Crowd Sourced Software Engineering; an Agile Approach

Ann Joseph, Reshma Sebastian, M Soumya Krishnan

Abstract: In this research, a methodology for crowdsourcing was developed to incorporate these methods into software development. The entire SDLC crowdsourcing model is being developed and attempted to integrate with Agile. This is done to speed up the process of agile development, as agile is a software engineering methodology based on rapid application development (RAD). The workers of the crowdsourcing process are assigned individually the roles of the agile methods, which perform the agile tasks and produce the results.

Index Terms: Crowdsourced Software Engineering(CSE), Software development tools, Scrum implementation, Agile Methodology.

I. INTRODUCTION

Crowdsourcing is a new approach to outsourcing that takes tasks to an undefined, large group of people as an open call. Crowdsourced Software Engineering is the process of performing any external software engineering task by an undetermined massive group of online workers through an open call format. The process recruits worldwide online employees to work on diverse software engineering task such as requirements extraction, design, coding and testing, deployment and maintenance. In an adjustable development environment, this model claims to minimize development time and costs along with reduced defect rates. The proposed crowdsourcing model represents the steps in the development of a process for the gradual design of crowdsourced application. Our first step towards developing an understanding of the crowdsourcing phenomenon was to study some of the key works in this area. This developed a common vocabulary and understanding of the concept of crowdsourcing among researchers. Because of the special complexities of software development, crowdsourcing in a software environment differs from other environments. This clearly has indications for which characteristics of crowdsourcing in a software development environment should be studied. In addition because crowdsourcing involves a number of actors or stakeholders, it is useful to take different stakeholder aspects to study the development of crowdsourcing software from diverse perspectives of stakeholders. We have identified three different aspects, namely the perspective of customers, employees and the online market platform.

II. LITERATURE REVIEW

Crowdsourcing is a type of participatory online activity in which an individual, an institution, a non-profit organization or an organization proposes the voluntary undertaking of a task to a group of people with varying data, heterogeneity and numbers through a flexible open call.

In an investigational study conducted by Nitasha Hasteer, Noshiba Nazir, Abhay Bansal and B.K Murthy [3] they reveal that crowdsourcing software development is still in its early stages. The analysis they conducted showed the benefits as well as the concerns that need to be examined. In 2018 Fiza Siyal and Xavier Franch[4] conducted detailed study and they arrived at a conclusion that crowdsourcing is a concept used to solve problems in a variety of fields. They selected the CSE domain to explore its environment by considering the CSE process, the actors, the objectives and the tendencies of the participating actors. According to [2] the systematic review shows how crowdsourcing projects are designed and implemented in the following aspects: Task implementation, quality management, and use of the platform.

Based up a comprehensive survey organized by Ke Mao, Licia Capra, Mark Harman and Yue Jia, [6] Crowdsourcing may be used to support software engineering and investigation activities. And also they show the use of crowdsourcing for software engineering activities.

III. BACKGROUND STUDY

A. Crowdsourcing Perspectives

In general, three types of actors are involved in Crowdsourced Software Engineering(CSE): Employers who have the necessary software development work to be carried out; employees who participate in the development of software; platforms that provide an online marketplace where requesters and employees can meet. Fig. 1 shows these actors and their relationships.

The expanding popularity of crowdsourced Software Engineering can be categorized to three different stakeholders:

Revised Manuscript Received on May 18, 2019
 Ann Joseph, Dept. of Computer Science and IT, Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Kochi, India.
 Reshma Sebastian, Dept. of Computer Science and IT, Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Kochi, India.
 M Soumya Krishnan, Dept. of Computer Science and IT, Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Kochi, India.
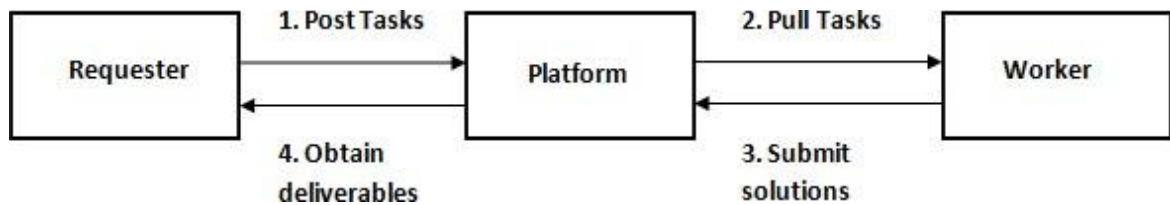
Fig. 1

