



Agile RACI Model for Extreme Programming Method

Mohamed A.I. AlFaki¹, Omnya Ali¹, Abdelelah Elgasim Babiker², Ashraf Osman Ibrahim³

¹Department of Software Engineering, Al Neelain University, Khartoum, Sudan

²Department of Accounting Information Systems, Al Neelain University, Khartoum, Sudan

³Department of Information Technology, Alzaim Alazhary University, Khartoum, Sudan

Abstract— Agile software development approaches have a wide spread reputation among software development methods. Agile methods core is a people. Thus, sharing responsibilities is an important and critical on software projects in order to accomplish task. However, there is confusion about who is responsible for and accountable to among the development team. In this paper, a new Agile RACI model which is based on RACI Matrix roles has been developed, to involve the development team and to find a collaborative way to make a clear accountability and responsibility for the activities. The proposed model consists of set of notations instead of RACI matrix to visualize the roles and responsibilities. The results show that the Agile RACI model is a great tool, in making a clear accountability and responsibility in the agile development methods.

Keywords— Agile Methods, Agile RACI Model, RACI Matrix.

I. INTRODUCTION

Recently in software engineering, there are some new terms used to describe software development methodologies. Agile software development (ASD) is one of them; it is an incremental and iterative development methodology and it has a famous reputation among the software development methods. Agile methodologies really represent a new approach for planning and managing software projects. Agile Project Management offers a method can characterize by its agility. It also enables organizations to increase the benefits of an agile methodology without introducing unnecessary risks [1]. Many studies in the literature indicate the significant of the people in agile methods as a crucial factor for success or failure in agile projects [1, 2, 3]. S.Nerur, V. Balijepally (2007) in their study [4], they compared the applicability and efficiency of agile methods for projects with small number of team to large number team. People are the main factor for success or failure in agile methodologies. In agile methodologies team members and management have the same level. However, technical staff do not take the role of managers [5,6,7]. By this way, it improves collaboration and limits constraints. People are the main factor for success or failure in agile methodologies. In agile methodologies team members and management have the same level. However, technical staff do not take the role of managers [5,6,7]. By this way, it improves collaboration and limits constraints. According to [8] there are many problems in defining and categorizing people contribution in working in small teams and that leads to difficulties in classifying individuals' contributions and how they can be pleased [8]. Since its first presented before more than 60 years, RACI became widely used among projects managers as software community due to its simplicity, adaptability and agility in responding to change. RACI also used as an effective communication means between project teams [9].

II. BACKGROUND AND RELATED STUDIES

Agile software development in general is characterized by the following attributes: incremental, cooperative, straightforward, and adaptive [10]. Incremental refers to small software releases, with rapid development cycles. Cooperative refers to a close customer and developer interaction. Straightforward implies that the method itself is easy to learn and to modify and that it is sufficiently documented. Finally, adaptive refers to the ability to make and react to last moment changes. Due to limited space, readers are referred to [10]. Scrum [11, 12], Crystal methods, feature driven development, and adaptive software development [16] are approaches have been developed for managing the software development process in a volatile environment. It is an empirical approach based on flexibility, adaptability and productivity.

Extreme programming (XP) [13, 14] is a set of famous software engineering practices. XP aims at enabling successful software development regardless of unclear or continually changing software requirements. The revolution of XP is founded on the technique of bringing all the development participants to work together. Some of the main features of XP are short iterations with small releases and rapid feedback, close customer participation, constant communication and coordination, continuous refactoring, continuous integration and testing, collective code ownership, and pair programming.

Extreme programming roles

Extreme programming has five roles: -

Programmer who is responsible for defining, estimating, implementing and testing requirements stories.

