

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

ANATOMIZE THE SUITABILITY OF CROWDSOURCING IN AGILE SOFTWARE DEVELOPMENT

Himanshu Pandey^{*1}, Dr. Santosh Kumar² & Dr. V.K Singh³

^{*1}Research Scholar, MUIT, Lucknow

²Associate Professor MUIT, Lucknow

³Professor, BBDNIIT, Lucknow

ABSTRACT

The intend of the present study is to uniformly examine the crowdsourcing application, derive the behavior which have been referenced, and applied these act to analyze the suitability in agile software development. Agile software crowdsourcing is egress domain of software engineering. To determine the suitability of crowdsourcing in agile software development, we examined the allied literary works on crowdsourcing structure along with related case study material and draw out the characteristics which they allude to as portion of development of agile software. Jeff Howe (2006) described the Crowdsourcing, is the act of taking a task traditionally performed by a designated agent (such as an employee or a contractor) and outsourcing it by making an open call to an undefined but large group of people.

Although the intention in this paper substantially contour a research agenda, preliminary facts and data show that concept of crowdsourcing can support agile software development with high quality performance.

Keywords: Crowdsourcing, Decomposition, Intellectual Property.

I. INTRODUCTION

In Crowdsourcing research perspective, tends to concentrate on one of three aspects: the supplier aspect, the customer aspect and the system aspect (crowdsourcing platform).is the practice of obtaining ideas, services, or content by soliciting contributions from a large group of people, or more specifically, an online community rather than from traditional employees or service providers. Crowdsourcing is a quickly growing industry, not only for graphic design but also numerous other servicesA study of crowdsourcing provides a number of useful benefits have been linked to the use of crowdsourcing in agile software development.

1. Deduction of cost-In the software development community, crowdsourcing is usually consumed as a cost saving manner to collect labels for training data. Within Crowd, unlabeled instances are outsourced to a large group of people due to this circumvention of the extra cost burden commonly incurred in hiring software developers.

2. Reduction of actual development time: In agile software development process, crowdsourcing reduces the time taken by the development phase. Parallel development of decomposed task decreases the actual time of software development. Flexibility in working time (technical talent who can achieve follow-the-sun development across time zones and who typically willing to work at weekends) plays a major role for faster delivery of the software product.

3. Broad participation means higher quality: Crowdsourcing provides the capability to get entrance to a wide and deep pool development professionals and talents. They select the task on the basis that they have the expertness and participate on those contests where the winner solution is chosen for the development process or any problem.

4. Open Innovation and creativity:

Availability of variety of professional and expertise, guarantee that more innovative solution, clarification, explanation can be discovered, which often avoid the fixed mindset that can be lie within particular organizations or companies. This concept is basically known as “Near field repurposing of knowledge”.

II. LITERATURE SURVEY

Based on the literature review, we categorized numerous work related to crowdsourcing which are of great importance in rapid and flawless software development process. The paper by Thierry Buecheler, Jan HerikSeig,

