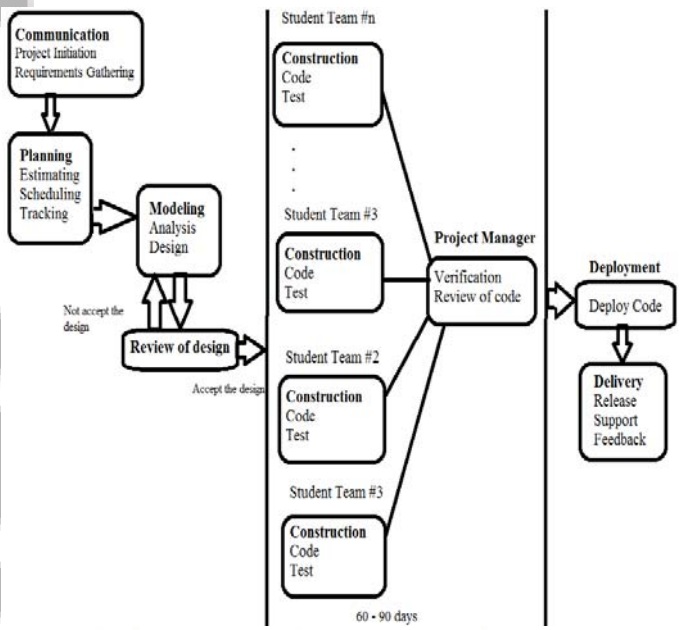

Figure 3: Ne Skill Process Model

The figure.3 shows the new skill process model. Here we collaborate waterfall model and RAD model, and also we introduce our new skill components this will minimize the software-development issues. The process starts with communication, the team of the project initiators gathering the customer needs, then the process moved to the planning team. The team of planners will be prepared the estimate that will refer the development cost then the team makes a scheduling that will define the time period of project

development and define which component has how long for complete their tasks. After completion of planning the process will shift to the next process level called modeling, the modeling has a vital role in the software process model, because if the team of modeling persons will construct a design for software development if they did any mistake it will affect the entire development. So the modeling team has a responsibility to construct a good design, which will help with the implementation team for developing bug free software.

4.3 Reviews of Design for minimizing Bugs

A number of industry studies (by TRW, NEC, Mitre Corp., among others) indicate that design activities introduce 50 and 65 percent of all errors during the software process, for debug these errors the company will spend some more money so it will increase the development cost also. In our new skill process model, we adopt a new skill for solving this error that skill component is called as review of design, this component "purify" the design activity. After completing the design the modeling team will go to arrange the presentation meet for review that design, the figure.4 shows the review of design team members.

Figure 4: Review of design Members

The review presentation of software design to an audience of customers, management, and technical staff is also a form of review. Sometimes it is also called as an inspection. Review of design is a "filter" for minimizing design mistakes; the review of design is an effective means for uncovering errors and also minimize the cost. When the design would satisfy the management, customers, and technical staff then only the review team allows that design for further development, otherwise the team will reject that design, the modeling team then reconstructs a new design, after the construction also they conduct a review of design, this is an iteration process while the review team agrees to the design, when the design gets an approval from the review team, the development step up to next component called construction or implementation. In implementation level also we introduce a new concept for minimizing our costs.

4.4 Student team for Minimizing Cost

Before begin the implementation the project leader should track the following process.

- Assign the modules to which team can going to handle which module.
- The project manager should been choose the exact technology for developing the particular software, after choosing the technology don't change it in middle of the development. It will squander our time and cost.
- The project manager analyze, what type of resist come from users side when we release this software, better to avoid that modules or better way to get feedback from team members and customers.

After did these three processes, we will move on to the next component called implementation. In the software process model implementation is a very crucial, critical, important and also very vital role. In any software development, construction process has 40 – 50 percent of time to implement coding. Here we need more developer's knowledge to write the code. We all known all the software development process has take long time for completing the code implementation. In waterfall model the construction will contain a single team and that team will implementing coding but it has very long time approximately 360 days with minimum bugs, but in RAD model the construction team split many teams they share the modules and develop the code for that particular module only so the code implementation will completed in short period approximately 60 – 90 days with many bugs. In RAD model, the development cost is also high because the RAD model needs sufficient human resources. We already add some one new skill called "review of design" that will used to minimize our bugs, now we have two issues namely time delay and high cost, if we use waterfall model it will reduce cost but the development time so long, if we use RAD model it will reduce time but the development cost is high. To solve this problem we are going to introduce one more new skill in our new skill process model that is, include the student team with project manager team. When using RAD model, the construction process had split into many teams that team members are completed their modules and submitted to the project manager, when the project will be divide into many modules that will be finished quickly. Each one team has the following person's one project head, one project leader and many developers. The project manager would be managing all the teams.