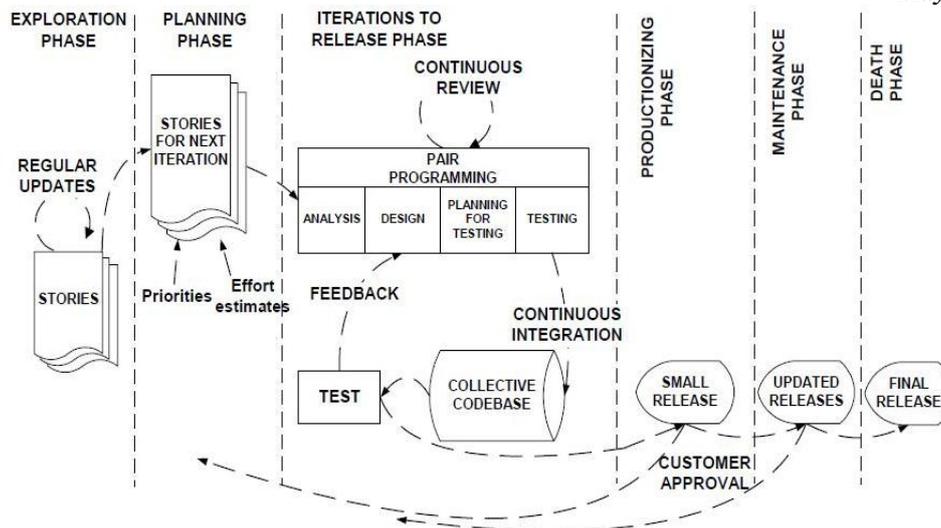


Figure 1: Extreme Programming Method

Customer main responsibility is to writing, prioritized requirements stories.

Coach is the person who is insuring.

Tracker tracks progress of the development team based on the planned work. Finally, Tester tests the release product [16].

Extreme Programming Phase

According to [16] there are six phases in XP: - exploration, iteration to release, productionizing, maintenance and death phase, each one consists of set of activities and involves some or all the development team.

Exploration phase it's the first phase which customers writing the story cards with the developer assistant, also all the technologies will be used in the developing project will be tested, this phase takes up to few months depending the project and team.

The second phase is **planning phase**, story priority is determining and schedule and cost is estimated, it takes about two days.

Iteration to release from the planning phase the project is divided into iterations with a predefined time from one week up to month for each one. Customer choose story for iteration and then the development team implement it.

Productionizing phase. At this phase testing is conducting for the release and thus, May new requirements are demanded. In this phase time of iteration is reduced to only one week.

Maintenance phase needs an additional work also for customer support activities. The maintenance phase may require adding new individuals into the team and rearrange team.

Death phase is the last phase in XP, there are no new changes or requirements and customer is satisfied and all documentation is written [16].

III. RACI MATRIX

RACI Matrix is a technique for identifying activities and roles in the projects. RACI was originally called by a more formal, academic name, the "Decision Rights Matrix" and is also known as "Responsibility Charting" [9]. RACI Matrix enables managers to participate in making a decision about assigning roles and responsibilities among the team. Discussion about linked process and how it depends on each other and how to utilize the team to accomplish their tasks. In order to deliver a successful product or service. RACI matrix maintains accountability and ensures it is placed with the person who really can be accountable for specific work .Everyone has some process role in their job.

Responsible is project team member(s) who really responsible for doing and completing the task.

Accountable is project team member who is finally accountable for the activity or decision. There is must be only one accountable per activity or action.

Consultant is project team member(s) that is referred to before taking final decision or action. And it is two-way communication.

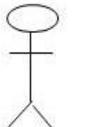
Informed is project team member(s) who wants to be knowledgeable after a decision or action is taken. It is a one-way communication [15].

IV. METHODOLOGY

Shared responsibilities are important on a software project in order to accomplish a task; however there should be a clear accountability among the development team. Someone must ultimately be answerable for the correct and thorough completion of the deliverable. We propose a new model called Agile RACI Model which is based on the same RACI roles but not its matrix. Agile RACI Model is a model to identify the key functional process area in a certain agile development phase and classify the agile team according to their roles and responsibilities in each phase. Agile RACI Model consists of set of notations similar to UML notations which makes the development team familiar with it. The

proposed model will organize the work in the agile environment and explain clearly each role and its responsibilities in each phase of the development model .also this mode will increase the accountability among the team as it could use as a learning training tools among the development team and organization. Our proposed model addresses this issue by dividing the RACI roles between the agile development team according to their role in each phase. RACI model consist of the following notations as described in the table below.

Table 1: Agile RACI Model

Notation	Description
	This is the agile team member
	This is artifact which is the output of process, actions, and events. Could a document, plan,...etc.
	This is a process or action that is done in specific agile method phase.
	This arrow comes out from the team member process or artifact to describe accountable role .
	This arrow to describe responsible role.
	This arrow comes in to the team member from the process or artifact to describe to describe informed role.
	This arrow to describe consultant role.

Example: - Extreme Programming (XP).