Rudolf M. Fuchslin and Rolf Pfeifer [1], clearly explain how nonprofit “Research Value Chain” can definitely enjoy advantages of Crowdsourcing. Research analysis was given which explains a) how crowdsourcing can be applied to fundamental science and b) the effects of results of Artificial Intelligence research on vigorousness of Crowdsourcing. Conclusions and results from different research series will be merged, for instance, Complex or Versatile networks. The research and works of Eddy Maddalena, Vincenzo Della Mea, Stefano Mizzaro[2] , enlightens the possibilities that if a task proposed on crowdsourcing platforms is appropriate two mobile devices. Our prior motive is to analyze that(i) which of the preexisting crowdsourcing platforms are more compatible to mobile devices, and (ii) what sort of the application of Crowds explain the use of crowdsourcing in educational activities. In this paper[3], “The application of Crowdsourcing in educational activities”, explain the use of Crowd Sourcing in educational activities. With the growing time, the Boom in Information and Communication Technologies via Internet and duration a wide range of options for these organizations to accomplish their objectives. Hence the soul motive of this paper is to give up brief explanation of how education organizations have opted crowdsourcing as part of their activities in the current scenario time to explain how use of outsourcing would be soon spread to other educational activities with the passing time. Andre VnHoek's [4], “Crowdsourcing in Software Engineering, Models, Motivations and Challenges” is about software engineering, the crowd, and the possibility that if these advantages can be included in a software or not. Till date, these imaginative examples are not yet practically applicable but crowdsourcing has definitely started developing a place in software development field. The paper [5],“Reactive Crowdsourcing”, by Alessandro Bozzon, Macro Brambilla, StafanoCeri, Andrea Mauri proposers and approach to Crowd Sourcing which provides find level, powerful and ex-controls. We initially design is crowdsourcing application as a blend of elementary task types. Further, we convert these hi-tech specifications into the features of a reactive execution environment which supports planning, execution and termination of a task and performer analysis as well. Controls are fixed as active rules in data structures which are obtained from the design of application. These rules can be removed, had it or altered as per requirement which ensures highest efficiency with the lowest efforts. The paper by Kathryn T. Stolee, Sebastian Elabaum[6], investigate the use of crowdsourcing as a process to address that challenge by helping in subject recruitment. Further, to be more precise, with the help of his work we can express how study can be performed under infrastructure that ensures the possibilities to approach a large base of uses and also allows managing those users while the study is being carried out. We discuss the observations and results of this experience which explains the capabilities and potential of crowdsourcing software engineering studies. RajanVaish[7], has introduced a research direction which illustrates the possibilities of expert outsourcing by Linking mentor with student crowd. This process will allow mentors to comprehensively used operators such as split commerce, remove or add on project ideas, cord on studying to daddy suit on project ideas, code or students to carry out research. The research procedure will include numerous stages like brainstorming, paper-pencil prototyping, and development and user evaluation to produce publishable results. Encouraged by prior pilot experiment findings, my doctoral research examines the possibility of crowdsourcing the research process using operators along the research stage, while solving resource and opportunity constraints among mentor and crowd. We are inspired by the work of Bernardo A. Humberman[8], research of a huge data set from YouTube that the efficiency exhibited in crowdsourcing provides a strong positive dependence on attention which is calculated by number of downloads. On the other hand comma unavailability of a tension leads to decline in quantity of videos uploaded that further leads to decline in productivity, weights in some instances leads to know uploads at all. Moreover part-time contributors compare their performance to that of an average contributor whereas the long-term contributors compared to their own media. Research paper by Hanning Yuan, Yanni Han, Jun Hu[9], proposes a nimble development procedure of service-oriented software for a research perspective. Primarily, the history of software development procedure is viewed while focusing on the controlling forces of software technology. Furthermore, we observed the software services efficiency, increased by the user-driven needs in network environment. Then the nimble development procedure is proposed based on users' individual requirements and priorities comma which is a “meet in middle” way of software development. As it is controlled by users' personalized needs, this method can be presented as a blend of service resources on demand and approach to the target slowly. Hence, it turns out to be an efficient way to meet rapidly increasing application software requirements. Tom Narock and Pascal Hitzler[10], suppose that implementing search algorithms with crowdsourcing can prove to be a reliable solution. To be precise, they observe Big Data within the geosciences and describe great questions regarding the merger of crowdsourcing and semantics. They present the work that is being carried out in this area and discuss directions for future research. De created a Crowd Sourcing portal that allows people of the jio science

community to link their conference presentations and funded grant descriptions to the Database used in those projects. Input from the user is converted into RDF and these links are plotted in subsequent data Discovery tools. The links needed in the process are difficult to generate automatically due to a limited and specific information in the available data sets (e.g. no reference to data set used on inconsistencies in the researcher's name along the datasets). Although, contradictory to the most "Crowd" is comprised of professional researches and not the ordinarily lot. This gives birth to new challenges, in stimulating the crowd, and arousing trust. RashmiPopli, Nareshchauhan[11], cast the spotlight on the research work in Agile Software Development and estimation in Agile. They recommended a technique for the accurate cost and effort estimation, to escape from the problems of current agile practices. Based on the concepts of adaptability and flexibility, the Agile methods, represent an emerging set of software, are currently used as a gimmick to these reoccurring problems and make but a clear way for the future of development.

