

**ENHANCING THE EFFICIENCY OF AGILE SOFTWARE
DEVELOPMENT USING CROWDSOURCING AND
VALIDATING THE REQUIREMENTS USING WEB
ONTOLOGY LANGUAGE**

Thesis

Submitted to

Maharishi University of Information Technology,

Lucknow, India.



For the degree of

Doctor of Philosophy

In

COMPUTER SCIENCE & ENGINEERING

By

Himanshu Pandey

Under the Supervision of

Dr. Santosh Kumar

**HEAD, Faculty of Computer Science
Maharishi University of Information Technology, IIM Road
Lucknow(U.P.) 226013, India**

YEAR, 2018



**Maharishi University of Information
Technology
Lucknow 226013, India**

Letter No.

Date

CERTIFICATE

This is to certify that **Mr. Himanshu Pandey** has completed the necessary academic term and the work presented by him is a faithful record of bonafide original work under the guidance and supervision of **Dr. Santosh Kumar, Associate Professor & Head, Faculty of Computer Science, Maharishi University of Information Technology, Lucknow, INDIA**. He has worked on **ENHANCING THE EFFICIENCY OF AGILE SOFTWARE DEVELOPMENT USING CROWDSOURCING AND VALIDATING THE REQUIREMENTS USING WEB ONTOLOGY LANGUAGE**. No part of this thesis has been submitted by the candidate for the award of any other degree or diploma in this or any other University around the globe.

Date: 12/10/2018

Place: Lucknow

(Dr. Santosh Kumar)

Associate Professor & Head
Faculty of Computer Science,
Maharishi University of Information technology
Lucknow, U.P., 226 013, (INDIA)



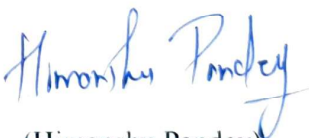
**Maharishi University of Information
Technology
Lucknow 226013, India**

CANDIDATE'S DECLARATION

I hereby declare that the research work embodied in this thesis entitled **ENHANCING THE EFFICIENCY OF AGILE SOFTWARE DEVELOPMENT USING CROWDSOURCING AND VALIDATING THE REQUIREMENTS USING WEB ONTOLOGY LANGUAGE** carried out by me under the guidance and supervision of **Dr. Santosh Kumar, Associate Professor & Head, Faculty of Computer Science, Maharishi University of Information Technology, Lucknow, India** is an original work and does not contain part of any work submitted for the award of any degree either in this University or any other University around the globe.

Date: 12/10/2018

Place: Lucknow


(Himanshu Pandey)

Research Scholar

CHAPTER 1

INTRODUCTION

1.1 Preamble

The research focuses on proposing an effective approach for raising the efficacy of agile software development by clubbing it with crowdsourcing. Ontological support to agile software development with crowdsourcing provides the necessary seeds for validating the agile requirements.

1.2 Need to Present Work

The major drawback in current agile development methodology is its inability to meet the growing number of requirements or change in the requirements during the analytical design of software systems. The inherent nature of agile software development methodology is to work in sprints, but when modification or addition of requirements take place at a greater pace than the agile development seems to fumble and it recursively increases with modification in the requirement. So we need a system that can leverage upon these requirement changes or addition of a team of developers, which can be handled the way we like. This problem can be best dealt with the process called crowdsourcing. Instead of taking the services of typically employed workers to perform the task of incorporating requirements and changing requirements in agile software development during most of the sprint process, the purpose is to take the services of the voluntary internet community, making the entire method time-efficient and cost-effective.

1.3 Objective of the Present Work

The sole purpose of this research is to change the existing paradigm of agile software development in such a way that organizational development runs contrary to the inherent expansion of the customer's precondition quantity. It is a major task to conquer the circumstance of intrinsic growth in requirements or modification amid the sprint procedure in agile software development. For that purpose, we proposed a solution by emulsifying crowdsourcing and agile software development. The thesis examines the nature of crowdsourcing and analyzes its compatibility with agile software development. The target of agile software development, crowdsourcing is to construct and execute quality software with next to zero cost required, by outfitting the intensity of the crowd. To meet this target, the crowdsource specialists in the crowd who consent to chip away at the assignment are given some budgetary or social motivating forces. The undertakings could be carried out in a cooperative or competitive way, taking into account the organizational type. Linux and Wikipedia are seen also known collaborative crowdsourcing illustrations, building up software through crowdsourcing breaks the boundary that exists between a software designer and a client, and takes after a co-creation standard.