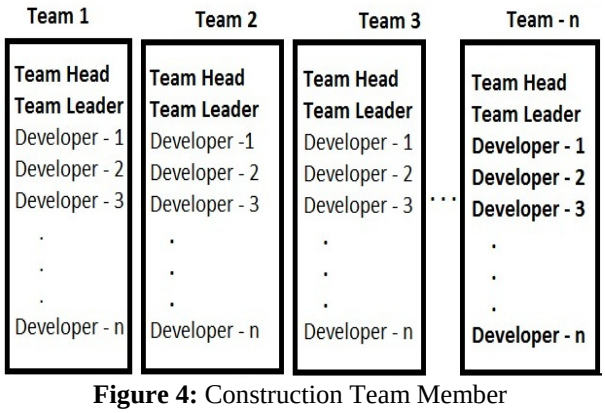

Figure 4: Construction Team Member

The figure 5 shows the team members of each team in a construction component in RAD model. From figure.5 we can understand the company spends most of the money for paying developers. This is the right place to introduce our innovation. All the final year master degree computer science students have a main project. The company can make use of them as developers. The figure.6 shows the how to form the student teams.

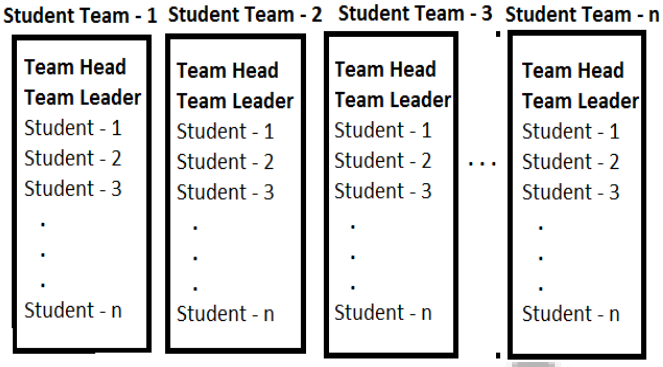

Figure 6: Construction Team Members using student Community

But we have one doubt for utilize student as developers, because many of the students having only theory knowledge, they don't have applicable knowledge. This is not a very big issue, we can solve it in the very easy way, before we use the students in our software-development process, and we give a one-month practical training to them. Some people think this one month is waste of time and the training get waste some more money it's absolutely not, because when we train the students in that time period, we can finish the other processes such as communication, planning, modeling and review of design. So we didn't waste our time because of the guide the students, and our next issue in this matter wasted the money for training, this is also enormously not because the student doesn't need any scale (remuneration) so the company has a lot of play down in money. Every student team has one project head and one project leader these two guys can manage the student developers and their activity, and furthermore these two are directly communicated with the project manager and send the everyday activities to the project manager. The project manager verifies that and sends the feedback about their team activity. This is a cycling activity up to the team will complete their module without bugs. If all the team complete their code implementation, then the implemented code will be tested, after the test process is completed, the project manager collects all the code and builds it before deploy the coding.

4.5 Deployment

When the code successfully builds it will be deployed. In software community, many of the peoples didn't know about deployment, and some other peoples think deployment is only the process of release the software. Deployment means to convert our developed software into an executable form. The customer or end users only working with our designs and get their results, but they don't see the source code. Why we need to deploy our software?

- After developing our software, our developed software has been full of designs and respective source codes.
- We cannot release this software with our developed source codes. Because customers or other persons can be a misuse, our source code and also can modify our source code.
- Which company developed the software they only have rights to modify that source code, and they can add patch files.

For these above reasons, we should deploy our software after developed.

5. Mathematical Equations describes cost variation

This is a very important place in this paper. We already see, our new skill process "Review of Design" will help us to minimize our design mistakes, if our design mistakes are fully cleared we can develop bug free software's, and also we include someone new skill model called "student team" here we utilize student as developers who will minimize our time and cost also. From the following equations, we can understand how our new skill processes model will minimize the development cost.

$$\Sigma T = \Sigma c + \Sigma p + \Sigma m + \Sigma i + \Sigma d$$

Here,

Σc -> Sum of Communication team expenses

Σp -> Sum of Planning team expenses

Σm -> Sum of Modeling team expenses

Σi -> Sum of Construction or Implementation team expenses

Σd -> Sum of Deployment team expenses

ΣT -> Total development Cost

Now, we are going to describe how our new skill process model will be minimizing the development cost rather than other two models. The following expenses are not absolute values that are our assumption only.

