USING SELF ORGANIZING TEAM IN AGILE SOFTWARE DEVELOPMENT

R.Rajadurai[1], S.Nandhini[2]

[1] Assistant Professor, [2] PG Student, Department of Computer Science.

Sri Manakula Vinayagar Engineering College, Puducherry.

[1] king8153@gmail.com , [2]nandhini.hayagreeve@gmail.com

***Abstract*—During the past ten years software development industry has faced a major change as different Agile software development methods have been started to use instead of traditional plan-driven methods. Agile methodology is an alternative to traditional project management, typically used in software development. It helps the teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Self organizing team are put forward to manage the challenges that are been faced in software development industry.**

***Keywords*— Self organizing team, Agile Software development, Sprints, Traditional plan-driven method.**

I. Introduction

Software engineering is about teams and it is about quality. It is also about communication. Problems to be solved are complex or large, that a single developer cannot solve them. Teams are needed to solve it out. A team does not have only developers, but also of testers, architects, system engineers, customer, project managers, etc. So we need to learn about what makes good teams, how to communicate with the customer and how to document not only the source code, but everything related to the software project. The problems you were solving were small enough so one person could master them. In the real world this is different: the problem sizes and time constraints are such that only teams can solve those problems.

II. AGILE METHODOLOGY

Agile Methods are increasing in its popularity. The motive of agile is emerging needs to be adapted and the changes in software projects are done quickly. There are two important levels of changes i) Requirement level and ii) People level. In Requirement level is to change rapidly with the emerging trends. In people level knowledge is fragmented and scattered with different people. Documentation is very expensive to solve this issue. It's difficult to update all information as the requirement change. The agile value propositions are created they are Individuals and their interactions, Working software, Customer collaboration, Responding to changes made when these values are used they give a distinct between Agile method and Traditional process for software development in new path.

A. Scrum

Scrum is used in agile software development. It is a framework, instead of providing complete, detailed descriptions of how everything is to be done on the project, much is left up to the software development team. The team will know best how to solve the problem they are presented.

A sprint planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint) instead of a set of Entry criteria, Task definitions, Validation criteria, and Exit criteria (ETVX) as would be provided in most methodologies. A sprint is defined as a 2-4 week increment of software development activities that delivers working software and the end of the increment. Scrum relies on a self-organizing, cross-functional team.

The Scrum Team is self-organizing in that there is no overall team leader who decides which person will do which task or how a problem will be solved. It is the issues that are decided by the team as a whole. The agile development teams are supported by two specific individuals: a Scrum Master and a Product Owner.

The Scrum Master is the coach for the team, helping team members use the Scrum framework to perform at their highest level. The product owner represents the business, customers or users and guides the team toward building the right product. Scrum projects make progress in a series of sprints, which are time boxed iterations no more than a month long. At the start of a Scrum sprint, team members commit to delivering some number of features that were listed on the project's scrum product backlog. At the end of the Scrum sprint, these features are *done*--they are coded, tested, and integrated into the evolving product or system. At the end of the sprint a sprint review is conducted during which the team demonstrates the new functionality to the product owner and other interested stakeholders who provide feedback that could influence the next sprint.

III. SELF ORGANIZING TEAM

Agile requires the team to be self-organize. Self-organizing teams manage their work and decide which task to commit. Team decides whom to do what task it's just assigning task to people. There are four principles of self-organization in holographic organization are minimum critical specification, requisite variety, redundancy of functions, and learning to Learn. Minimum critical specification defines critical factors which are needed to direct team and place some restrictions to team. Requisite variety and redundancy of functions it need for control system to match complexity and diversity to be controlled. Learning to learn is team ability to reanalyze problems, working method and reconsider the output. There are many advantage of using self-organizing teams in organizations three major advantages are performance effectiveness with respect to quantity and quality, member attitudes, and behavioral outcomes. Performance effectiveness means more effectiveness and productiveness, smaller response time, better quality and customer satisfaction, and more innovations. Most of the projects improvements are made in quality, amount of rework, inventory level, and on time deliveries. Self-organizing teams are effective because if any problem occurs in the project this team can react quickly instead of waiting for the command. This team has the rights to take decisions in case of any problem. Member attitude is increase job satisfaction, bigger commitment to the organization and the trust to the management. The Behaviour outcome has the turnover and safety to the organization.

