

MANAGING UNCERTAINTY OF TIME IN AGILE ENVIRONMENT

Rashmi Popli¹ and Priyanka Malhotra² and Naresh Chauhan³

¹Assistant Professor, Department of Computer Engineering,
YMCAUST, Faridabad
rashmimukhiya@gmail.com

²M.Tech, Scholar, Department of Computer Engineering,
YMCAUST, Faridabad
jayant.malhotra1@gmail.com

³Professor, Department of Computer Engineering, YMCAUST, Faridabad
nareshchauhan19@gmail.com

ABSTRACT

Agile software development represents a major departure from traditional methods of software engineering. It had huge impact on how software is developed worldwide. Agile software development solutions are targeted at enhancing work at project level. But it may encounter some uncertainties in its working. One of the key measures of the resilience of a project is its ability to reach completion, on time and on budget, regardless of the turbulent and uncertain environment it may operate within. Uncertainty of time is the problem which can lead to other uncertainties too. In uncertainty of time the main issue is that the how much delay will be caused by the uncertain environment and if the project manager comes to know about this delay before, then he can ask for that extra time from customer. So this paper tries to know about that extra time and calculate it.

KEYWORDS

Slack, Optimistic time, Pessimistic time, Probability of Delay

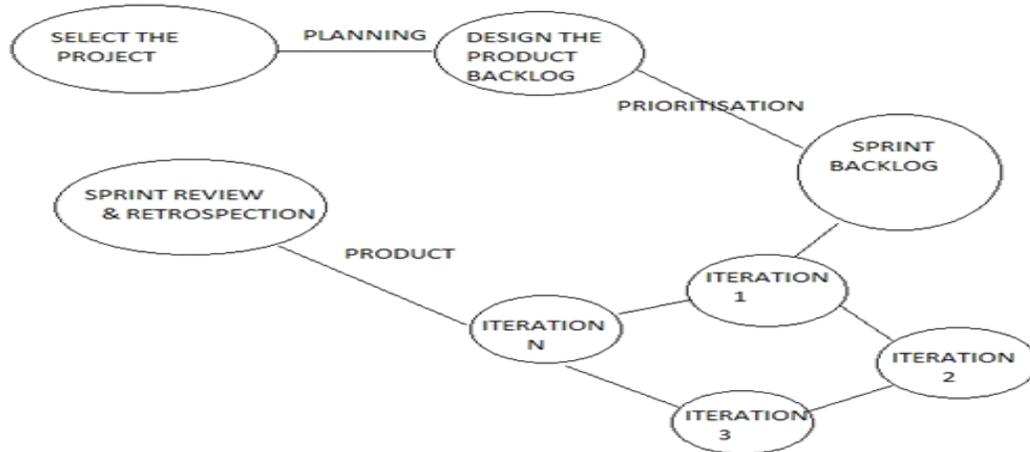
1. INTRODUCTION

Agile software development methodologies become increasingly popular as the word spreads about the benefits they provide under certain project conditions. A key characteristic of any agile approach is its explicit focus on time estimation and business value for the clients. The goal of time estimation is typically to develop potentially shippable product. The accurate estimations of time is critical for both developer and customer. Ignorance of estimation methods may cause serious effects like exceeding the budget, poor quality and not right product. The key factor which is causing the problem in estimation is time uncertainty, so there is a need of some mechanism for minimizing uncertainty of time.

In Section II the life cycle of agile is described. Section III describes what uncertainty is. In Section IV related work in this field is discussed, section V proposes scenario calculation of slack time and uncertainty in time in agile. In Section VI shows evaluation and results of proposed algorithm, section VII concludes the paper.

2. AGILE LIFE CYCLE

Meaning of Agile is “moving quickly”. When applied to software development, it means that delivering the software that meets the customer requirements in shortest possible time. The success and failure of a software project is determined by accurate estimation. This is the process to calculate the time that will be taken to finish the project, cost of the project and the effort required to complete the project.



Estimation is very important task as improper estimation may lead to failure of the software project. It may also increase the budget of the customer and sometimes the nature of the project is also affected. The estimation in the Agile environment is a difficult task due to the changing requirements. The figure 1 shows the Agile software development life cycle. Agile software lifecycle is an iterative process where software is ready at each iteration but can always be improved in next iteration. Or in agile terms a part of the project is ready at each iteration and that part itself can be improved at each iteration or can be free from bugs at each iteration.