Our work demonstrates step-by-step refinement of agile software development using crowdsourcing. We have also validated our model using the Protégé ontology tool, which is used to create and manage ontologies. Ontological support provides the necessary seeds for validating agile requirements.

We contrasted the conventional development of technology with crowdsourcing & agile crowdsourcing software development.

In contrast to the crowdsourcing model applicable to both methods, the distinction is made relying on both agile and conventional software development strategies.

In particular, the purpose of this research is to accomplish the following goals:

- Identify the challenges and drawbacks of agile software development.
- Peruse the methodologies of traditional software development crowdsourcing.
- Use crowdsourcing as a remedy to counteract the disadvantages of agile development of software.
- Examine the appropriateness of crowdsourcing in agile software development.
- Apply crowdsourcing rules in agile software engineering to enhance proficiency.
- Provide validation to crowdsourced results by utilizing ontological structure provided by Protege ontology tool.
- Explore the idea of agile software development for crowdsourcing and the opportunities for the software sector and other consumers of software.
- Identify the factors by making a differentiation of utilization of crowdsourcing in conventional software development and agile software development techniques based on adequacy.

Software engineering is imaginative and regularly advancing gives the strategy and technique to be pursued in the Software development and respond as a spine of computer science engineering methods. Software development is a systematic and very much arranged process that twists to give items in a quick, prevalent and reasonable way. Software has converted into numerous assorted fields, and is winding up more convoluted. Changing necessities from clients is making it much more troublesome.

Previous strategies to technology development are not entirely capable of meeting the changing and inspired requirements of the market. New software development approaches, such as agile software development methods, have been built along these lines, essentially to take care of such issues.

Many technology-driven companies have endured the impacts since the start of the current world money crisis-related emergency, being forced to lay off individuals or drastically reduce costs (Wauter, 2009). The company's own survival comes to an end with the opportunity to promote, deliver to the customer on time and reduce costs. Scientific research proliferates circumstances in which mission completion decides the success of the enterprise or, on the other hand, a plan makes the company bankrupt.. Consequently, reducing danger and heading in an organized manner towards projects wind up essential factors of achievement. In recent years, software development teams have heard about the effectiveness of agile methodologies like XP and SCRUM. The practical writing and business diaries show various examples of overcoming adversity with the benefits of partnerships that adopted agile methods successfully. As a result, there are probably several organizations trying to adopt agile methods.

Agile software development is somewhat, a propelled approach in software engineering. It can be said that agile processes or development techniques are a perfect approach to coordinating and managing software changes. Agile software development fluctuate from ordinary and set up approaches, as it features more on instrument for change administration than on the in advance designs, amid the undertaking. The establishing standards of agile software development depend on some current morals of programming improvement, inspite of being another approach, both from the area of software engineer-

ing, data framework and many more, for instance, generation administration. We have broken down the materialness of crowdsourcing in agile software development. We have pinpointed a few papers in which companies have utilized imaginative agile techniques for Software development under different circumstances use of enormous information and distributed computing have been broadly utilized in taking care of crowdsourcing. Customarily enormous information system has likewise been utilized for crowdsourcing improvement.

Crowdsourcing, as articulated by Jeff Howe (2006), is a method used by a delegated employee to take up a vocation and outsource it as an open call to an unknown, usually large gathering of people. We have audited different raise maturing of new advances making more communitarian corporate culture and concentrating on chance to separate the traditional approaches.

The proposed crowdsourcing methodology is the initial move towards contriving a procedure for step by step outlining crowd sourced application. Our initial phase in building up a comprehension of the crowdsourcing wonder was the investigation of a portion of the fundamental works in this area. This built up a general vocabulary and comprehension among the researchers about the idea of crowdsourcing.

Because of the particular subtleties of software development, in such a software framework, crowdsourcing are not the same as different situations, such as Amazon Mechanical Truck.

It may seem to have connections to what crowdsourcing aspects should be presented in the sense of software development. With all this in mind, in the illumination of a sophis-

ticated literature review, we defined some major points in the development of agile crowdsourcing technology.

