International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 4, Issue 1, January 2015 ISSN 2319 - 4847

RACI Scrum Model For Controlling of Change User Requirement In Software Projects

Aymen S. Elhady[1] , Hisham M. Abushama[2]

Department of Computer Science Faculty of Mathematical Sciences
University of Khartoum ,Khartoum ,Sudan

ABSTRACT

Requirement's evolution is mandatory for any software project. Users can propose requirements changes at any stage of SDLC. Although changes in requirements may affect on cost, schedule and quality of a software project; but change should be allowed when it is inevitable to meet the customer expectations. Traditional software development processes are not much efficient to manage the rapid change in requirements. This paper explains why users change their requirement and what is the impact of change in a project's cost, schedule and quality and it attempts to decrease change in required by proposed model can do this task.

Keywords:- Change requirement, RACI, Scrum, Software Project.

1.INTROUDCTION

The requirement includes descriptions of system properties, specifications for how the system should work, and constraints placed upon the development process. Generally, requirement are statements of what a system should do rather than how it should do it [1]. Requirement comes from end users, from customers, and sometimes from developers; therefore It's important to have all these groups (end users, customers, developers) contribute to the requirement document to create a fuller description of the system. Although writing a complete requirement document is time-consuming [1], there are many advantages to having one. Involving major stakeholders in writing requirement helps to ensure that everyone agrees on what is to be done; therefore this can avert misunderstandings down the road and save time that later might be wasted in rework. As stated by McConnel, 1993 "Stable requirement are the holy grail of software development ", changing requirement have become an accepted fact of life for software developers, and can be changed at any stage in the Software Development Life Cycle (SDLC) from requirement phase all the way through several iterations into the maintenance phase. In fact, it is likely that more than half of the system's requirement will change before deployment [3]. Changes can have different risks associated with them in terms of cost and time and the impact increases with the propagation of those changes from one phase to the next. When a change occurs during the implementation of an existing requirement, its impact is not only limited to that particular phase where change was proposed, but also propagates to other subsequent phases of SDLC [2]. Because of this propagation effect cost in terms of development effort is directly affected [4]. Frequent changes in project requirement interrupt the project and contribute to a greater effort each time work is resumed. Although change represents risks, but there are times in a project life cycle, when change is encouraged. For instance, at the time of requirement definition, the need for business requirement to go through change as people brainstorm, collaborate and refine their needs. However, as the project crosses over into the execution phase, desire for business requirement to be stable, and any change to happen only after the right people have approved it. Even this change should be such that, stakeholders (upstream and downstream requirement owners) should be able to quickly locate and address it; therefore the software development community needs to rethink its approach to changing requirement. Should be stopping seeing change requirement as a problem and start to see it as an opportunity for improving the quality of product that achieve the customer's satisfaction and that represents one of the major goals for the software development community.

2.LITERATURE REVIEW

1. Causes of Change Requirement

A study by Edberg and Olfman (2001) looked at the motivations behind software change requests at a variety of organizations during the software maintenance phase. Corrective maintenance (i.e. Bug fixing) accounted for only 10-15% of workers while functional enhancements accounted for over 60% of changes. This 60% was broken down into four categories:

- **External changes -** changes required to meet some need from outside the
- Organization, for instance a changed legal requirement.
- **Internal changes -** changes required because of company changes such as new products, or restructuring.
- **Technical changes -** changes required to meet new technical demands.
- **Learning -** changes resulting from learning by individuals or groups.

Edberg and Olfman suggest that 40% of these changes were primarily the result of learning. By changing software, organizations can pass on the benefits of one group's learning to the whole company potentially saving money and/or time and improving efficiency. Interestingly, though, users who requested changes often didn't attribute their request to learn, they preferred to cite other internal or external factors as the motivation. Although the changing requirement was likely to increase the estimated cost and time, it definitely will improve the quality of the software. Most of the studies or researches are finding the ways to improve the requirement engineering process to acquire quality requirement . As stated by Allen Kelly [5], the requirement specification documents the users' needs that have been collected before the software project start. But, as the project proceeds, both the software developer and the user will have more clear understanding of the business activities and the software respectively. This may even give greater value to the system. As mentioned by Allen Kelly [5], instead of giving bad impact of the changing requirement, it should be considered as an opportunity. The varying requirement gives the developer a chance to improve the quality of the software. The good quality software compliance with the users needs with no defects and meets the intended purpose.
The changing requirement is vital for the business not to be avoided as it might have been forgotten or ambiguously mentioned during the requirement process. The requirement might also be changed due to the technology or environmental change. So the software development team should always welcome the requirement change as it will certainly increase the success of the project by satisfying the users. Now the question is does all changes will be accepted ? . Not all the changes will be accepted. Only the changes having an impact on the project will be accepted. This will lead to a successful project development. If the changes are accepted without a change control process, this will lead to project failure as it may direct the project in another direction.