3. UNCERTAINTY

Meaning of Agile is “moving. In releasing a particular plan or **user story**, it is needed to fix a set of release dates and then determine how much functionality can be achieved by those dates. Also this can be done by deciding the functionality first and then deriving the release date. In either case the functionality value is assessed against the cost and time to develop the system, In previous used methods cost and time both get neglected in assessing the functionality which lead to uncertainty in both time and cost. Also the size of the user story is not certain. These all factors leads to poor estimation in agile project and hence time uncertainty. In this paper there is an attempt to find solution to problem of these uncertainties and calculating the percentage of uncertainty.

4. RELATED WORK

McDaid, D. Greer, F. Keenan, P.Prior, P. Taylor, G. Coleman[8] proposed a set of key practices which includes the practice, termed “Slack”, of only signing up to for what the team is confident of achieving. Within this approach it is always possible to add more stories, time permitting, thus delivering more than was actually promised. This practice acknowledges that there is a significant amount of uncertainty in the estimated time to complete releases.

Rashmi Popli, Anita and Dr. Naresh Chauhan[5] proposed a common life cycle approach that is applicable for different kinds of teams. This approach describes a mapping function for mapping of traditional methods to agile method.

Siobhan Keaveney and Kieran Conboy[1], gave the study of the applicability of current estimation techniques to more agile development approaches by focusing on four case studies of agile method use across different organizations. The study revealed that estimation inaccuracy was a less frequent occurrence for these companies. The main estimation techniques used were expert knowledge and analogy to past projects also the Component of the process; fixed price budgets can prove beneficial for both developers and customers, and experience and past project data should be documented and used to aid the estimation of subsequent projects.

S.Bhalerao and Maya Ingle[4] presented the study of both traditional and agile estimation methods with equivalence of terms and differences. This study investigated some vital factors affecting the estimation of an agile project with scaling factor of low, medium and high. Also, an algorithm Constructive Agile Estimation Algorithm (CAEA) is proposed for incorporating vital factors.

Daniel D. Galorath[3] proposed a 10-step estimation process that begins by addressing the need for project metrics and the fundamental software estimation concepts. It shows how to build a viable project estimate, which includes the work involved in the actual generation of an estimate, including sizing the software, generating the actual software project estimate, and performing risk/uncertainty analysis.

5. PROPOSED WORK

Client gives customer the requirements in form of user stories, and a backlog is created with those requirements. The time required for completion of project depends upon size of the user story. It needs to be certain about time taken by the project to be completed. The uncertainties in the timing is a big problem so project takes some marginal time called as slack that will compensate for extra time taken in the project work other than the optimal development time. However there is no formula for calculating this Slack. The proposed algorithm and formula suggests that how to calculate the slack time. For calculations, the time taken as hours rather than days because that will lead to actual result.

Effective time per day for Sprint Related work = Average work day time - Time allocated for other activities.

For example Average work day time = 10 hours

Time Allocated for other activities = 5 hours

Emails and Phone: 1 Hrs

Lunch: 1 Hrs

Meetings: 2 Hrs

Bug fixes: 1 hrs

Available time for Sprint Related work = 5 hours

5.1 Proposed Diagram

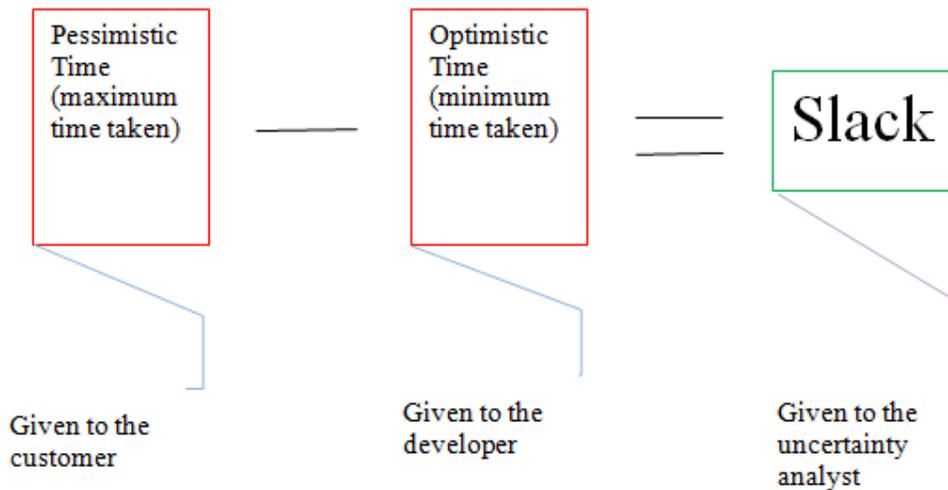


Figure 2: Showing what actually happening in this scenario of calculating slack

In the project there is a team of about 7 persons, one is project manager, four developers and two testers who are doing pair programming. For four developers total development time will be the ideal available time for development related work * 4.