However, as crowdsourcing requires various actors or stakeholders, considering the creation of crowdsourcing technology is helpful to take different stakeholder perspectives. They have defined three different views, namely that of consumers, practitioners and the platform talking to an online market.

1.4 Introduction to Agile Software Development

In the 1990s, agile development started ferociously as a denunciation of the framework with its somewhat stable and seemingly stupid development techniques to illustrate the waterfall model or v-model, generally referred to by names. These much more seasoned strategies attracted criticism for missing deadlines, going over budget or completely ineffective and agile offered a way to make this a legacy of past days. With most insurgencies the thoughts are once in a while new and frequently guarantee more than they can really convey, thus it has demonstrated with agile techniques. While agile approaches have incredible virtues, they are in no manner, shape or form a panacea for past aces and should only be obtained after sincere consideration and careful arrangement.

Agile is one kind of software development strategy. The main focus is on improving transportation customer satisfaction. The focus of Agile is more on restricting the scope of the project. An agile plan sets a specific number of preconditions and turns them into a commercially viable product.

An iterative and incremental (transformative) way to deal with software development which is performed in a very community oriented way, independent from anyone else arranging groups inside a viable administration system with "simply enough" service that produces astounding arrangements in a practical and convenient way which meets the changing needs of its Stakeholders. Agile software development is a myriad of different software development methodologies in terms of iterative and gradual growth, where necessary conditions and structures progress through a joint effort between pin-useful, identity-sorting organizations. In 2001, the agile manifesto was composed by the specialists, uncovers which things are viewed as significant by agile software development methodologies. As appeared in Table 1.1.

Table 1.1: Agile Manifesto

More Valuable Items		Less Valuable Items
Customer collaboration	Over	Contract negotiation
Responding to change		Following a plan
Individuals and Interactions		Processes and tools
Working software		Comprehensive Documentation

Agile technique for design provides opportunities to assess the path of a project throughout the entire development cycle of life. The agile strategy is described as "Iterative" and "Incremental" by concentrating on the reiteration of shorter work cycles and the functional things which they generate. Enhancement teams have only one opportunity in the waterfall system to get every aspect of a function correct. Agile project sets a basic amount of preconditions and makes them an implementable object. Agile means what it really

sounds like: fast and efficient, low cost reduction, less focus, shorter activities. Seventeen people made the manifesto for agile software development during February 2001 with the goal of seeking elective ways to deal with software development. Each of them had a significant effect on restricting common progression modes of programming, which they thought was inflexible, heavyweight, and unnecessarily quartered in documentation.

1.4.1 Advantage of Waterfall Model included in Agile Software Development

- Simple to implement. (Small sprint planning)
- Quality focused.
- After each stage, output is generated, so it has high visibility.
- Works well for smaller projects where there is a very good understanding of requirements
- The client and the project manager get the impression that substantial progress is being made.

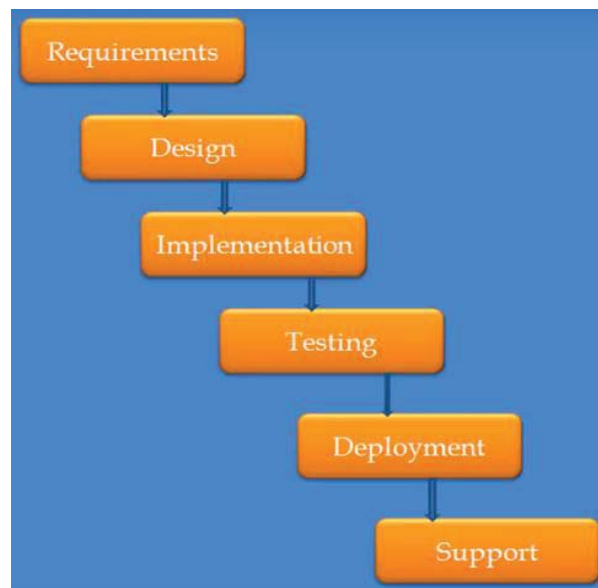


Figure 1.1: Waterfall model

1.4.2 Advantage of Prototype Model included in Agile software development

- Quicker user input for better solutions is available.
- An error can be detected much earlier.
- In the development, users are deeply involved.
- It is easy to identify the missing features.