When using Waterfall model,

$$\Sigma T = \Sigma c + \Sigma p + \Sigma m + \Sigma i + \Sigma d$$

$$\Sigma T = \Sigma (8\%) + \Sigma (12\%) + \Sigma (16\%) + \Sigma (38\%) + \Sigma (17\%)$$

$$\mathbf{\Sigma T = 91\%}$$

The above equation describes expenses of total development cost, when using waterfall model. The Communication team has 8 % of expenses from the total development cost. Respectively, planning team got 12%, Modeling team got 16%, the development team got 38%, and the deployment team got 17%. When using a waterfall model the total development cost is 91%.

We all know waterfall model has a "blocking states" so every process level has one team that team will handle that project until their process level has been finished, so the development cost is low when using a waterfall model, but it has been so long to finish. We assume that waterfall model will take 360 days to finish our work with above cost.

When using RAD model, unlike a waterfall model, the RAD model has components of team members for modeling and constructing the code, the RAD model will take 90 days to finish the same work what we were done in a waterfall model, so we form four teams to modeling and constructing the project. Those four teams have named as m1, m2, m3 and m4 when modeling and i1, i2, i3 and i4 when constructing. If the company form four teams those teams

don't work in same time. So two teams working in the day-shift and two teams working in night-shift. The night shift staffs need more payment than day shift staffs. And also the company's other expenses also high like a office rent, electricity charge and so on...the following equation describes the total development cost when using a RAD process model.

$$\Sigma T = \Sigma c + \Sigma p + \Sigma m + \Sigma i + \Sigma d$$

We add modeling and construction teams

$$\Sigma T = \Sigma c + \Sigma p + (\Sigma m1 + \Sigma m2 + \Sigma m3 + \Sigma m4) + (\Sigma i1 + \Sigma i2 + \Sigma i3 + \Sigma i4) + \Sigma t + \Sigma d$$

We write it as,

$$\Sigma T = \Sigma c + \Sigma p + \Sigma(m1 + m2 + m3 + m4) + \Sigma(i1 + i2 + i3 + i4) + \Sigma t + \Sigma d + \Sigma o$$

Σt -> Sum of testing cost
Σo -> Sum of other expenses

Compare with the waterfall model, the RAD model will get more bugs, to minimize these errors. We should do many tests that are why here we add one more expense called ΣT now we will do some calculations for finding the salary of modeling and constructing teams. In the waterfall model, a single modeling team has got 16% of cost for 360 days.

1 day expenses of modeling team = 16 / 360
1 day expenses of modeling team = 0.044%
For 90 days expenses of 1 modeling team = 0.044 * 90
For 90 days expenses of 1 modeling team = 3.96%
The two day-shift teams cost (i.e. m1&m2) = 3.96 * 2
The two day-shift teams cost (i.e. m1&m2) = 7.92%

But the night shift teams need more cost than day shift teams, they need 1/3rd of amount higher than day shift teams, so we finding 1/3rd amount of day shift team.

The 1/3rd cost is = 3.96 / 3
The 1/3rd cost is = 1.32%

We can add this cost with night shift staffs salary

The two night-shift teams cost (i.e. m3&m4) = (3.96+1.32) * 2
The two night-shift teams cost (i.e. m3&m4) = 10.56%

Like the above calculations we can find the construction teams expenses cost. In waterfall model a single construction team has got 38% of cost for 360 days.

1 day expenses of construction team = 38 / 360
1 day expenses of construction team = 0.105%
For 90 days expenses of 1 construction team = 0.105 * 90
For 90 days expenses of 1 construction team = 9.45%
The two day-shift teams cost (i.e. i1&i2) = 9.45 * 2
The two day-shift teams cost (i.e. i1&i2) = 18.9%

But the night shift teams need more cost than day-shift teams, they need 1/3rd of an amount higher than day shift teams, so we are finding 1/3rd cost of the day-shift team.

The 1/3rd cost is = 9.45 / 3
The 1/3rd cost is = 3.15%

We can add this cost with night-shift staff's salary.

The two night-shift teams cost (i.e. i3&i4) = (9.45+3.15) * 2
The two night-shift teams cost (i.e. i3&i4) = 25.2%

Now we substitute these two calculations in our equation.