Requesters: Applicants are companies or individuals that have software tasks to develop. Customers may have different motivations for the development of crowdsourcing software; for example, a company may want to expand its software capacity temporarily. Through "outsourcing" some works to the crowd, the company can increase and reduce its production capacity quite flexibly.

Workers: They are the people who do the work. They develop the pieces of software that are published by the client.

Platforms: The platforms offer an online marketplace to meet employees and customers. TopCoder is the largest platform for the development of crowdsourcing software, with a developer community of over 600,000 as of January 2014. There are also indications, however, that only a small fraction of its registered users participate in the development of software.

IV. CROWDSOURCING SUPPORT TOOLS

Given the distributed nature of crowdsourcing software development, a strong development environment is needed to accelerate the design, coding, testing and implementation of software over a distributed and diverse infrastructures. All members of the participating team should have access to the same software development environment tailored to a specific project. In addition, many submissions from these team members need to be screened, assessed, evaluated and merged quickly. Software crowdsourcing attempts therefore need improvements to general software development tools for coding, testing and implementation in order to support automatic project construction, integration, performance evaluation and security checks.

A. Software Development Tools

A crowdsourcing IDE integrates tools for requirements, design, coding, compilers, debuggers, analysis of performance, testing and maintenance. Software development tools assists the management and evaluation of large-scale system logs and help team members solve software problems and improve system reliability by logging.

B. Social Networks and Association Tools

Facebook, Twitter, wikis, blogs and similar web-based tools allow participants to share and collaborate with each other. For example, organizers can create an effective crowdsourcing team using Facebook profiles, even if they don't know each other.

C. Tools for Project Management

Project management in Crowdsourcing should assist project cost estimation, development groundwork, decision-making, error monitoring and repository maintenance, all

of which are specialized in the dynamic developer community background. In addition to these regular functions, ranking,

reputation and award systems must also be included for products and participants.

V. CROWDSOURCING: SOFTWARE DEVELOPMENT APPROACH

The participation of the crowd depends on the motivation of the contributors and the incentive mechanisms developed by the applicant. The methodology in which these individuals or organizations co-ordinate can vary depending on the model underlying them. These models are called crowdsourcing software models. The study about crowdsourcing proposed a typical methodology for software development in crowdsourcing.