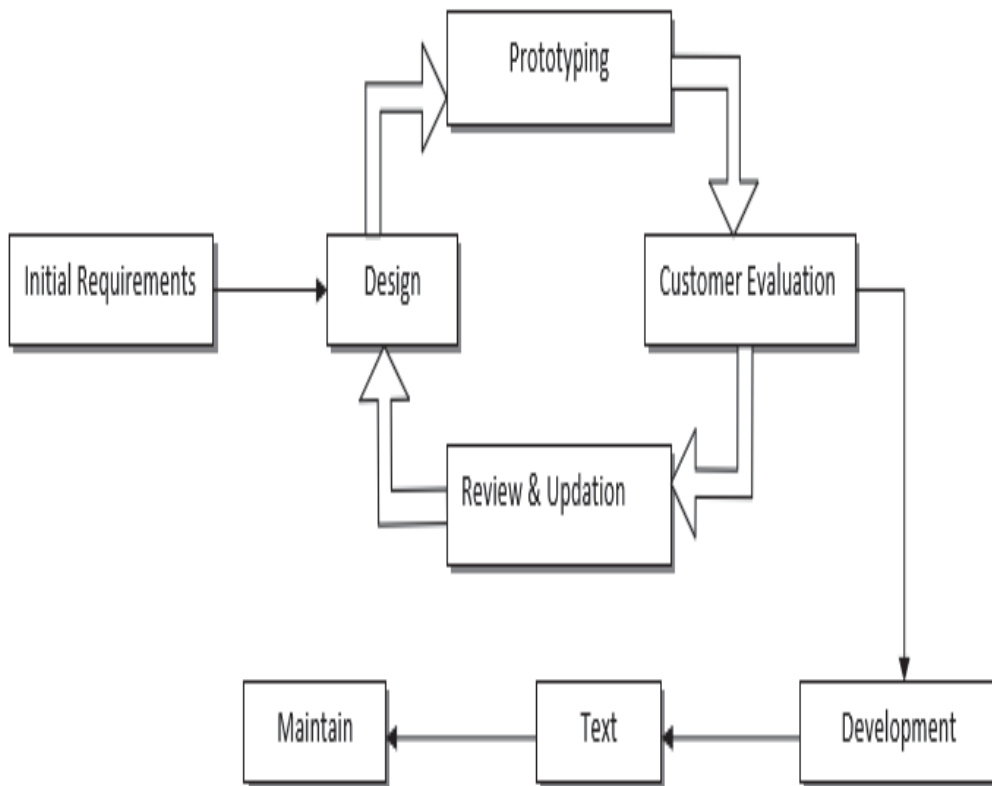


Fig-

ure 1.2: Prototype model

1.4.3 Advantages of V Model adapted in agile software development

- For security purpose Verification and Validation concept is taken from v model in agile development.
- Proactive defect tracking
- Avoids the downward flow of the defects.