$\Sigma T = \Sigma(8\%) + \Sigma(12\%) + \Sigma(7.92\% + 10.56\%) + \Sigma(18.9\% + 25.2\%) + \Sigma(8\%) + \Sigma(17\%) + \Sigma(6\%)$
ΣT = 113.5%

Compare with the waterfall model, the RAD model has 22.5% in excess of cost, but we have done our development in short time. From these equations, these two models have someone advantages and disadvantages also.

When using the new skill process model, The communication cost, Planning cost, modeling cost all these factors are same like a waterfall model, here we add one more process level called "review of design" it has some expenses called Σr and we add one more expense for the student training programs called Σtr Now we should be finding the sum of student team expenses.

We know 90-day expenses of two day-shift construction teams as well as two night-shift construction teams. Respectively, 18.9% and 25.2%. Consider that this cost for ten developers. But in each student team, we have to spend money for two persons only this is shown in fig.6. So we divide the above cost by 10.

The two day-shift teams cost (i.e. c1&c2) = 18.9/10
The two day-shift teams cost (i.e. c1&c2) = 1.89 * 2
The two day-shift teams cost (i.e. c1&c2) = 3.78

The two night-shift teams cost (i.e. c3&c4) = 25.2/10
The two night-shift teams cost (i.e. c3&c4) = 2.52 * 2
The two night-shift teams cost (i.e. c3&c4) = 5.04

The training period cost is approximately Σtr = 5 %

The review of design team cost approximately Σr = 6%

We substitute these values in our equation,

$$\Sigma T = \Sigma c + \Sigma p + \Sigma m + \Sigma r + \Sigma tr + \Sigma(i1 + i2 + i3 + i4) + \Sigma d$$

$$\Sigma T = \Sigma(8\%) + \Sigma(12\%) + \Sigma(16\%) + \Sigma(6\%) + \Sigma(5\%) + \Sigma(3.78 + 5.04) + \Sigma(17\%)$$

ΣT = 72.82%

This is our total development cost when we are using our new skill process model. Compare with other two process models this cost is very low to develop our software.

5.1 Cost Variation Table

	Waterfall Model	*RAD Model*	*New skill process Model*
Total Cost **ΣT**	91%	113.5%	**72.82%**

6. Conclusions

In this paper, we have proposed a new skill process model for minimize the software-development issues. This new skill process model uses "review of design" for analyze design mistakes that will minimize bugs, more construction teams that will minimize time delay and utilize students as developers this will minimize the development cost.

7. Future Scope

This paper will help to the software community for developing a low cost and a worth software with bug free.

8. Acknowledgments

I would like to thank to my colleagues and my dear students.

References

[1] Roger S Pressman, “Software Engineering”, TATA McGraw-Hill Edition, Sixth Edition.
[2] Chetankumar Patel, and Muthu Ramachandran,"A Software Process Improvement framework for Agile Software Development Practices ", IJSE, Volume 2, Issue 1, January 2009, Pages 3-28.
[3] Walt Scacchi, "Process Models in Software Engineering**",** Institute for Software Research, University of California, Irvine, February 2001.
[4] Gerardo Canfora, Darren Dalcher, and David Raffo, "Software: evolution and process A new journal is born", dblp Journal of software, Vol. 24, Issue 1, page 1, 24 JAN 2012.
[5] Huzefa H. Kagdi, Malcom Gethers, Denys Poshyvanyk, and Maen Hammad, "Assigning change requests to software developers", dblp Journal of software, Vol. 24, Issue 1, Page 3,4 FEB 2011
[6] N. Fenton and Martin Neil, “A Critique of Software Defect Prediction Models,” IEEE Transactions on Software Engineering, September/October 1999, pp. 675-689.
[7] P.H. Feiler and W.S. Humphrey. (1993). Software Process Development and Enactment: Concepts and Definitions, Proc. 2nd Int. Conf. on "Software Process"
[8] Colette Rolland (1994). A Multi-Model View of Process Modelling. Requirements Engineering. Vol 4, Nr 4. Springer-Verlag.
[9] M. Dowson (1998). Iteration in the Software Process, Proc 9th Int. Conf. on Software Engineering.
[10] J. Mendling, H. A. Reijers, W. M. P. van der Aalst, Seven process modeling guidelines (7pmg), QUT ePrints Report 12340, Queensland University of Technology (2008)

Author Profile

Mohamed Azharudheen received the B.Sc., (Computer Science) and M.C.A degrees from Thanthai Hans Roever College in 2007 and 2010, respectively. During 2010 - 2011, did M.Phil in PRIST University. From july 2012 onwards working as a Assistant Professor, Department of Computer Applications in Thanthai Hans Roever College, Perambalur. He likes to have a research in software engineering.