2. Agile Methodologies

Agile methodologies started to become popular in recent years, mostly due to their ability to counter many weaknesses of other development methodologies. Agile methodologies try to change and improve classic ways used in software development. They aim to improve all areas of software development, including the most important constraints of cost, schedule, resources and quality. The main idea behind these methodologies is efficiency, flexibility, adaptation to change and satisfaction of customer's needs. This is achieved mostly by lightweight, creative processes and better communication. Agile project management is approaches based on a few principles such as "welcome change during project development ", Agile approaches are aimed to deliver value and increase customer's satisfaction; therefore to achieve these goals, managers whose implement Agile should focus on collaboration and continues learning because its style is based upon collaboration, flexibility, dynamism [5]. Due to the defects in traditional approaches such as its weakness in projects that have complex structure, uncertain, requirement is continuing changing and strict time bound [5]. Agile is burst as a methodology to be more suitable for these types of projects. Customer involvement in traditional software development projects is typically limited to provide the requirement in the beginning and feedback towards the end, with limited regular interactions between the customer and the development team [5]. In contrast, customer collaboration is a vital feature and an important success factor in Agile software development [6]. Although Agile involves customer in software development life cycle – customer involvement – is represents as merit in Agile methodology but this concept will become weak point if customer unavailable all the time with developer , moreover if key customer is one of the high level managers , also spend customer with developer side by side might lead to a weak documentation, moreover when customer not collaborate efficiency in software development life cycle that may lead to problems in gathering and clarifying requirement and that may lead to frequent changes in requirement during software development life cycle ; therefore the productivity will be losing as a resulting to that business will be losing [7] . Agile methodologies are looking for customer's satisfaction by adopted for many methods or practices under its umbrella and it addressed change requirement in its principles and it alleged it embrace change even it comes in later stage in software development life cycle and that by involve the customer in all development phases but when customer don't collaborate or not be available enough with Agile's team this may create a big challenge that leads to one of two major problems , the first one frequent change and the other the project failure .

2.1 Scrum Methodology

Scrum is one of most Agile methodology used, as mentioned by Larman (2003) Scrum life cycle consists of four phases: planning, staging, developing and release [8]. In the planning phase – the vision, expectations are being established as well as funding is being secured. Staging phase follows with identification of more requirement and their prioritization for first release. In the development phase a system ready for release is being implemented in a series of 30-day iterations. Release phase is operational when documentation and training for the product is being developed. Scrum is very precise on the length of iterations it is fixed to 30 days, thus providing a customer with the release every month . Larman (2003, page 134) lists strengths of Scrum methodology: "simple practices and management work products; individual and team problem solving and self management; evolutionary and incremental requirement and development, and adaptive behavior ; customer participation and steering; focus; openness and visibility; easily combined with other methods; team communication, learning and value building; teambuilding via daily Scrum" [8] .
Scrum has been successfully used over thousands of projects in 50 organizations producing significant productivity improvement [8] . Rising and Janof [8] suggest that "Clearly, Scrum is not an approach for large, complex team structures, but we found that even small, isolated teams on a large project could make use of some elements of Scrum .

Schwaber (2004) discusses that the iteration in Scrum is known as Sprint, this sprint as it has been mentioned previously has a duration of 30 days[8]. Every day starts with a Daily Scrum meeting where the team members report on the estimate of number of hours needed to finish the task. According to Subramaniam and Hunt (2006) a Sprint backlog contains tasks scheduled for the current iteration, if at some point the estimate of hours to finish the tasks on the backlog will exceed the amount of hours left, these tasks will be moved to the next iteration [8].

2.1.1 Scrum Roles

Scrum has three main roles :

Product Owner – the person who is responsible for creating and prioritizing the Product Backlog, choosing what will be included in the next iteration-sprint, and reviewing the system (with other stakeholders) at the end of the sprint.

Scrum Master – knows and reinforces the product iteration and goals and the Scrum values and practices, conducts the daily meeting (the Scrum meeting) and the iteration demonstration (the Sprint Review), listens to progress, removes impediments (blocks), and provides resources. The Scrum master is also a developer (see below) and participates in product development (is not just management).