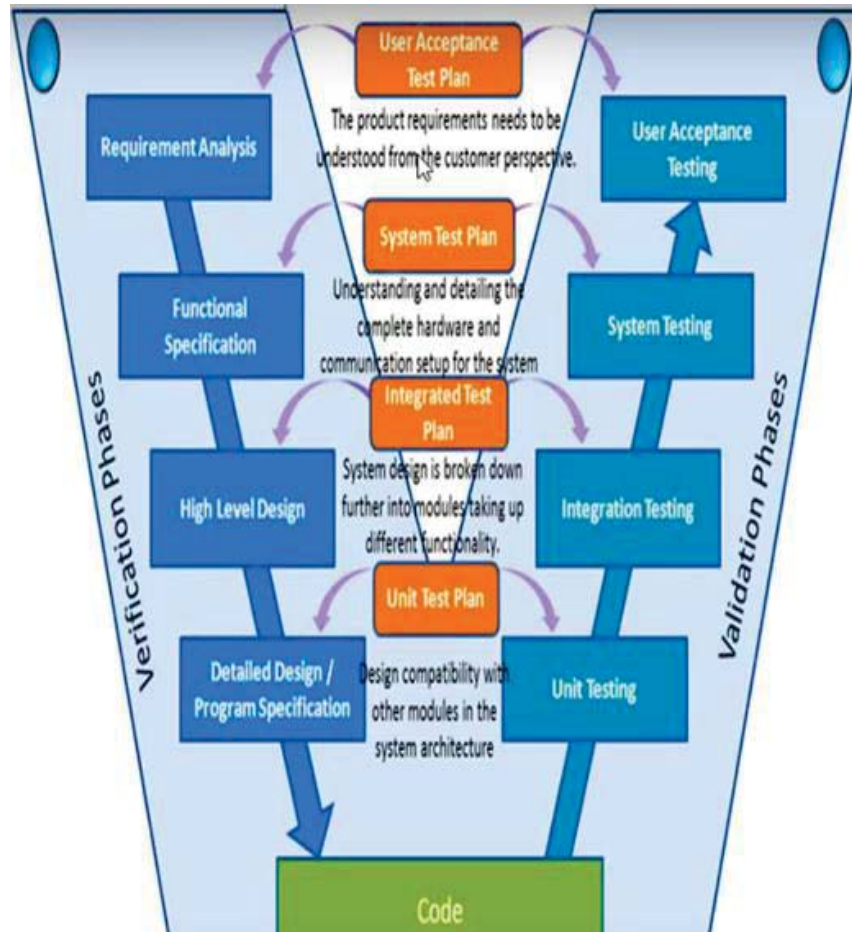


Figure 1.3: V- Model

1.4.4 Advantages of Spiral Model included in Agile Software Development

- Customers can see the work product at the beginning stages of the software development life cycle.
- Additional features or modifications can be rendered at a later stage.
- Regular or repetitive design helps to control risks.

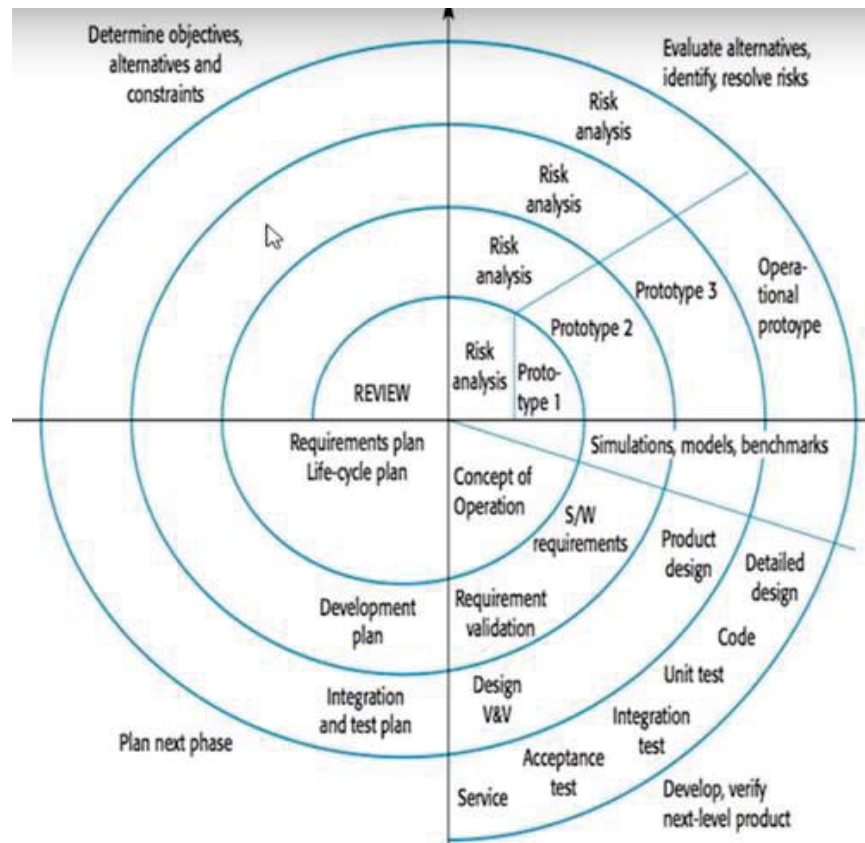


Figure 1.4: Spiral Model

1.4.5 Advantages of RAD Model included in Agile Software Development

- Changing requirements can be accommodated.
- Increases reusability of components.
- It is easier to transfer deliverables using high-level abstractions and intermediate codes as scripts.
- Use of generators of code and software for automation.
- Skilled team members.