A. Development Process

The software designing procedure for crowdsourcing usually follows a waterfall model as depicted in Figure 2. The designing procedure initiates with a requirements extraction stage. At the time of this stage, the project owner, who actually appear from the crowd, is responsible for handling the following stages and conveying with client organizations in order to decide their project strategies, task plans and cost estimate. The requirement statement is then described and moved to the next phase. The requirement statement is considered as the input to the next stage. The waterfall model divides the development process into several components. The finished components are combined during the assembly phase and additionally certified through system-level testing. The fully functional solution is finally deployed in the quality assurance environment of the customer during the implementation stage. All developed assets will be given to the customer after the user acceptance test.
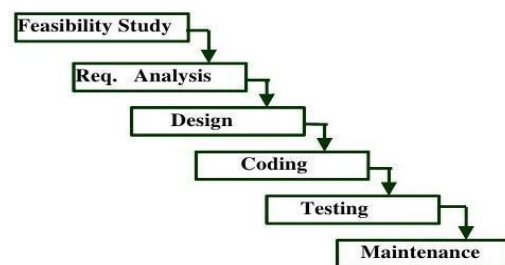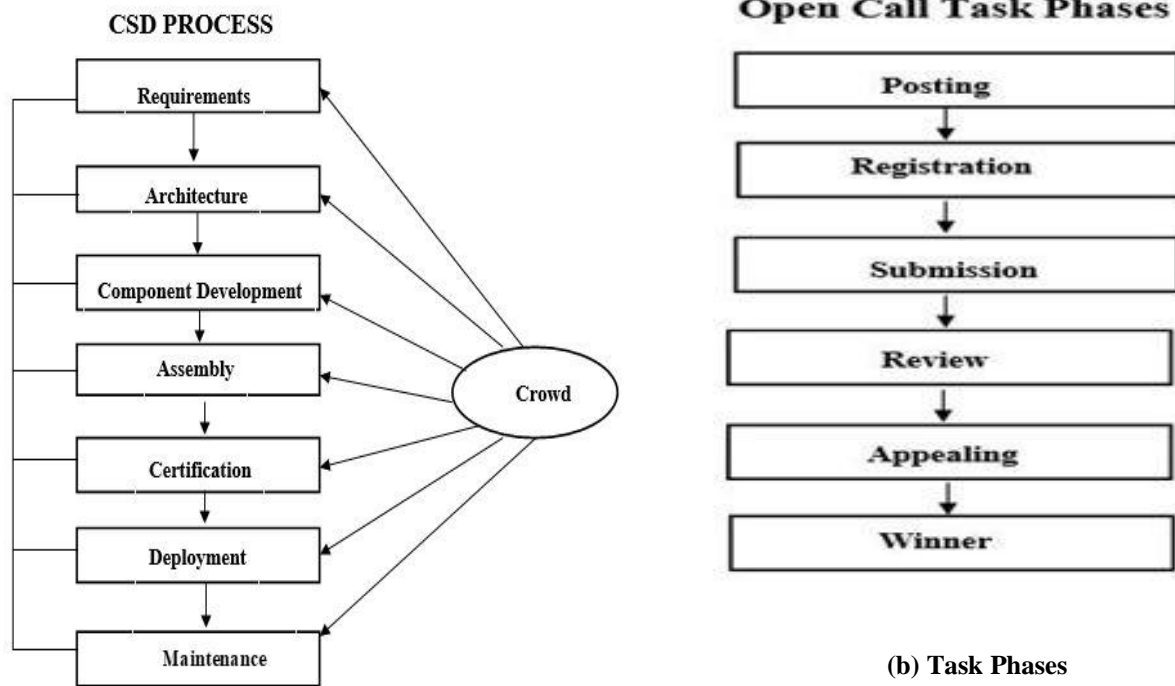

Fig. 2

CSD PROCESS

Open Call Task Phases

(b) Task Phases

(a) Crowdsourced software development process

Fig 3

B. Open Call Tasks

The CS method applies an open call pattern to find online employees to contribute to the published problems. It is generally in the form of a contest. Every single crowdsourced development job is categorized into an open competition. The complete process time span for a single task is usually one or two weeks. The method for performing an open call is shown in Figure 3. Initially, a problem or job categorized according to its kind of development is published on the website with instructions such as explanation of tasks, payment amounts and time periods. Time periods include two important dates: Registration and submission time limits. The participants who are ready to participate should publicly broadcast their choice by registering the competition. So that the online workers can notice the details of opponents, including previous performance and expertise rankings. After the deadline for submission, all the solutions submitted are collected and evaluated by peer review using predefined screening and scoring cards. The screening solutions are scored. After completing the evaluation process, the contestants are privately informed. The contestants have a chance to argue with the reviewers if they are not satisfied with the results. The corrected results are ultimately published on the website and participants are rated. The first champion receives the complete amount and the runner-up will receive half that amount.