Developer – member of the Scrum team. The Scrum Team is committed to achieving a sprint goal and has full authority to do whatever it takes to achieve the goal. The size of a Scrum team is seven, plus or minus two.

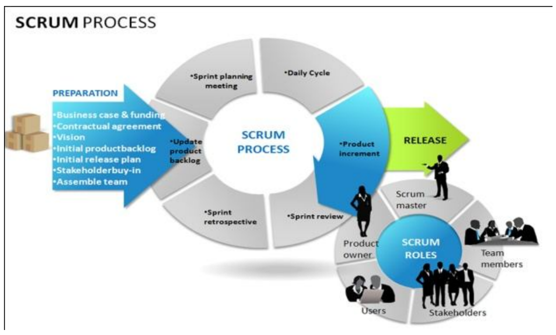

Figure 1 : Scrum Process [10]

3. RACI Matrix

Failure of software projects traditionally was inevitable matter in world-wide, failure happens when software project was exceeded predefined cost and schedule [9]. RACI can be used as a first step in institutionalizing the use of RACI matrix on software projects to minimize project delivery challenges and improve the chances of a successful outcome of business critical software projects [9]. RACI was first introduced in the 1950s, but it is ideally suited for today's business environment, which demands flexibility and adaptation in response to almost constant change. RACI was originally called by a more formal, academic name, the "Decision Rights Matrix" and is also known as "Responsibility Charting". The RACI tool is in wide use in companies as diverse as Proctor and Gamble, eBay, Amazon, and the U.S. Department of Defense [9]. It has been widely embraced by the software development industry and also by project managers worldwide. RACI is simple, and yet very powerful, because it gives the software project managers a language to talk with project team in more precise ways about their roles [9]; therefore it will make the workplace more fluid and more dominated by project work.

RACI is a term refers to:

- Responsible: The person is assigned to get the work done. May delegate work or may be supported by others. Only one person is responsible, think of the leader or manager [9].
- Accountable: The person who will sign off on work-packages/deliverables. Ultimately, only one person, but often includes others (e.g. A sign-off document requiring signatures of multiple approvers) [9].
- Consult (Contribute): Those people who contribute to the work by providing information (consultancy), either by providing information or directly working at the direction of the person responsible [9].
- Informed: Those people who need to be Informed, but not contributing (i.e. Do not have active role) [9].
- Although there are numerous modern and technical tools available for assignments of responsibilities, but the benefits of RACI can only be accomplished completely if the project management team understands and employs it by tailoring it to the organizational context. For example, if the work environment is volatile and people involved in the project keep moving in and out, role-based RACI is better. On the other hand, if the resources are stable, then name based RACI is good to be employed.

When RACI is shared with all the stakeholders, it also adds value by discovery any missing work-packages, missing roles, missing stakeholders, thus providing an opportunity for early correction. Not only that, but also RACI can be helpful as a checklist or reference when assigning resources, duration and cost estimates, to ensure that everyone who has a role in tasks has been properly accounted for, and that help project manager for controlling and monitoring project team. Globalization and the internationalization of the markets in IT-industry have increased competitive pressures on business enterprises. These pressures have led companies to engage in projects that are not only critical to their performance, but also vital for survival of the enterprise. The majority of the companies continually strives to produce better results by undertaking strategic projects. Unfortunately, recent industry trends have found that the majority of the projects exceeds budget, are completed past scheduled deadlines and do not meet original business objectives [9]. Several lines of research have evolved in the growing body of literature to help improve project delivery performance of organizations managing software development projects, but there is limited evidence that RACI tools are being employed effectively for.

3.METHODOLOGY

Proposed Model for Distributing and restricting roles in order to decreasing change in requirement depended on taking one of the most Agile methods used means Scrum and explained its process and roles and break up each phase to the activities which will be performed in this phase and assigning Scrum's role in each activity. To add more restriction on Scrum's activities in order to guarantee and controlling on project's productivity and reducing and elimination ambiguity that may happen on roles has been using RACI tool which has a positive impact on software projects for assigning the four roles which adopt it and applied these four roles with Scrum's activities to create RACI Scrum matrix. The Proposed model will manage and organize the work in an Agile Scrum environment and explain clearly the roles in that environment, thus project's managers can easily manage their team and decrease customer changes; therefore they gain project's productivity within predefined cost, schedule and quality. Scrum methodology has four phases, each phase includes many activities inside, so has been distributed RACI levels on Scrum's roles in order to create the proposed model.