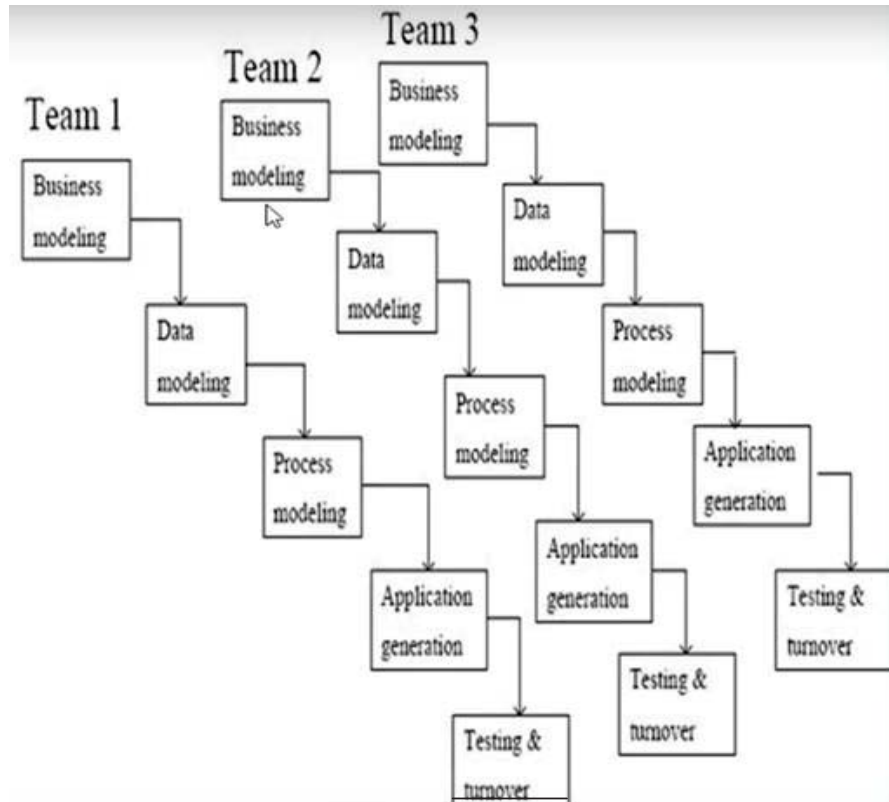


Figure 1.5: RAD Model

1.4.6 Working methodology of Agile Software Development

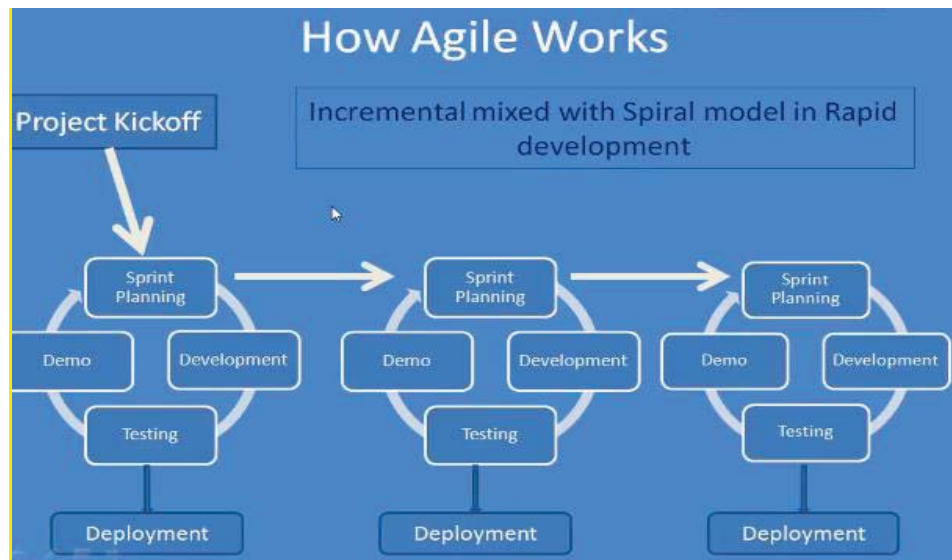


Figure 1.6: Functionality of Agile development process

1.5 Introduction to Crowdsourcing

In 2005, the word "crowdsourcing" was written by Jeff Howe and Mark Robinson [72], to describe how companies used the Internet to "outsource the work of the community". Howe [72], in his June 2006 blog post, published a description of the word crowdsourcing Wired article entitled "The Rise of Crowdsourcing". "Literally delineated, crowdsourcing represents the demonstration of an entity or agency that first executes a worker task and outsources it to an undefined, usually huge network of organizations as an open call.