C. Participation Incentives

A reward in the form of cash is a dominant element for the

involvement of the public. But not everyone registers the award winning competition. Various types of rewards are there for contestants to carry out development tasks. The incentives include the acquisition of skills, feedback, friendship, fun earning money and in a view of achievement. So that the two top task champions can usually be rewarded with money, developers often do not get any money for their efforts. However, developers can find it exciting by competing in the task competition and the developer's posts are tested by the specialists the specialists of the community and further reviewed. They are thus rewarded with pleasure, recognition and expert rating to improve their talent.

VI. PROPOSED AGILE METHODOLOGY IN CROWDSOURCING

In order to build the initial CRM methods, the designer takes the list of terms or concepts and organizes them into tasks and solutions and produces an initial where tasks are the goals to be accomplished and solutions are the bottom up for the final product. For example, taking the case study Student Registration System (SRS) into account, the requirements are first obtained. Then begins the analysis process, which includes interviews and the selection of participants. The participants are subsequently assigned to staff. A hierarchical task diagram involves the Task Reduction Model.
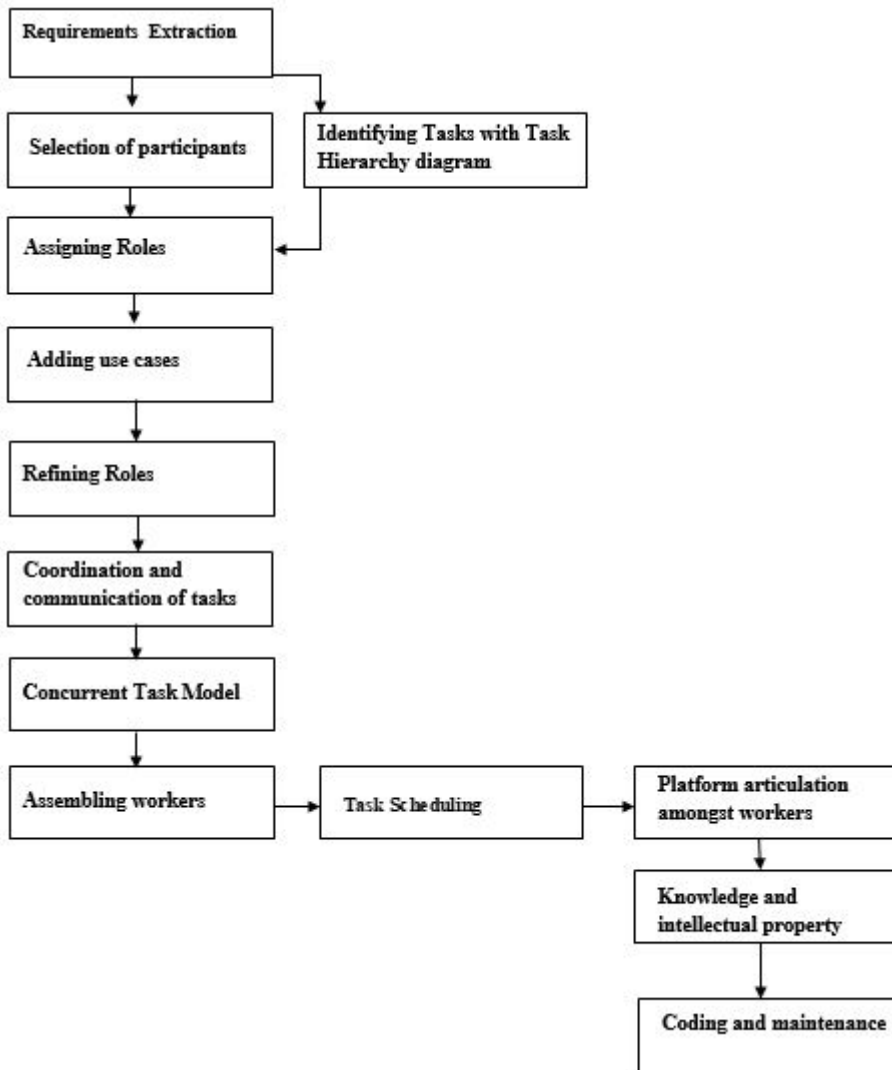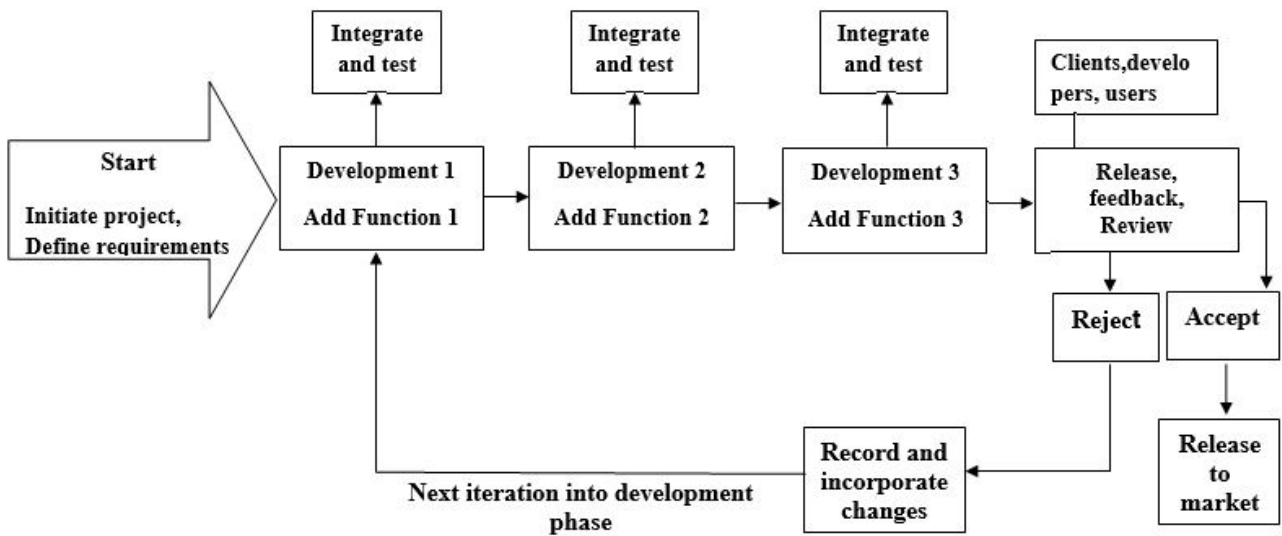
Fig 4. Participation Incentive Model