EmalAltameem[12], explains several methods , in which Agile methodology has been, proved to be influential in software development. It also elaborates the advantages and the restrictions of Agile Technique. This paper motivates developers, to adopt this technique, so as to develop software that proves out to be a remedy for their changing needs. Gaurav Kumar, Pradeep Kumar Bhatia [13], recognized the significance of this methodology, in terms of its quality within the cultural framework. KudaNageswaraRao, G.KavitaNaidu,PraneethChakka[14] paper, A study of the Agile Software Development Methods, Applicability and Implications in Industry, has been executed with the factual intentions of examine and gain acuity. Into, the latest agile concepts and Techniques, distinguishes between the strengths and weaknesses of agile methods and various issues regarding their applicability. WenjunWu,Wei-Tek. Tsai [15], examine the data in detail, gathered on software Crowdsourcing and abbreviates major lessons,learned, then analyze two Software Crowdsourcing processes, including artistic ways. PreetiRai, SaruDhir[16],elucidate the influence and comparison of several traditional techniques and a new methodology. Top Coder and App Stori processes.Concludingly, they identify the min-max nature among participants as a crucial element of design in software Crowd-Sourcing for software quality and investigate the reasons for which software industries, shifted from Traditional RE to Agile RE. Gaurav Kumar, Pradeep Kumar Bhatia [17], recognize the fact agile methodology has a remarkable impact over software development processes with respect to quality, within the organizational, methodical and cultural framework. Kiran Jamaalam adakal, V. Rama Krishna [18], emphasize on few confrontations with Agile->scrum and gives vision to the user whether the Agile is WONDER DRUG. Malik Hneif, Siew Hock Ow[19], showcase their review over three Agile approaches including Extreme programming, Agile Modeling, and scrum distinguishes between them and advice, when to use them. Gurleen Singh, Tamanna[20], reviews various agile techniques like on their characteristics, objectives, boons and banes of using agile methodology and their unique characteristics.

III. KEY CONCERNS REGARDING USE OF CROWDSOURCING IN AGILE SOFTWARE DEVELOPMENT

There is no definite boundaries defined that can cover all concerns regarding use of crowdsourcing in agile software development. But some general concerns can be drawn as we study crowdsourcing in terms of agile software development considering the general literature of crowdsourcing we synthesized a set of six concerns which have relevance in agile software development context:

- Task Partitioning or Decomposition Problem,
- Coordination & Communication gap between the crowd and organization,
- Planning, Scheduling and Execution
- Quality Concerns,
- Knowledge and Intellectual Property,
- Motivation and Remuneration,

(1)Task Partitioning or Decomposition Problem-

The base of crowdsourcing is that a whole job is divided or partitioned into smaller sub tasks and then distributed as an individual task to the crowd. An issue arises in this context and Herbsleb and Grinter reminded us of a definition of a module as: “*a responsibility assignment rather than a subprogram*”. The issue that is of major concern is that

the developers developing a software under agile development knows well how their code fits and functions into the resulting software being developed as they have already made certain assumptions and worked on the interfacing of the modules.

Whereas in general context of crowdsourcing we know that the tasks are divided in small modules which are generally independent. But when talking in terms of agile software development we now that the task being developed is complex and interdependencies exist between modules to a greater extent as compared to general tasks. So a primary concern arises is to break down the task into smaller sub tasks, which are independent and can be effectively crowdsourced. Kulkarni described this as “the workflow design problem”, thus we can conclude the degree of success can be achieved when tasks are crowdsourced efficiently and more efficient and independent decomposition leads to higher extent of parallelism.

(2). Coordination & Communication gap between the organization and crowd-

Success of any task depends upon how coordinated and well planned are its team members. So with reference to agile software development there too arises a need of coordination for proper flow of execution.

Malone and Crowston defined coordination as “*the process of managing dependencies among activities*”. In accordance with management, definition of coordination means directing efforts of individuals towards a common achievable goal and linking different aspects of an organization to achieve a common set of tasks.

The above discussed terms of coordination assumes that the task is carried out within an organization and no outsider is involved. But in agile software development term there is a continuous contact with the client so as to development and changes the specifications according to the interest of the client. But in crowdsourcing the crowd who is submitting the ‘solution’ is not a part of the crowdsourcing organization and in fact the task that have some interdependencies among them are performed by different workers, who doesn’t even now about the existence of the other one and this in turn can cause incompatibilities when integrating the whole software as a single unit.

The more the people involved, the higher is the communication overhead. This whether applies in crowdsourcing context depends upon the factors that whether the work is done in a collaborative manner or in a competitive manner.

(3). Planning, Scheduling and Execution-

With crowdsourcing, one or more task are given out to an unknown workforce for completing and execution and as a result the organization is at some extent letting go the control over that particular task. In agile development environment, the advantage is that the task can be timely delivered and reviewed as the work are completed in parallel to each other and independently.