This can take place as the method of peer development (whenever the project is conducted cooperatively), but on the other hand it is often followed by individuals. Crowdsourcing continues to serve as a framework for individuals or organizations to procure goods and projects. These resources provide opinions and funds from a wide community of web clients

that is generally open and constantly changing; to achieve a cumulative result, it isolates work among contestants.

The nascent area of software engineering is crowdsourcing code or crowdsourcing software development. Crowdsourcing software is an official request to participate in any target or project for software development that includes documentation, design, coding and testing.

Typically, either individuals from a customer campaign or individuals hired by the company guide these goals or tasks. But, in crowdsourcing systems, all tasks can be delegated to or preferred to be assigned by individuals from the general public. Individuals and organizations may also be interested in the complexities of crowdsourcing. Crowdsourcing is the process of working with a 'crowd' or team for a shared goal — often developing, learning to solve problems, or skill. New advances are driving this, social media and Internet 2.0. Crowdsourcing can take place through different companies at a wide range of levels and cross-sectionally.

It is now healthier for individuals to commit to an effort or cause individually, whether it be with emotions, time, skills, or money, because of our growing connectivity.

This deadpool will offer association access to additional ideas and strategies, deeper customer engagement, open doors to pre-creation, growth of business, and lower costs.

Internet and social media have conveyed links closer to their investors, laying the foundation for better plans to work together and putting trust together more than ever before.

"Crowds are a hit. A great many individuals, associated by the internet, are contributing thoughts and data to ventures of all shapes and sizes. Crowdsourcing, as it is called, is taking

care of precarious issues and giving restricted data. What's more, with the correct learning, adding to the crowd and utilizing its astuteness is less demanding than any time in recent memory."

1.6 Introduction to Ontology

Ontology can also be represented as an explicit configuration of conceptualization, i.e. the domain structure with probable limitations. Ontology, according to Guan [63], is the representation of domain system information in a particular modeling language. Ontology is an independent state of the model.

The term ontology was taken from philosophy. It can be defined as:

- A branch of metaphysics ' association with the being's life and experiences.
- A specific theory about the being's nature or form of life.
- There is no universal definition of the term ontology itself, but it is necessary to provide a clear definition of ontology. The definitions can be divided into approximately three categories, as in Knublauch [86].
- Ontology is a philosophical term and its interpretation is "existence theory".
- Ontology in Gruber (1993) is an explicit conceptualization specification.
- Ontology is a body of knowledge that defines a certain area of knowledge, usually common sense.

1.6.1 Ontology as specification of Conceptualization

According to Guan [63], conceptualization described intentional semantic structure encoding implicit domain knowledge and its specification is ontology.

Hence ontology is a logical theory that expresses conceptualization in some language. Ontology is a category of meaning and relationships that can exist for a group of agent perception. The ontological vocabulary is such that agents can communicate with ontologies and exchange information between these agents. Knowledge is based on conceptualization design, which is an abstract view of the problem domain along with various purposes. Each base of information, knowledge-based system or agent-based system is dedicated to conceptualization. The set of objects represented by the knowledge base is called the universe of discourse. The relationships among objects are reflected by representational vocabulary. In the context of artificial intelligence, we can describe the ontology by defining a set of representation. This ontology includes interpretations of the discourse universe entities (e.g. classes, relationships, functions, or other objects). Formal, it would seem that ontology is the logical theory's statement.

1.6.2 Operations on Ontologies

Multiple ontologies may be used in a single application. For these some ontological operations are needed, as given in Aziz [1], Akerman [5], and Bhatia [18]. These ontological operations are required for maintenances and integration of ontologies.

The operations are:

- ***Ontology Merge***

Through connecting the existing ones, it is creating a new ontology. The current ontology imports from the original ontologies selected expertise so that the result is consistent. The new merged ontology introduces from the original ontologies new concepts and relationships.

- ***Ontology Mapping***

It is the translation of one ontology into the another one i.e. translation between concepts and relations.