A. Performance

Self-organizing teams are encouraging and give high production when compared to traditional work group. Six elements that make self-organizing teams successful are Autonomy, team orientation, shared leadership, redundancy, learning and communication & collaboration. Autonomy is helpful for team effectiveness. Team orientation describes the way how goals are faced by an individual and by a team. When an individual oriented it face the goal which give many ideas and not sure whether it will suit and it will achieve the goal. But in team discussion are made many ideas are given they can predict whether it can achieve the goal so the best solution is team oriented goal. Shared leadership says ideas must be shared within the team any individual person does not knows all knowledge. Redundancy is that anyone in team will have knowledge of the work of their team member. In traditional work group each one will do their own work which will take more time for project delivery and many other issues in it. Learning is to gain knowledge from their team member about their work progress. Important factors to perform these actions are the communication and collaboration. When there is lack in communication then their work will not satisfy the clients, collaboration is combining their work and during decision making. For Self-organized team this communication and collaboration are most important roles to be performed.

Fig3.1 shows how the Self-Organizing Teams are built in order to finish the project successfully. In Autonomy stage teams are Authorized and it needing someone to be protected. The Communication and Collaboration stage teams must have close contact with customers, they have to work together, share information daily with their team and visualize the progress. The Redundancy and Learning stages get their needed knowledge from communication and collaboration stage.

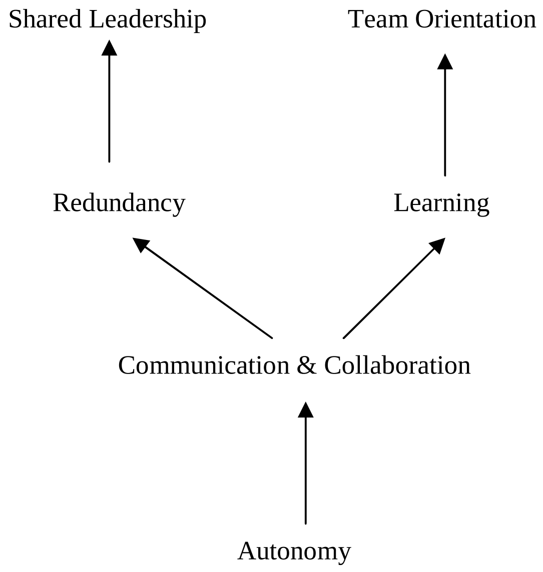


Fig 3.1 Building Self-Organizing Teams

With these details Redundancy shares the working responsibility and accepts uniformity. Learning need feedback continuously, testing and integration when these are fulfilled then there is iterations to be followed at end of each iteration review are conducted and progress is tracked. Team Orientation is followed by allowing team to participate in iteration planning and goal setting. It also has to guide team with mission. Shared leadership is practiced with lead-and-collaborate principle and Cross-functional teams with knowledge. These are the things that are followed in each stage while building Self-Organizing teams.

IV. GROUNDED THEORY TECHNIQUE

Grounded Theory (GT) defines "as a systematic guidelines use to gather information, categorizing, coding and integrating to perform a theoretical analysis of Empirical problem. GT focus in on human experience and interactions in particular field. It performs their operations in a cyclic manner. The terms that perform the operations are Minor Literature Review, Theoretical Sampling, Open Coding, Constant Comparison method, Memoing, Core Category, Selective Coding, Theoretical Saturation, Major Literature Review, Sorting, Theoretical Coding and Write up.

In Minor Literature Review a literature review is made to know what is been done before and their issues. Theoretical Sampling is a process to collect code and analyse data and decide what to be collected next. Open coding is the first step of analysing and collecting key points from raw data. Constant Comparison Method is process in which codes are raised in interview conducted it's been compared with the other interview and observe the high levels of data. Memoing is capturing the changes in every categories of process that are performed. Core Category is to search for the best result which best suits and easy with other categories. Selective Coding is ceasing open coding and using selective coding to know whether the code for code category are closely related to core. Theoretical Saturation is during data collection and analyzing a particular category it leads to different results in that case the data collection is stopped and coding will be started. Major Literature Review is that many reviews are conducted to know how it relates to emerging theory. Sorting is arranging the theoretical memos. Theoretical Coding can also be used as a framework to describe the relation of categories to each other and its integration into theory. Write up is writing theory with the result of sorting and theoretical coding.

V. ROLES OF SELF ORGANIZING AGILE TEAM

Grounded Theory is the core of Self-Organizing Agile teams. It describes how software development teams get their roles and perform the acts with the environment factors. The Roles are Mentor, Co-ordinator, Translator, Champion, Promoter and Terminator. Mentor teaches the new team about the categories are raised from the data analysis that is been done. Co-ordinator is representative of the team to manage customer expectations and collaborate with team. They help to change request given by the customers. Translator understands and translates the business language and technical language which are used by the customers and team. It improves the communication between team and customers. Champion is to secure support from senior management. Senior management are convinced with cost effectiveness, time to market, customer demands and process improvement. Promoter is the one who promotes agile with customers to secure their involvement and collaboration to support the self-organizing agile team. Terminator will identify team members threatening functioning and productivity also involve in removing such members from the team.