The optimistic value of time period for one iteration is given to the developers so that they have to complete task in that period, and the pessimistic value is given to the customer or client. The calculation for pessimistic timing will be done by using probability of delay.

5.2 Proposed Formulae

The difference between these two timings will be helpful for exactly calculating the slack time.

1. **Duration for developer** = $\sum_{i=1}^n T_i$ [optimistic time] / work per time
2. **Duration for customer** = $\sum_{i=1}^n T_i$ [pessimistic time] / work per time
3. **pessimistic time for each task** = (percent probability of delay * optimistic time) / 100 + optimistic time
4. **Slack time for a task** = duration for developer - duration for customer
5. **Time Uncertainty** = (slack / duration for developer) * 100
6. **Total slack time** = (Total pessimistic time - Total optimistic time) / (Total working hours per day * Number of developers)

5.3 Proposed Algorithm

The proposed algorithm explains the various steps involved in removing the uncertainty in time in agile environment.

1. Identify the tasks of each user story based on the requirements, suppose for each user story there are n number of tasks so for each user story the tasks are $t_1, t_2, t_3, \dots, t_n$.
2. Identify the optimistic time for each task and probability of delay for each task. Metric for Optimistic time value is hours and probability of delay will be in percentage.

3. Calculate the time (pes) for each user story by using the formula $\sum_{i=1}^n T_i$ [pessimistic time] where “i” denotes the tasks and pessimistic time for each task = (percent probability of delay * optimistic time)/100+optimistic time
4. Compute the overall Slack time for all the user stories using the formula Slack time = duration for developer - duration for customer
 Duration for developer = $\sum_{i=1}^n T_i$ [optimistic time] / Effective working hrs
 Duration for customer = $\sum_{i=1}^n T_i$ [pessimistic time] / Effective working hrs
5. Calculate the uncertainty percentage which is
 = (slack time/duration for developer)*100

5.4 Proposed Activity Diagram

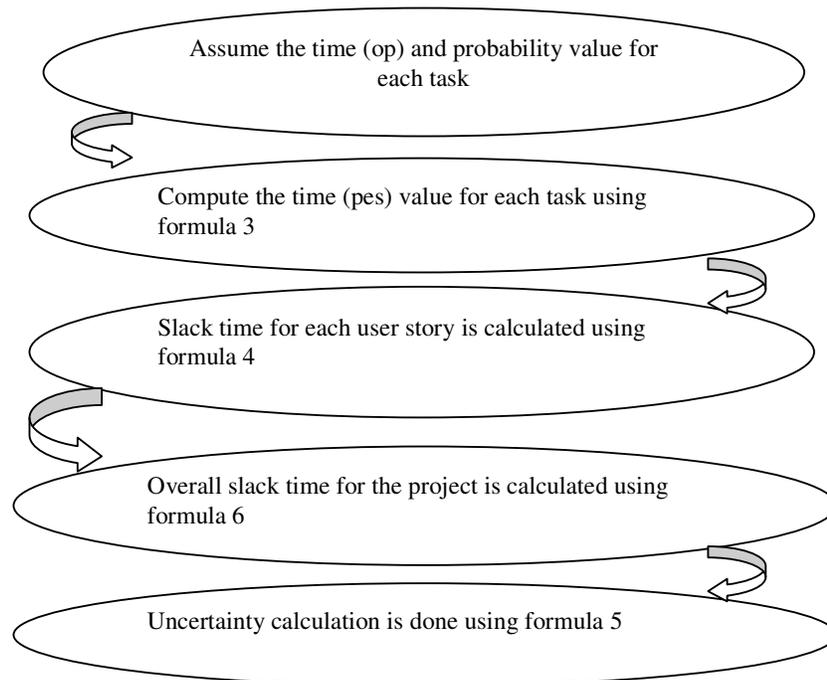


Figure 3: Steps involved in calculation of slack and uncertainty

6. EVALUATION AND RESULTS

In this section the feasibility of our algorithm is shown by calculating the estimated values of pessimistic time using probability of delay, value of slack and percentage of uncertainty for a project. We had considered the number of effective working hours as 5 per day. In this section the feasibility of our algorithm is shown by a case study in which the values are being calculated. We have considered the user stories of project which is **Letters of Credits**; the client is HP-client. A graph can be drawn taking the optimistic time and pessimistic time values against each other, the bar graph and line graph both show the difference between both the values.