Diagram 1. Crowdsourcing Methodology Block Diagram

After the task reduction has been completed, roles are assigned and use cases are developed. The roles are further refined by allocating the specific task(s) to specific roles.

VII. RESULT ANALYSIS AND DISCUSSION

Obtaining users requirements before moving to the next steps of the CRM becomes an unavoidable activity. Interviews, participation level and the platform on which the CRM must run are submitted on paper. The process comprises,

1. Discovery of Requirements.
2. Classification of requirements and organization.
3. Prioritization and negotiation of requirements.
4. Specification of requirements.

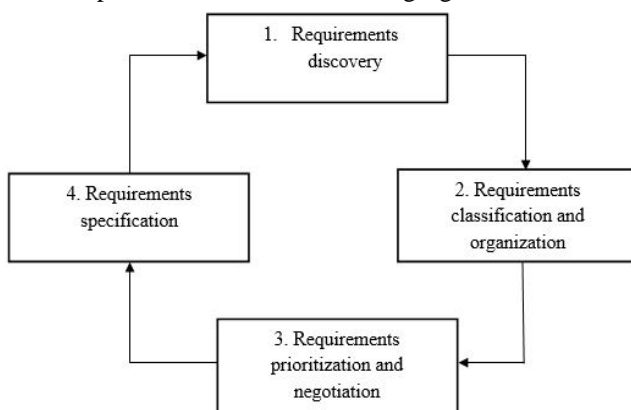These steps are shown in the following figure.

