Optimization of Software Reuse in Agile Software Development (OSRAD)

Jayasudha Subburaj\textsuperscript{1}, Soundarya Veni,C\textsuperscript{2}, Shanthi Palaniappan\textsuperscript{3}

\textsuperscript{1,2,3}Department of Computer Applications
Sri Krishna College of Engineering and Technology
Coimbatore 641008, Tamil Nadu, India
e-mail: jayasudhasubburaj@gmail.com, soundaryaveni777@gmail.com, shanthi.slim@gmail.com

Abstract
Software reuse plays a major role in agile software development. In the modern changing world, software development faces lot of challenges and the speed of the software development need to be increased. Reusability of existing components will reduce the time, cost and improve the productivity. Reusability in Agile development process helps to avoid the repetition of same work. In this paper a measure is taken to reuse the existing component and parallel development of same component in multiple projects is implemented. Different role of the agile project development is analyzed and the developer developing same component works for multiple projects. Finally the role of full stack developer in agile software development is also analyzed.

Keywords: Software Reuse, Agile Software development, Full stack developer

1. Introduction

The motto of this paper is to understand the concepts of Software Reusability and the software development using Agile Methodology. Now a daze a corporate world face more challenges in developing new technologies. The software reusability helps in overcoming such difficulty and helps the programming developers to come up with different new ideas and frame works. That can be implemented in short term duration. The software reusability plays a major role in developing new
software’s with increased product quality and decreased product cost which make software reusability to reach higher end success for good quality software. Software reusability will be done on existing system. Those existing system will be restructured and modified with new ideas so the existing code will be updated newly. As benefit of software reuse developer gets good quality software with increasing productivity, quality, reliability, and decreasing of cost and implementation time.

Software reuse, Updating the new software or implementing the software based on existing system. In the current trending technology software reuse plays a major role in developing new software products. Software reuse will be done for various applications that have same methodology with different ideas. It can be implemented easily with existing source code with reusability techniques, where the developer should have clear ideas and knowledge about the software reuse. So that it make developers to come up with new ideas for different problem statement with small changes in code that can be reused later. By this way we are reducing our time and space complexity in the software development process. The key role of software reuse is to “code once and use it later” for others in sometime. The programming developer should be trained well and good in software reuse process. If the approach are good and well planed in decision making the software reuse and the productivity will be increases. So, the project development time will also be reduced along with the project cost. Finally by software reuse methodology all the software productivity and the cost of the project become beneficial for both the end users.

Originally a Software is built completely from the scratch to a great extend. In software reuse methodology the developed software document contains all the source code, document design, etc are copied and adapted to fit new requirement. Based on the two major verification various software are reused. Verification goal is to make code effective. Verification specifically done for software reuses and it verifies the reuse components between projects, environments, hierarchies, and also reuses base class, patterns and methodology. In order to verify good quality software the software reuse follows two major paradigms:

**Horizontal Reuse:** In horizontal verification the software verifies the project details. Without any changes in the role of their project which holds the details about responsibilities, structure and use-case.

Horizontal reuse does checks the inputs and the output of the program and verifies the code coverage and message passing. The API inputs presented to user or test writer to be used for verification process. Horizontal verification designed for more
than one unknown use-case in the program and it’s typically achieved by adaption and flexibility with good configuration.

**Vertical Reuse:** The vertical reuse does verification on various different role of the code and its responsibility, structure and use-case. The API checks the input in different layered with high level possibilities. Sometimes inputs will be verified and message will be covered on the base of data process. The vertical reuse achieved by encapsulation and extensibility of the code reusability and allow user to do other process.

The term AGILE defines “able to move quickly and easily”. Agile is one of the most famous software development processes that are used in every face of project development in software field. Using agile software methodology a programmer can develop their project with a short duration of time. It can be adaptable for all small, large, and big end projects. When comparatively to other modules in software development agile module has more benefits in project development and implementation.