On the other hand, however, this introduces a level of uncertainty as to whether or not the work will be completed on time. As in an organization the employees have deadline pressure and work pressure of authorities above them which makes them to complete it within the allotted timeframe. But in crowdsourcing uncertainty arises that the people from the crowd who took up the job will complete it on time or not. One of the promises of crowdsourcing is to shorten the product development cycle. In order to achieve this, it is important that the desired schedule of a crowdsourcing organization can be adhered to by the crowd. For instance, a core challenge is to ensure that sufficient workers are available when needed. While there may be expertise within the crowd, very specific domain knowledge that is required may not always be available at the moment it is needed. Furthermore, it is to be checked that sufficient time should be available to the coders to complete the

task within the time, relating the issue of planning to the size and scope of a task.. Lakhani reported that TopCoder “*community members worked best when contests lasted less than two weeks*”. Too large or long-lasting projects could result in decreased interest from the community, and thus fewer submissions.

(4). Quality Concerns-

Claims have been made by the crowdsourcing advocates that when a task is crowdsourced, its solution's quality is high as and every submission's quality is high. But at the same time there is probability of presence of noise in the submissions, where solutions are of low quality. In agile development crowdsourcing context, the idea of solutions from a large wide variety of developers helps in detecting and fixing defects and anomalies. It can be better understood by the explanation of the *"given enough eyeballs, all bugs are shallow"*. This gives an idea that the crowd comprises of several expertise and sometimes even the amateurs can spot something that is missed out by several expertise. This gives efficiency in problem solving but the main challenge is attracting sufficient contestants, under the assumption that given enough contestants, the required expertise will be present. Transparency plays a well-defined role in quality assurance and improvement. Like sites like TopCoder is fully transparent where prior to participating, contestants must register for a certain competition. However, that the greater the number of contest participants, the lower the quality of the work. One characteristic sometimes ascribed to the crowd is that it consists mostly of amateurs, thus suggesting that the resulting quality of output may not be on par with professional work. But Brabham pointed out this as a myth. Quality assurance is a key concern in software development, whether the software is developed in-house or by external parties and the customer has no knowledge of the developer or the methodologies they follow. There can be disagreement on the solution which comes from crowd. While the software either fulfills a set of requirements or not, disagreements may still arise regarding certain functionality or the scope of a task.. Furthermore, quality attributes of submissions, such as performance and maintainability of the code may still vary. One approach to quality control is peer-review. At TopCoder, members of the community perform peer-reviews of the submitted solutions. Similar to peer-reviews in open source, such reviews are *"truly independent"* given that the peer-reviewers would usually not know the creator of the work, and would therefore be unlikely to be either positively or negatively biased.

(5) Knowledge and Intellectual Property-

One type of knowledge of particular concern in crowdsourcing software development tasks is that of knowledge and intellectual property (IP). Confidential information leakage and consequent loss of advantage in competitive environment is a primary concern in adopting crowdsourcing in agile software development. Organizations may hesitate to share every specification and too many detail about a certain task that is to be crowdsourced because of the security and privacy clauses of the project. But sufficient detail in the specification is necessary for developers in the crowd to understand what the crowdsourcing organization is requesting and what they will be working on. Another issue that prevails is the ownership and copyrights as tasks on general are highly creative and in course of time requires new methodologies and inventions for efficient implementations and development. Thus organizations want to ensure that they can claim and patent any potential inventions that emerge during crowdsourcing and with no confusion over ownership.

(6) Motivation and Remuneration-

Motivation has gained considerable attention in field of software development and plays a vital role in success of a project. The motivations can be intrinsic or extrinsic where extrinsic is the conditions and surrounding (physical factors and environment) and intrinsic can be the job itself like gaining recognition and sense of achievement. The compensation or emolument of a crowdsourced task is dependent of the complexity and duration of the task. Task can be simple such as simple coding competitions and complex such as coding competition which span over 4 weeks. However crowdsourcing reduces the cost to a greater extent as for tasks for which professional were being hired is now being completed by expertise from the crowd at a very economical price. But, a problem arises is determining an appropriate price for a particular task is a key challenge for crowdsourcing in general and also for software development specifically.

IV. FUTURE SCOPE

Over the last decade, crowdsourcing has been emerging as a satisfactory practice for use by organizations wishing to source new ideas and solutions to micro-tasks from the Crowd. In parallel with the spreading of this practice among

innovative organizations, scholars have been increasingly keen to investigate crowdsourcing, thus it has been a popular topic among the scholars.

With the growing scope of Crowdsourcing we can conclude that in terms of agile software development in future the applicability will extend its hand in these dimensions:

- Crowdsourcing processes will be segmented i.e., for each subtask, professional organizations will evolve and occupy the market which may or may not create a monopoly.
- Crowdsourcing will be applied to different fields of software development and specification will be strict and bounded. The input and output of the crowdsourced task will be strictly defined and constrained which in turn will result in overall quality improvement of the output.
- More and more developers will be attracted towards this method as it is not bounded by any factors like deadline pressure and will enable them to work as per their interest and emerge with quality solutions.