Scrum Phases

Planning phase considers as initial phase; therefore we make customer holds accountable level in order to ensure that requirement comes to the next step after accepting it. Moreover holds accountable level for determinate the order of requirement in backlog in order to well understanding for his needs. While Scrum master holds accountable level in definition of the delivery date and definition of project team(s). In case of estimation of release cost all of them (stakeholder and Scrum Master) are holding accountable level.

- Development of a comprehensive backlog list.
- Definition of the delivery date and functionality of one or more releases.
- Selection of the release most appropriate for immediate development.
- Mapping of product packets (objects) for backlog items in the selected release.
- Definition of project team(s) for the building of the new release.
- Assessment of risk and appropriate risk controls.
- Review and possible adjustment of backlog items and packets.
- Validation or reselection of development tools and infrastructure.
- Estimation of release cost, including development, collateral material, marketing, training, and rollout.
- Verification of management approval and funding.

According to this, the Scrum RACI matrix will be as follows:

Table 1: RACI Scrum matrix in planning phase

	Product Owner	Scrum Team	Scrum Master
Development of a comprehensive backlog list.	I	R/A	I
Definition of the delivery date and functionality of one or more releases.	R/A	C/I	I
Selection of the release most appropriate for immediate development.	C	A	R
Mapping of product packets (objects) for backlog items in the selected release.	I	C/A	R
Definition of project team(s) for the building of the new release.	R/C	I	I

Assessment of risk and appropriate risk controls.	R/A	I	I
Review and possible adjustment of backlog items and packets.	R	A	R/C
Validation or reselection of development tools and infrastructure .	R/A	I	I
Estimation of release cost, including development, collateral material, marketing, training, and rollout.	R	A	I
Review and possible adjustment of backlog items and packets.	R	A	R/C

Architecture/High Level Design considers as the second phase in scrum methodology in other references it merges with the planning phase in one phase. The activities in this phase are:

- Review assigned backlog items.
- Identify changes necessary to implement backlog items.
- Perform domain analysis to the extent required to build, enhance, or update the domain models to reflect the new system context and requirement.
- Refine the system architecture to support the new context and requirement.
- Identify any problems or issues in developing or implementing the changes.
- Design review meeting, each team presenting an approach and changes to implement each backlog item. Reassign changes as required.

In this stage customer also holds accountable level in each activity related with requirement because here cost of change not expensive.

Table 2: RACI Scrum matrix in Architecture phase

	Scrum Master	Product Owner	Scrum Team
Review assigned backlog items.	C/I	A/I	R
Identify changes necessary to implement backlog items.	I	A	R
Perform domain analysis to the extent required to build, enhance, or update the domain models to reflect the new system context and Requirement.	R	A	C/I
Refine the system architecture to support the new context and Requirement.	R	A	I
Identify any problems or issues in developing or implementing the changes.	R	A	R
Design review meeting, each team presenting an approach and changes to implement each backlog item.	R/A	I	C/I
Reassign changes as required.	I	A	R

The Development phase is an iterative cycle of development work. The management determines that time, competition, quality, or functionality are met, iterations are completed and the closure phase occurs.

Development consists of the following macro processes:

- Meeting with teams to review release plans.
- Distribution, review and adjustment of the standards with which the product will conform.
- Iterative sprints, until the product is deemed ready for distribution.
- A sprint is a set of development activities conducted over a pre-defined period, usually one to four weeks. The interval is based on product complexity, risk assessment, and degree of oversight desired.
- Sprint speed and intensity are driven by the selected duration of the sprint. Risk is assessed continuously and adequate risk controls and responses put in place.

Each sprint consists of one or more team performing the following:

Develop: Defining changes needed for the implementation of backlog requirement into packets, opening the packets, performing domain analysis, designing, developing, implementing, testing, and documenting the changes. Development consists of the micro process of discovery, invention, and implementation.

Wrap: Closing the packets, creating an executable version of changes and how they implement backlog requirement.

Review: All team meeting to present work and review progress, raising and resolving issues and problems, adding new backlog items. Risk is reviewed and appropriate responses defined.

Adjust: Consolidating the information gathered from the review meeting into affected packets, including different look and feel and new properties.

Each sprint is followed by a review, whose characteristics are:

The whole team and product management are present and participate. The review can include customers, sales, marketing and others. The review covers functional, executable systems that encompass the objects assigned to hat team and include the changes made to implement the backlog items.