We used our proposed model in XP to show its possibilities and how Agile RACI Model can be used in the other remaining agile Methods. XP consists of six phases:-exploration, planning, iteration, productionizing, maintenance and death phase. But in this example we implement the proposed model in the main three phases: - exploration, planning, and iteration which they contain the major activities in XP.

Exploration phase. It all about writing story cards by the customer and with developer assisting.

- Customer and programmer are responsible for writing story cards.
- Customer is accountable to story cards and has ‘Yes’ ‘No’ authority. See figure(3) below

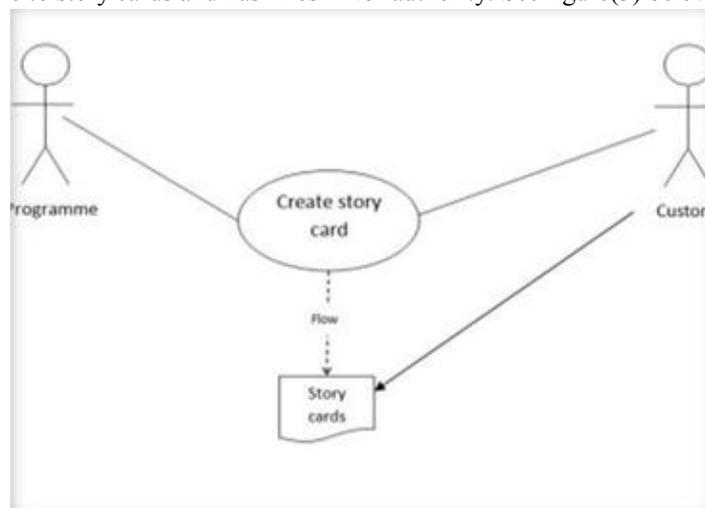


Figure. 2 Exploration Phase

Planning phase in this phase the priority and estimation of time and effort required for each stories is made.

- Customer is responsible and accountable to set priorities of the stories.
- Programmer is a consultant of prioritize process.
- Programmer is also responsible and accountable to estimate the stories. See figure(4)

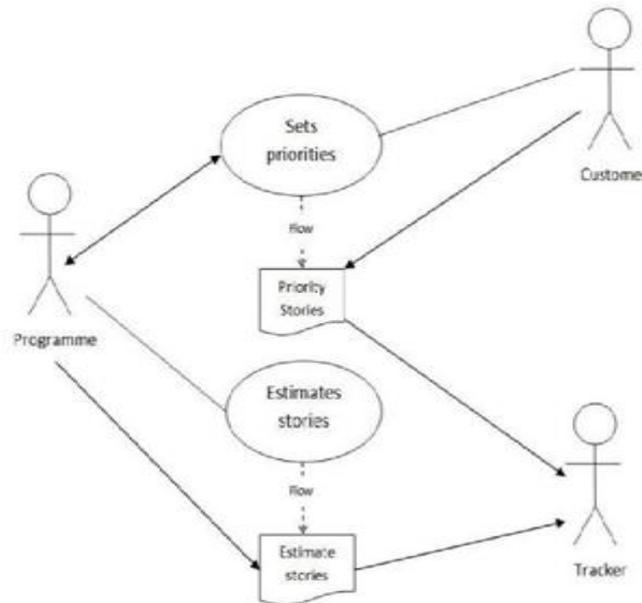


Figure. 3 Planning Phase

Iterations to release this phase include the following activities:-

- Customer is responsible and accountable from selecting stories for iteration.
- Tracker and programme should be informed with stories selected for iteration.
- Customer is responsible for writing function test.
- Tester is accountable to function test.
- Tracker should be informed with outcomes of this phase. See figure (5).