Diagram 2. Tasks Hierarchy Diagram

A. Assigning Roles

The conversion of objectives into functions in general is one-to-one with each objective depicting a function. Although, there are circumstances in which it is useful to have a single role, including convenience or efficiency, in charge of several goals. The following is a mapping of the objectives of our above illustration to a set of functions.

1. Administrator
2. Students
3. Faculties

B. Refining Roles

In this phase, roles are refined. There are already many applications-specific roles that have been allocated to entities. Now that roles in the order of similarity and action are grouped, the roles identified are refined in groups.

VIII. SCRUM FRAMEWORK

Scrum architecture enables to implement agile development methodology. In contrast to the waterfall software development life cycle, the iterative development process is the unique feature of Scrum. Development is done in several stages. Each phase results in a ready-to-use outcome. At the end of each phase, the ready product is delivered to the customer (called sprint in Scrum architecture). The customer's feedback helps to disclose possible problems or to change the initial plan if necessary.

Fundamental roles involved into the procedure are:

1. product owner
2. scrum master
3. scrum team

scrum "ceremonies" are the project phases, also called sprints (daily scrum meetings.) scrum "artifacts" contains a product backlog, a prioritized list of client demands; a sprint backlog or a task list for accomplishing the priority elements of the product ; and the burndown graph shows how much work remains as the backlogs.

A. The process

The project manager compiles the project backlog. on the top few priority items a sprint planning team is working in the backlog. the team must work within a time limit. daily meetings allow the scrum master to track progress and guide the team to complete the sprint. the sprint completion results in a deliverable. an evaluation of the resolved sprint precedes the task on the next sprint, which is the succeeding priority classification. the procedure continues until the project is completed with a reduced budget or time period.

B. Implementation analysis

The successful implementation of an agile /scrum approach requires considerable training and skills. Several project teams do not completely understand the requirement for training and skills or they doesn't want to make an attempt and try to do agile / scrum automatically without completely understanding the concepts after it. An agile / scrum technique may also require a certain amount of organizational modifications to succeed. Business users need to cooperate in a spirit of trust and partnership with the development team. this may require the removal of certain organizational barriers, which make this difficult or impossible. Scalability – an agile / scrum approach for large, complex projects can be difficult to scale. Certain models are there to do this (scrum- of- scrums, less and safe are examples), but none of these are trouble free to deploy cookbook solutions. An agile / scrum architecture cannot be entirely suitable for projects that require a more planned approach. Although, there are several ways to create a composite approach that combines a traditional plan- driven approach with an agile / scrum approach to the right extent. An agile project primarily involves:

1. requester
2. platform.
3. crowd

The requester is itself the scrum entity that manages the development of the Agile process. The iterations are periodically assigned to variable and fluctuating crowds, such as a chronograph. The idea is to fulfill few or many of the requirements that the SCRUM has to fulfill, because sprinting in Agile software development requires many iterations and when the project load exceeds the maximum capacity of the development servers, it becomes a final option.

Scrum begins with the assumption that the development of software is too complex and unpredictable to plan exactly in advance. To guarantee visibility, assessment and adaptation, empirical process control must instead be applied. The various environmental and technical variables must be constantly monitored so that they can adapt flexibly to changes. This can be attained across an iterative and gradual process of development. The initial requirements are set out in group.

1. The results of group 1 are tested and integrated and then transferred to crowd.

2. The development phase goes on until the hierarchical diagram is covered. Stakeholders can review the results and submit feedback after the development phase. The product can be accepted or rejected according to the feedback. When the product is rejected, the whole process is repeated by adding changes.

C. Scrum implementation in crowdsourcing through Online Shopping Management System

Online shopping is a form of electronic shopping where the buyer is usually online via the internet to the seller's computer. No intermediary service exists. The purchase and sale transaction is completed in real time electronically and interactively. This online shopping application or software dispatches the product directly to the customer's house.