The way backlog items are implemented by changes may be changed based on the review. New backlog items may be introduced and assigned to teams as part of the review, changing the content and direction of deliverables. The time of the next review is determined based on progress and complexity. The sprints usually have a duration of 1 to 4 weeks. In this stage cost of change will increase, according to Agile's principle which it related to change in requirement in later stages and in order to consider this principle, we will let stakeholders holds accountable level if customer need to change the requirement.

Table 3: RACI Scrum matrix in Development phase

	Scrum Master	Product Owner	Scrum Team
Develop	I	C/I	R/A
Warp	C/I	A	R
Review	C/I	A	R
Adjust	I	I	R/A

Project Closure is the last phase in Scrum methodology when the management team feels that the variables of time, competition, requirement, cost, and quality concur for a new release to occur, they declare the release "closed" and enter this phase. This phase prepares the developed product for general release. Integration, system test, user documentation, training material preparation, and marketing material preparation are among closure tasks .

Table 4: RACI Scrum matrix in Closure phase

	Scrum Master	Product Owner	Scrum Team
Integration	C/I	C	R/A
System test	I	R/A	C
User documentation	I	A	R
Training material preparation	A	I	R

DISCUSSION

The solution which has been implemented depends on two major factors: the first one the Scrum methodology one of frequently methods used which it belongs to Agile methods which embraces of the customer change. Scrum methodology contains four phase planning, architecture, development and project closure. Each phase includes multi-processes inside, also Scrum has three roles (product owner, Scrum master and Scrum team), and that represent as variables in the second factor RACI matrix which it proved it has a positive effect on software projects, and assigning RACI levels as values of variables matrix.

CONCULISION

The changing requirements should be considered as a chance to improve the quality of the product and mitigate the risk of project failure. Feature of customer involvement, which it adopted by Agile methodologies could be a trouble if the customer is not available all time in SDLC. The customer doesn't understand his role clearly in SDLC; therefore change in requirement became inevitable in software projects. Change in requirement might happen during the development life cycle; therefore the project manager should make sure that changes are coming through proper channel (e.g. The competent authority). One of the most straightforward tools which can be utilized by project managers to establish a clearer structure of roles and responsibilities on a team is called RACI. This model still doesn't apply to industries for evaluation.

REFERENCES

[1] Rachel S. Smith ,Writing a Requirement Document .[Ebook] Available : http://www.cdl.edu/uploads/Qd/S6/QdS615B1DcnwRZlnSuTDnQ/writing-requirements.pdf. [Accessed 19 Nov 2014] .
[2] M. W. Bhatti, N. Ehsan, "An Investigation of changing Requirement with respect to development phases of a software project", 2010 International Conference on Computer Information Systems and Industrial Management Applications, pp 323-327.
[3] G. Kotonya and I. Sommerville, "Requirement Engineering: Processes and Techniques", John Wiley & Sons, Chichester, West Sussex, England, 1998.
[4] D. Zowghi, N. Nurmuliani, "A study of the Impact of Requirement Volatility on Software project performance", Proceedings of Ninth Asia Pacific Software Engineering Conference 2002, pp 3-11.
[5] Dr. S. Arumuga Perumal and G.Kavitha , "Changing Requirement -Correlated to Risk or Quality?" , IACSIT International Journal of Engineering and Technology, Vol.3, No.1, February 2011 .
[6] Vishvadeep Tripathi, Arvind Kumar Goyal," Changing Roles and Responsibilities from Traditional project management to Agile project management" , International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 2 Issue: 5 .
[7] Adel Hamdan Mohammad," Agile Software Methodologies: Strength and Weakness",International Journal of Engineering Science and Technology (IJEST).
[8] Besiana Alite , Nikolay Spasibenko , " Project Suitability for Agile methodologies " , Umeå School of Business , 2008 .
[9] P. M. Khan, Kaleem A. Quraishi , " Impact of RACI onDelivery & Outcome of Software Development Projects " , Fourth International Conference on Advanced Computing & Communication Technologies , 2014 .
[10] Cprime . "What is agile ? What is scrum ?", "www.cprime.com".[online] . Available: https://www.cprime.com/resources/what-is-agile-what-is-scrum/ [Accessed Jan . 26 , 2015] .

AUTHORS

Aymen Sulieman Elhady received the BSc and Msc Degrees in Computer Science From Elneelain University in 2011 and Khartoum University 2015 , respectively . He is currently working as a software developer in Nile Center Of Technology and Research - National Authority for Communications - Ministry of Science and Communication .

Hisham Mohammed Abushama received the PhD holder since 2006 from the UK . He is now working as assistant professor – Department of Computer Science – University Of Khartoum