VI. CONCLUSION

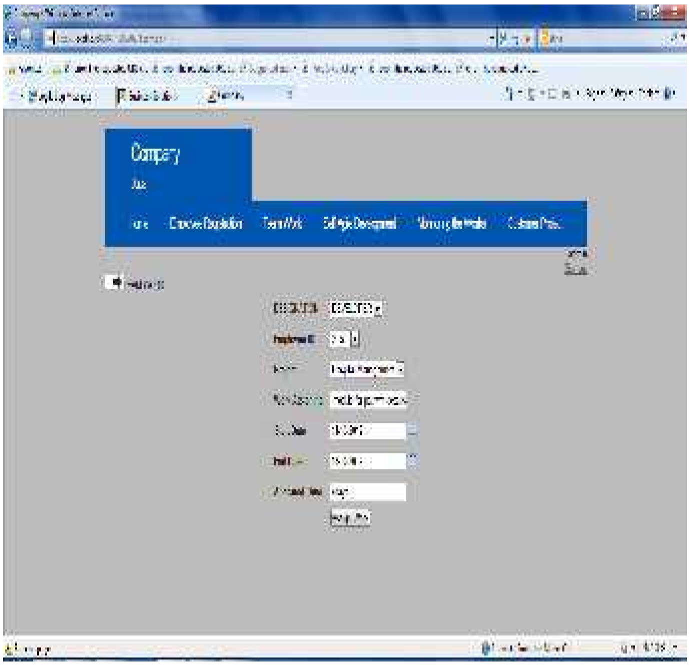
In Self-organizing Agile team roles where single team can achieve and maintain self organization by separating it to different needs of the team such as need for training, senior management support, customer involvement, etc. All software development is performed with multiple self-organizing Agile teams in Agile organizations. A theoretical framework is created first then its analysed against empirical data. The empirical evidence shows that most of the practices of theoretical framework are useful for building self-organization in teams. There are three practices to the theoretical framework first, important to learn else team are forced to make compromise in project. Secondly, having tasks small increases redundancy then lost work is small when team member is away. Thirdly, tasks are prioritized; each team members will know which task is most important and decide according to chose the task. There will be a close contact with customer to know their change request. When this self organizing teams are used works that are been performed by the team are monitored and their works are updated in simple manner. By using the Grounded Theory the team roles can be changed this is done when the team roles are misguiding their work and it is been followed throughout the project once the work is misguided then the team roles are changed. In future Agile Software Development companies can explore in organization wide roles which enable self-organization at an organizational level.

VII. IMPLEMENTATION DETAILS SCREENSHOTS

HOME PAGE:

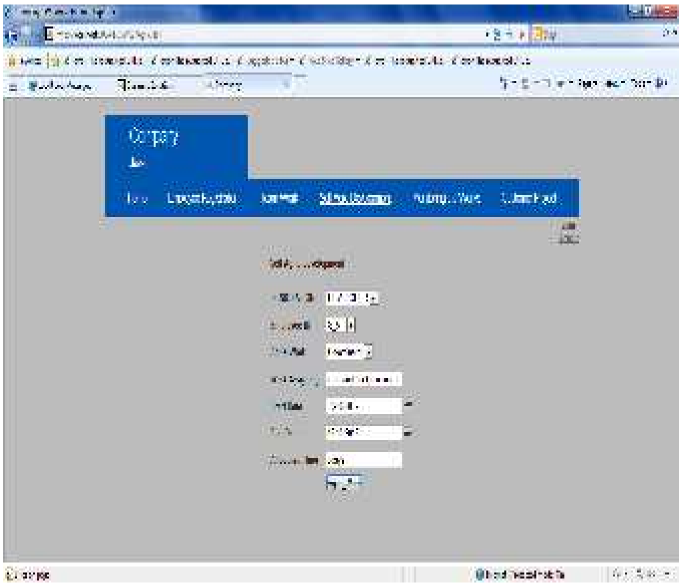
It is the homepage of the company in which the user login and enter into the company profile to register and give their need to the employee.