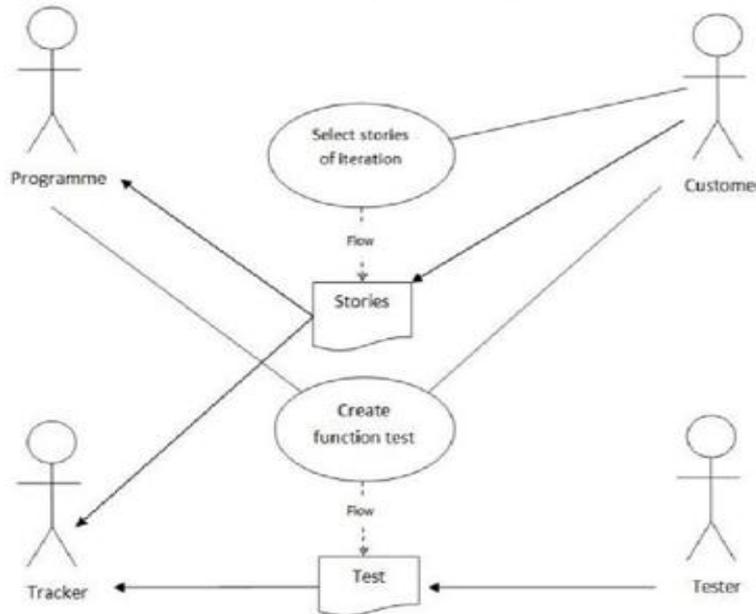


Figure. 4 Iteration Phase

V. RESULTS

The proposed solution model has been introduced based on two facts: firstly, agile methods are the most popular development method in the last two decades. Secondly, we need to clearly state what people are accountable to and responsible for in any agile development method. So we created this model based on RACI matrix which it proved it has a positive effect on software projects and seems a nice way to clearly state who is uniquely accountable and responsible for. Our proposed model helps team understand their roles and responsibilities. In addition, it solves the confusion about who is responsible for what, that makes roles and responsibilities very clear among the development team and for the project manager as well.

From the previous sections, it is clear from the figures 3-5 the three phases which were applied in our model using XP method that makes the roles and responsibilities very clear in each phase.

The three mentioned phases proved that it is possible to implement the proposed model in each agile development method. They can decrease the confusion among the development team with the regard of their roles and responsibilities. Moreover, the proposed model can be used as an agile training tool.

VI. CONCLUSION

This paper introduced a new model that can be used in agile methods as an assistant tool to provide more awareness and increase accountability and responsibility among the agile development team. The proposed model applied in XP method as an example, it consists of three phases which are exploration, planning and iteration phase. The results show that the Agile RACI model is a great tool can be used in other different agile methods. For the future work, we will implement and develop software tool that will use the proposed model in Scrum Agile method applied in Agile RACI model.

REFERENCES

- [1] Kuda Nageswara Rao, A Study of the Agile Software Development Methods, Applicability and Implications in Industry, International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011
- [2] M. A. Awad , A Comparison between Agile and Traditional Software Development Methodologies , This report is submitted as partial fulfillment of the requirements for the Honors Program of the School of Computer Science and software Engineering, The University of Western Australia, 2005.
- [3] W.H. Morkel Theunissen, Derrick G. Kourie and Bruce W. Watson, Standards and Agile Software Development, Proceedings of SAICSIT, Pages 178– 188, 2003.
- [4] S.Nerur, V. Balijepally, Theoretical reflections on agile development methodologies, Communications of the ACM 50 (3) (2007) 79–83.
- [5] Paul E. McMahon, Are Management Basics Affected When Using Agile Methods?, The Journal of Defense Software Engineering, November 2006.
- [6] B. Henderson-Sellers, M.K. Serour, Creating a Dual-Agility Method: The Value of Method Engineering, Journal of Database Management, vol. 16, Issue 4, 2005.
- [7] Alistair Cockburn, Agile Software Development: The People Factor, Software Management t, Page 131-133, November 2001.
- [8] Cervone, Frank, “Understanding agile project management methods using Scrum”, 2010.
- [9] P. M. Khan, Kaleem A. Quraishi , " Impact of RACI on Delivery & Outcome of Software Development Projects " , Fourth International Conference on Advanced Computing & Communication Technologies , 2014 .
- [10] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, *Agile software development methods: Review and Analysis*. Espoo, Finland: Technical Research Centre of Finland, VTT Publications 478, Available online: <http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf>, 2002.
- [11] K. Schwaber, "Scrum Development Process," presented at OOPSLA'95 Workshop on Business Object Design and Implementation, 1995.
- [12] K. Schwaber and M. Beedle, *Agile Software Development With Scrum*. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [13] K. Beck, "Embracing Change With Extreme Programming," *IEEE Computer*, vol. 32, pp. 70-77, 1999.
- [14] K. Beck, *Extreme Programming Explained: Embrace Change*, 2000.
- [15] Erwin, Michael L Smith and James, “Roles & Responsibilities Charting (RACI)”.
- [16] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen and Juhani Warsta, “Agile Software Development Methods, Review and Analysis”, 2002, University of Oulu.