Table 1: Showing tasks of the project under taken and related values of optimistic time and probability of delay

N O.	DEVELOPMENT TASKS	OPTIMISTIC TIME	PROBABILITY OF DELAY
1	FSD REVIEW EXPORT OPENING	20	30
2	FSD REVIEW EXPORT REVIEW	10	20
3	FSD REVIEW OPENING CREATION	26	20
4	FSD IMPORT REVIEW CREATION	42	10
5	FSD EXPORT REVIEW CREATION	33	10
6	PH-2 REQUIREMENT STUDY	20	30
7	FSD EXPORT REVIEW CREATION	30	10
8	FSD IMPORT OPENING SIGN-OFF	45	20
9	FSD IMPORT REVIEW SIGN-OFF	50	10
10	FSD EXPORT OPENING SIGN-OFF	54	30
11	FSD EXPORT REVIEW SIGN-OFF	32	10
12	DEVELOPMENT REVIEW 1	34	20
13	DEVELOPMENT REVIEW 2	65	10
14	BACKUP ARCHIVE	30	20
15	PROJECT MONITORING	40	30
16	CONFIGURATION	23	20
17	UNIT TESTING	25	10

18	INTEGRATION TESTING	35	10
19	SYSTEM TESTING	25	20
20	TRAINING	20	20
21	PH1 UAT	5	30
22	PH1 UAT SIGN OFF	15	20
23	PH2 UAT SIGN OFF	20	10
24	PH2 DEVELOPMENT REVIEW 1	20	20
25	PH2 DEVELOPMENT REVIEW 2	19	20
26	PH2 UAT	7	10

Table 2: showing user stories along with there corresponding associated tasks

NO.	USER STORY	ASSOCIATED TASKS
1	SRS	1-5
2	SRS REVIEW	6
3	DOCUMENTATION	6,7
4	FSD REVIEW IMPORT OPENING	8-10
5	NON-FUNCTIONAL DATA COLLECTION	20-23
6	PRE ENGAGEMENT SUPPORT	13-16
7	REWORK CODING	11,12
8	TESTING	17-19
9	CODE REVIEW	24,25
10	GO LIVE SUPPORT	21,26

Table 3: user stories with their optimistic time and calculated pessimistic time

USER STORY NO.	TOTAL OPTIMISTIC TIME	TOTAL PESSIMISTIC TIME
1	131	152
2	20	26
3	50	59
4	149	179.2
5	60	70.5
6	158	187.1
7	66	76
8	85	96
9	39	46.8
10	12	14.5

6.1 Numerical analysis

Total slack time= (Total pessimistic time-Total optimistic time)/Total working hours per day*Number of developers

Here total working hours per day are=5 hours

Number of developers=4

Total Slack time = $(907.1-770)/20=6.85$

Now the Percent uncertainty in the project= $(\text{Slack time}/\text{Pessimistic time}) * 100 = (6.85/45.35) * 100 = 15.10\%$