The developer can change their ideas in the mid of the project where in comparatively it’s not allowed in other methodology like water fall module. A comparison is done to get a clear view and new idea about the good quality software. Agile success is based on agile life cycle where it gives clear idea about the next process. A life cycle consists of six different phases (1.Requirement, 2.Analysis, 3.Plan, 4.Design, 5.Implentation, and 6.Monitor). Each and every phase does some process to improve software development. Agile approaches various methodologies in software development field (eg. Extreme programming, Scrum) are following agile methodologies. Agile Success is based on four Major factors they are

**The Four major role of Agile Methodology**

- Team coordination and Motivation
- Enough planning
- Workspace considerations
- Missions success

**2. Literature Review**

Agile software methods are listed and a comparative study is done for ten types of agile software development methodologies. The pros and cons of each method is identified and also the common process if identified in the each method. [1] Among all the ten methodology the SCURM is the one of the best method of agile is well proved.
Different agile software development methods problems were identified and a new style of methods identified with all the merits of the different models. [2] Outcome of using agile methods for software development is analyzed clearly.

Semantic based software component reuse is specially mentioned in this paper. [5] This semantic helps to retrieve the relevant component by understanding the similarity of the component easily.

Relevant component plays a major role in effective utilization of the software. This will help to improve the precision and recall of the retrieval. [6] Genetic Algorithm helps to find the relevant component and Particle Swarm Optimization helps to build the structure of repository.

Customer involvement is one of the important merits of agile; this paper helps to identify the progress of the project at various stages and gives the report to customer at every stage. [3] This helps in doing changes at the earlier stage of the project which reduces the cost of stages.

The role of full stack developer is most important since the developer need to have complete knowledge. [4] In this paper full stack web developer role is fully analyzed and their knowledge level is categorized.

Ontology based repository creation helps to create a systematic software storage architecture. [7] The time taken to retrieve software artifacts will be reduced since the components are arranged semantically.

Roles in software development improve the process speed and increase the productivity. [8] Each software developer has to play multiple role and also same role in multiple projects for a common cause.

Storage of components in the Cloud helps to retrieve the components from anywhere. [11] The scale of software size can be increased any time since the repository is stored in the cloud.

3. **Research Objectives**

   1. To improve the utilization of various resources during project development.
   2. To reduce the cost and time of the project development using software reuse.
   3. To examine the role of full stack developer in the project development and communication with the customers of different project.

4. **Research Methodology**
Agile software development mainly, Scrum gives importance to various roles. The productivity of the human resources can be improved by using them parallelly for different projects at the same point of time. The proposed diagram shows the role of different people which saves the time and cost of the process of software development of multiple projects.

In the fig 4.1 each project will have list of product backlogs, from the product backlog of different projects common backlog is identified. Sprint backlog of each common product backlog is given to one developer and after the sprint process final component is identified and stored in the component repository.

Fig. 1 System Design

Component repository will have the set of all the common sprints which can be used by multiple projects. As per SCRUM each sprint will undergo daily meeting and weekly meetings with the customer. The role of sprint developer is utilized well since the developer will develop common sprint for multiple project.

The developer should act as a full stack developer since he needs to communicate with the users of different projects. A full stack developer should have the following skills

1. Should be good in developing both front end and backend development skill.
2. 90% of knowledge in one domain and 75% of knowledge in all the section of the project.
3. Should act as a developer, analyst, tester, User Interface designer and repository manager.
5. Results and Discussion

This Agile based Software reuse process with same role in different project will save the time and cost of the project and increase the productivity of the projects.

![Graph showing time taken for completion of projects]

**Fig. 1 System Design**

Around 5 different projects has been taken and implemented the development of common components and this shows around 25 percentage of time has been reduced and the project release on time.

![Graph showing costs incurred for projects]

**Fig. 1 System Design**

The above graph shows that a reasonable amount of time in developing the common sprint has been reduced since one common sprint is developed and utilized by the entire project which also helps in smart utilization of resources.
6. Conclusion

The research understands the Agile software which helps to create innovative projects with dynamic updates of the user requirements. Software reuse will reduce the cost and time of developing the same product backlog for multiple projects. Effective utilization of resources will improve the overall productivity of the project and increases the customer satisfaction.

7. Limitations and Future Research

The different project will have common task which is used in different platform hence common development is not possible for those types of tasks. The availability of full stack developer is less and hence domain developer has to develop the common module. Since the projects were different and the developer has to communicate with the different customer, the task is tough in this regard.

In future the common process stored in the repository will be utilized by different project this will even reduce the cost of developing the task. The structure of the repository has to be designed in a way which the repository will accept the all types of solutions for the common problem. Hence the reusable components will be stored and retrieved effectively.

References
8. Łukasiewicz, K. and Miler, J., 2012. Improving agility and discipline of software development with the Scrum and CMMI. IET software, 6(5), pp.416-422