REFERENCES

1. Thierry Buecheler, Jan HenrikSieg, Rudolf M. Fuchslinl and Rolf Pfeifer, “Crowdsourcing, Open Innovation and Collective Intelligence in the Scientific Method: A Research Agenda and Operational Framework”, Artificial Intelligence Laboratory, Department of Informatics, University of Zurich.
2. Vincenzo Della Mea, Eddy Maddalena, Stefano Mizzaro, “Crowdsourcing to Mobile Users: A Study of the Role of Platforms and Tasks”, DBCrowd 2013: First VLDB Workshop on Databases and Crowdsourcing.
3. Monika Skaržauskaitė “The application of crowd sourcing in educational activities”, ISSN 2029-7564 (online) SOCIALINĖS TECHNOLOGIJOS SOCIAL TECHNOLOGIES 2012, 2(1), p. 67–76.
4. André van der Hoek, “Crowdsourcing in Software Engineering Models, Motivations, and Challenges”, Focus: The Future of Software engineering.
5. Alessandro Bozzon. Marco Brambilla, Stefano Ceri, Andrea Mauri, “Reactive Crowdsourcing”, Dipartimento di Elettronica, Informazione e Bioingegneria – Politecnico di Milano Piazza Leonardo da Vinci, 32 – 20133 Milano, Italy.
6. Kathryn T. Stolee, Sebastian Elbaum, “Exploring the Use of Crowdsourcing to Support Empirical Studies in Software Engineering”, Department of Computer Science and Engineering University of Nebraska – Lincoln, NE, U.S.A.
7. RajanVaish, “Crowdsourcing the Research Process”, University of California at Santa Cruz1156 High Street, Santa Cruz, California 95064, USA
8. Bernardo A. Huberman, “Crowdsourcing, attention and productivity”, Social Computing Lab, HP Laboratories, Palo Alto, CA, USA.
9. HANNING YUAN, YANNI HAN, JUN HU, “RESEARCH ON AGILE DEVELOPMENT METHODOLOGY OF SERVICE-ORIENTED PERSONALIZED SOFTWARE” PUBLISHED BY IEEE XPLORE ,2008 INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND SOFTWARE ENGINEERING.
10. Tom Narock and Pascal Hitzler “Crowdsourcing Semantics for Big Data in Geoscience Applications”, Semantics for Big Data AAAI Technical Report FS-13-04.
11. RashmiPopli, NareshChauhan, “Cost and effort estimation in agile software development”, published by IEEE 2014 International Conference on Reliability Optimization and Information Technology (ICROIT).
12. EmanA.Altameem, “Impact of Agile Methodology on Software Development” published by Computer and Information Science; Vol. 8, No. 2; 2015 ISSN 1913-8989 E-ISSN 1913-8997 Published by Canadian Center of Science and Education 9.
13. Gaurav Kumar, Pradeep Kumar Bhatia, “Impact of Agile Methodology on Software Development Process” published by International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 4, August 2012.
14. KudaNageswaraRao, G. Kavita Naidu, PraneethChakka, “A Study of the Agile Software Development Methods, Applicability and Implications in Industry”, published by International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011.
15. Wenjun Wu, Wei-Tek Tsai, “Creative software crowdsourcing: from components and algorithm development to project concept formations”, published by Int. J. Creative Computing, Vol. 1, No. 1, 2013.

16. PreetiRai, SaruDhir, “Impact of Different Methodologies in Software Development Process” published by , International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014.
17. Gaurav Kumar, Pradeep Kumar Bhatia, “Impact of Agile Methodology on Software Development Process” published by, International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 4, August 2012.
18. Kiran Jammalamadaka1, V Rama Krishna, “Agile Software Development and Challenges” IJRET: International Journal of Research in Engineering and Technology. ISSN: 2319-1163, ISSN: 2321-7308 Volume: 02 Issue: 08 | Aug-2013.
19. Malik Hneif, Siew Hock Ow, “Review of Agile Methodologies in software development” International Journal of Research and Reviews in Applied Sciences ISSN: 2076-734X, EISSN: 2076-7366 Volume 1, Issue 1(October 2009).
20. Gurleen Singh, Tamanna , “An Agile Methodology Based Model for Software development”, published by, International Journal of Advanced Research in Computer Science and Software Engineering Research Paper, Volume 4, Issue 6, June 2014.