6.2 Graphs

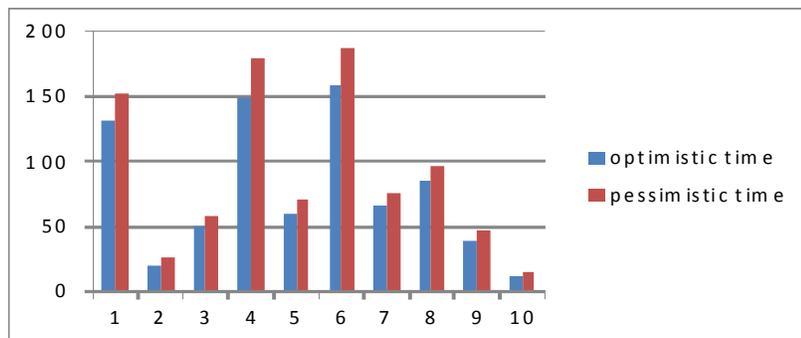


Figure 4: Bar graph representation of optimistic and pessimistic timing values

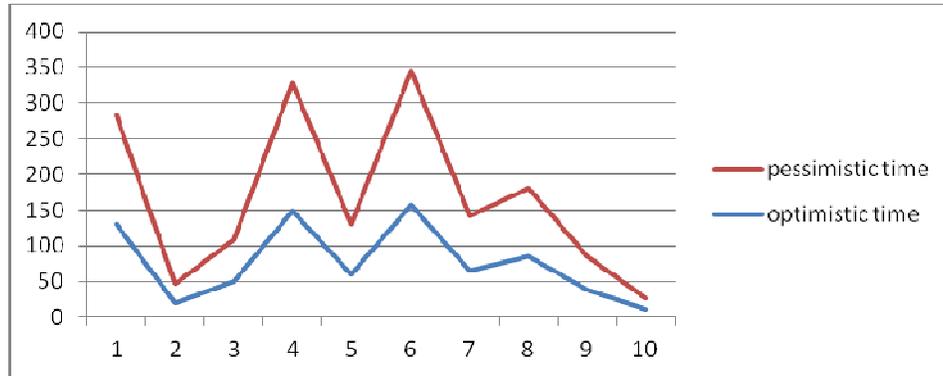


Figure 5: Difference between optimistic time and pessimistic time shown by lines

7. CONCLUSION

This uncertainty percentage tells us that by this percentage there are the chances of the project to get faulty. If this percentage value is much higher, then this may lead the developers think about removing the uncertainties first and then start the project. The slack value helps in improving the project as we can remove some causes which are leading to the delay. Also the value of slack is important because now the project manager is sure about the completion time of the project and the relations with the customer can be improved as customer is well satisfied with the project completion at the given dead line. The approach used in the paper is very easy to understand and do not need any hard and fast calculations. After having exact values it would be easier for the manager to read out the reports because the work is shifted from the theoretical portion to the exact numerical data. Hence this paper is an approach to solve the problem of uncertainty of time.

REFERENCES

- [1] Siobhan Keaveney and Kieran Conboy , “Cost Estimation and agile development projects” Product-Focused Software Process Improvement, 435-440
- [2] Du, G., McElroy, J., &Ruhe, G. (2006). Ad hoc versus systematic planning of software releases—a three-staged experiment. Product-Focused Software Process Improvement, 435-440.
- [3] Daniel D. Galorath, “The 10 Step Software Estimation Process For Successful Software Planning, Measurement and Control” galorth incorporated 2006.
- [4] S. Bhalerao and Maya Ingle, “Incorporating Vital Factors In Agile Estimation Through Alogorithmic Method” International Journal of Computer Science and Applications, Ó2009 Technomathematics Research Foundation ,Vol. 6, No. 1, pp. 85 – 97
- [5] Rashmi Popli, Anita and Naresh Chauhan. “mapping of traditional software development methods to agile methodology”
- [6] Logue, K., &McDaid, K. (2008). Agile Release Planning: Dealing with Uncertainty in Development Time and Business Value. Engineering of Computer Based Systems, 2008. ECBS 2008. 15th Annual IEEE International Conference and Workshop on the (pp. 437-442). IEEE.
- [7] McDaid, K., Greer, D., Keenan, F., Prior, P., Taylor, P., & Coleman, G. (2006). Managing Uncertainty in Agile Release Planning.Proc. 18th Int. Conference on Software Engineering and Knowledge Engineering (SEKE’06) (pp. 138-143).
- [8] Logue, K., &McDaid, K. (2008). Agile Release Planning: Dealing with Uncertainty in Development Time and Business Value. Engineering of Computer Based Systems, 2008. ECBS 2008. 15th Annual IEEE International Conference and Workshop on the (pp. 437-442). IEEE.
- [9] RashmiPopli, NareshChauhan,” Research Challenges of Agile Estimation” Journal of Intelligent Computing and Applications” July- Dec 2012.

- [10] RashmiPopli, NareshChauhan,” “Scrum- An Agile Framework”, International Journal of Information Technology and Knowledge Management (IJITKM) ISSN: 0973-4414”, Vol-IV, Number-I, 20 Aug 2010.

AUTHORS

Rashmi Popli is pursuing her Ph.D in Computer Engineering from YMCA University of Science & Technology, M.Tech(CE) from M.D University in year 2008,,B.Tech(IT) from M.D University in the year 2004.She has 9 years of experience in teaching. Presently she is working as an Assistant Professor in department of Computer Engineering in YMCA University of Science &Technology, Faridabad, Haryana, India. Her research areas include Software Engineering, Software Testing and Software Quality.



Priyanka Malhotra is a research scholar pursuing her M.tech in computer engineering (Compter Networks) from YMCA University of Science & Technology, Faridabad, Haryana, India. Completed B.tech (Cse) from Kurukshetra Unviersity, Kurukshetra



Dr. Naresh Chauhan received his Ph.D in Computer Engineering in 2008 from M.D University, M.Tech(IT) form GGSIT,Delhi in year 2004,B.Tech(CE) from NIT Kurukshetra in 1992.He has 22 years of experience in teaching as well as in industries like Bharat Electronics and Motorola India Pvt. Ltd. Presently he is working as a Chairman and Professor in the department of Computer Engineering ,YMCA University of Science and Technology, Faridabad, Haryana, India.. His research areas include Internet Technologies, Software Engineering, Software Testing and Real Time Systems.