Proposed System:

The following activities are part of the development of this application

- Secure customer account management provision.

- All products can be accessed via e-mail, accessories such as home decors and electrical appliances.

- Customers can check the total amount of the selected items with the help of shopping cart.

- New offer updates are send to customers via mail.

Modules:

This application contains 4 modules

1. Employee
2. Admin
3. Customer
4. Reliability and authentication

Customer: This module includes product and offer details and product prices. The customer should register in the application, for this registration form is created, new users must register and shopping can be continued. Employee: This module for employees includes employee details who works in the organization. Every employee's details have stored in the database. The workers execute the customer's information and orders. Admin: The admin module contains user details and entry of categories, sub-category entry by the product master. The entire catalogue of products and their details can be stored. Reliability and Authentication This module provides customers and employees with access privileges for different operations. Safety is very significant for online transactions to be carried out precisely without hacking. For implementing the system using crowdsourcing, a class containing 100 students were selected. Out of these 100 students 4 groups were formed, since there are four modules in this system. And each group consists of 25 members, termed as a crowd.

After the requirements elicitation phase, the implementation of customer module was done by diving it into 2 parts.

1. Front end/User interface
2. Coding Section

With the help of the requirements gathered, crowd with 25 members started their work. The crowd was again divided into smaller groups for the successful outcome. Once the grouping was done, the first group designed the user interface. Since each smaller group consists of 5 members, each of these five members generated or designed the user interface individually and the best one was selected by the next group with five members who had the responsibility of coding. With the best outcome from the previous phase they started coding. With the help of the designed user interface the customer module was divided into different functions such as customer registration function, function for managing product details etc. Corresponding to the designed user interface the next group completed a particular function, it was then passed on to the next group in the same crowd, after testing, they chose the best outcome from the previous group for the same function and then they generated the next function, the process continued until the entire functions of the customer module was generated. After the integration of all functions it was passed on to the feedback or review phase. Here testing and error fixing was done successfully. The completion or designing of other modules were done parallelly, and all the modules from different crowd were integrated, tested and the error fixing was done. Since all the modules were developed simultaneously we were able to develop the system was in a shorter period of time.

IX. BENEFITS OF CROWDSOURCING

Topcoder is one of the platforms based on a model for the development of crowdsourcing, in which competitive development takes place. It's a platform in which software is produced via online competitions.

A. Merits with respect to schedule

It is important to develop and deliver the software on time. Today, developers that are competitive and flexible enough to work in different time zones and have the ability to perform multiple tasks are in great demand.

B. Merits with respect to cost

The overall cost of development is reduced by crowdsourcing software development. The overhead costs for traditional software development involve significant costs and small costs. The main costs include costs of purchasing hardware and software, transport costs, hiring and training software engineers.

C. Merits with respect to quality

Crowdsourcing software development would lead to more innovative solutions than traditional software development solutions in which team expertise is fixed.

The reason why the development of crowdsourcing software leads to the production of quality software is due to its wider participation.

X. CONCLUSION

Compared to conventional software development methods, CSE has different possible dominance. Crowdsourcing can help software development organizations merge elastic, external human resources to minimize cost from internal employment, and exploit the distributed production model to accelerate the development procedure. In this paper, a model for the development of Agile Software using crowdsourcing have been devised. This is especially done in order to conduct the SDLC phases through crowdsourcing workers. The benefits of agile approach and crowdsourcing are combined to obtain an better quality software with minimal defects. From our study, we arrived at the conclusion that the cost included and the man power required in this approach is least than any other development cycles. Since the software became an inevitable part of human life. It is an eminent option to adapt into agile crowdsourcing model for the effective results. By implementing the Online Shopping Management System with the agile crowdsourcing methodology we obtained the helpful software within short time. This study is limited to a particular Online Shopping Management System; more projects can be spearheaded for rectifying the drawbacks associated with this work.