TEAM WORK ALLOCATION



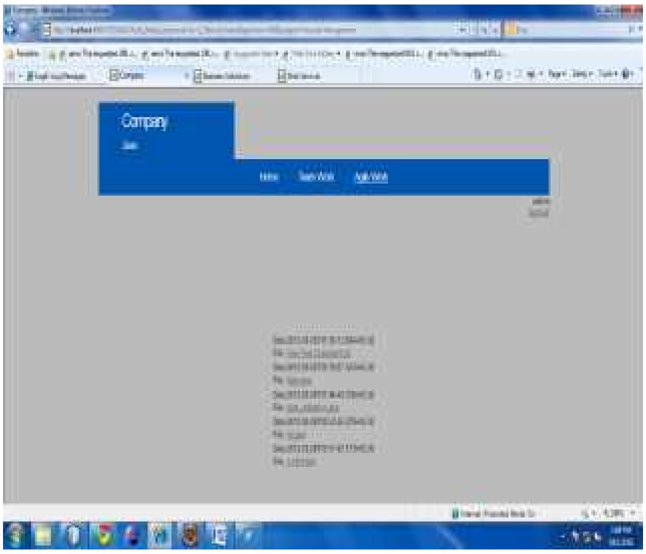
Work is been allocated to team. This gives the option how many days the work should be completed it is the client choice to chose it.

AGILE WORK ALLOCATION



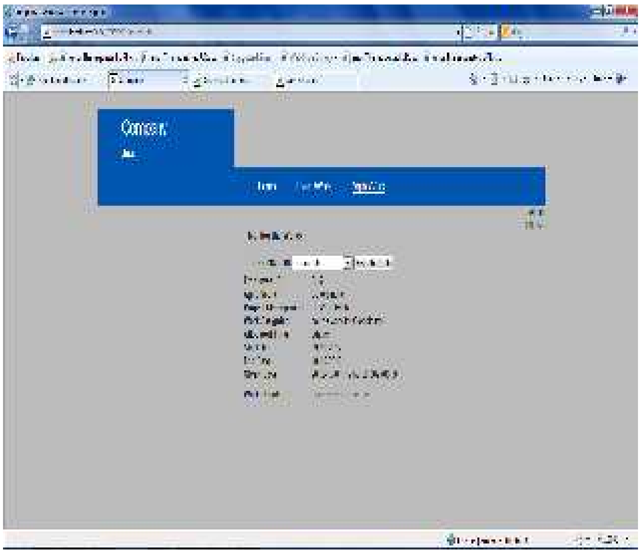
The Agile work allocation is work are given to the self organizing roles they are the mentor, coordinator, translator, promotor, champion and terminator. Client can select which role to do the work and allocate the work to them with the days allocation. Within that the work should be completed.

WORKS



This shows the work allocation. Client can view what are the works been allocated and how far is the work been done. Work updation is been known to the customers through this .

MONITOR AGILE WORKS



Monitoring work is to know what is the current process goes in the project. This can be viewed by the client. They can view whether the self organizing roles are performing their work correctly. If there is any misfits grounded theory make the alternate role to perform the work. If such things happen it can be viewed in this monitoring.

REFERENCE

[1] J.Highsmith and M. Fowler, “The Agile Manifesto,” Software Development Magazine, vol. 9, no. 8, pp. 29–30, 2001.

[2] T. Chow and D. Cao, “A survey study of critical success factors in agile software projects,” J. System Software, 2008.

[3] A. Cockburn and J. Highsmith, “Agile software development: The people factor,” Computer, vol. 34, 2001.

[4] R. Martin, Agile Software Development: principles, patterns, and practices. NJ: Pearson Education, 2002.

[5] K. Schwaber, “Scrum guide,” Scrum Alliance Resources, 2010.

[6] H. Sharp and H. Robinson, “An ethnographic study of XP practice,” Empirical Software Engg., vol. 9, 2004.

[7] J. Highsmith, Agile Project Management: Creating Innovative Products.USA: Addison-Weasley, 2004.

[8] XP, “Extreme programming: A gentle introduction,” World Wide Web electronic publication, last accessed on 23rd Oct 2010.

[9] S. Augustine, Managing Agile Projects. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[10] Hoda, R, Noble, J, Marshall S. The Impact of Inadequate Customer Involvement on Self-Organizing Agile Teams. Journal of Information and Software Technology, Vol. 53, 521-534, May 2011.

[11] Hoda, R, Noble, J, Marshall S. Organizing Self-Organizing Agile Teams. ICSE, Cape Town, South Africa, 2010.

[12] Hoda, R, Noble, J, Marshall S. Using Grounded Theory to Study the Human Aspects of Software Engineering. HAoSE workshop at SPLASH, Reno/Nevada, USA, 2010