REFERENCES

1. Himanshu Pandey, Dr. Santosh Kumar, Waseem Ahmad, Dr. V.K Singh, "Proposed methodology for crowdsourcing and agile development", International Journal of Advanced Research in Engineering and Technology (IJARET) Volume 9, Issue 2, March – April 2018, pp. 68–76.
2. Fábio R. Assis Neto, Celso A.S. Santos, "Understanding crowdsourcing projects: a systematic review of tendencies, work flow, and quality management", Information Processing And Management 54(2018) 490-506.
3. Nitasha Hasteer, Noshiba Nazir, Abhay Bansal, B.K Murthy, "Crowdsourcing software development: many benefits many concerns.", International Conference on Information Security & Privacy(ICISP2015), 11-12 December 2015.
4. Fiza Siyal, Xavier Franch, "A Goal Model for Crowdsourced Software Engineering" (2018).
5. [5]Bin Lin, "Crowdsourced Software Development and Maintenance", 2018 ACM/IEEE 40th International Conference on Software Engineering: Companion Proceedings.
6. Ke Mao, Licia Capra, Mark Harman, Yue Jia, "A survey of the use of crowdsourcing in software engineering", The journal of systems and software ,2017.
7. Klaas-Jan Stol, Brian Fitzgerald, "Researching Crowdsourcing Software Development:Perspectives and Concerns", CSI-SE'14, June 2, 2014, Hyderabad, India.
8. Asiegbu Baldwin C, Oluigbo Ikenna V, Ajakwe Simeon O, Onyike Gerald O, "Crowdsourcing Software Development: Concept, Benefits and Adoption", International Journal of Scientific Research in computer Science and Engineering, June (2017).
9. Chuanyi Li, Liguo Huang, Jidong Gea, Bin Luoa , Vincent Ng, "Automatically classifying user requests in crowdsourcing requirements engineering", The Journal of Systems and Software 138 (2018).
10. Ang Liu, Stephen C.-Y. Lu, "A crowdsourcing design framework for concept generation", CIRP Annals - Manufacturing Technology 65 (2016).
11. David Gefena, Gavriel Gefena , Erran Carmel, "How project description length and expected duration affect bidding and project success in crowdsourcing software development", The Journal of Systems and Software 116 (2016).
12. Anders Bruun, Jan Stage," New approaches to usability evaluation in software development: Barefoot and crowdsourcing", The Journal of Systems and Software 105 (2015).
13. Debra Zahaya, Nick Hajlib , Debika Sihic, "Managerial perspectives on crowdsourcing in the new product development process", Industrial Marketing Management 71 (2018).
14. Oguz A. Acar, "Motivations and solution appropriateness in crowdsourcing challenges for innovation", Research Policy.
15. Rimantas Gatautisa, Elena Vitkauskaitea, "Crowdsourcing application in marketing activities", Procedia - Social and Behavioral Sciences 110 (2014).

AUTHORS PROFILE

ANN JOSEPH, PG Student, Master of Computer Applications, Department of Computer Applications, Amrita Vishwa Vidyapeetham School of Arts and Sciences, Brahmasthanam, Edappally North P.O Kochi 682024, Kerala.

RESHMA SEBASTIAN, PG Student, Master of Computer Applications, Department of Computer Applications, Amrita Vishwa Vidyapeetham School of Arts and Sciences, Brahmasthanam, Edappally North P.O Kochi 682024, Kerala.

M SOUMYA KRISHNAN, working as Assistant Professor in CS and IT Dept., Amrita Vishwa Vidyapeetham, Kochi Campus, has over 16 years of experience in teaching. She has done MCA (B'level), M.Sc. CS and M.Phil. in Computer Science. She is currently doing her PhD in Computer Science. Her areas of interests are Data Analytics and Business Intelligence, Data Mining, Software Engineering, Software Development Privacy and ITES.