- ***Alignment***

It is a method of mapping in both ways, i.e. changing the original ontologies in order to have the correct translation. This allows new concepts and relationships to be added to ontologies.

- ***Refinement***

The archaic notions of ontology A may correspond to non-primitive (defined) concepts of ontology from ontology A to another ontology B.

- ***Unification***

It is the convergence of all the ontological concepts and relationships. It is possible to map the inference in one ontology to interact with the other ontology and vice versa.

- ***Integration***

It is developing new ontology from previous ontologies, which allows translation between previous ontologies and interoperability between systems that uses previous ontologies.

- ***Inheritance***

Ontology inherits all of the concepts, relationships and restrictions or axioms from ontology B, and there is inconsistency introduced by additional knowledge contained in new inherited ontology.

1.7 Introduction to Web Ontology language (OWL)

The Web ontology language (OWL) (World Wide Web Consortium recommendation), extends Resource Description Framework. Web ontology terminology is anticipated to be used by applications involving processing of the data content instead of simply presenting information to the people of Kosinar[88]. Constructing on the foundation of resource classification, the Web ontology language offers more machine-interpretable semantics by specifying additional vocabulary and formal semantics. Web ontology language is based on logic of explanation, which is a limitation of the logic of first order. Web ontology language provides three thread-languages that are progressively expressive: OWL Lite, OWL DL, and OWL Full. An extension of their simpler predecessor is each of these sublanguages. Web ontology language DL is often used as the ontology modeling language compared to the other two sublanguages due to its ability to articulate and infer equal semantics. Most of the web ontology

language reasoners available support the language of web ontology. Web ontology language typically consists of classes, properties, class instances, and instance interactions. Instances of classes in web ontology language are called individuals. Web ontology language classes are described by means of "class description" that can be combined into "class axioms".

1.8 Introduction to Crowdsourcing Methodology (CRM)

To develop the elementary methods of crowdsourcing methodology, the designer pick up the list of terms or ideas and sorts out them into tasks and solutions and generates an initial where tasks are the objectives or goals to accomplish and solutions are the bottom up manufacturers for the developed product. For instance, taking research journal paper submission process as the contextual investigation, foremost the requirements are obtained. At that point the procedure of analysis starts that consolidates interviews and selection of contestants. The contestants later are appointed to workers. A task hierarchy diagram including Minsky's Model [57], of task reduction. Afterward task reduction is completed, Roles are relegated and use cases are explained. Roles are additionally refined in that the particular task is apportioned to particular Role(s).The procedure of design and execution of CRS at that point continues eventually meeting to the last crowd sourced item.

CHAPTER 2

LITERATURE SURVEY

Based on the literature analysis, we categorized numerous work related to crowdsourcing which is of great importance in rapid and flawless software development process. Ab Aziz and Abdullah [1], examined primary design is to make ontology to speak to the learning that has been put away in the information store subsequent to catching the experience learning from the Development group in ASD. The ontology framework would benefit by presenting a relationship that will help the Community of Practice (CoP) to easily share the experience information later on. A writing survey of the basic things in the ASD been directed, we at that point making Ontology display by utilizing instruments called Protégé. Abbas [2], conducted research with the essential objective of taking RE activities that can be considered by agile methodology if reliability is the cardinal consideration. The investigation is designed for the study of the past research announced in the existing together writing and the practices which are being sought after in the region. Afuah and Tucci [3], represents crowdsourcing as a solution to distant search. Aitken and Ilango [4], performed a comprehensive analysis of conventional software engineering and agile software development in order to uncover the

basic similarities and any possible incongruities between these two ideal software development models. Akerman and Tyree [5], studied ontology to support the development of software architectures. They proposed a way to deal with software development that spotlights on engineering choices and includes the utilization of ontology. In this approach, the engineering is caught by an example of Ontology. The ontology has four noteworthy parts: design resources, engineering choices, partner concerns, and a design guide. Al-Zewairi et al [6], conducted an investigation of the studies of the distinctive deft procedures extending from January 2000 to December 2015 utilizing a natural research system called "Look at and Review" (CR). Besides, these student papers were ordered into four noteworthy classifications as indicated by their region of study. Also, the recently proposed agile methodologies that have not been tended to yet in some other writing survey were checked on and looked at regarding where the progressions that they proposed lay on the SDLC. Alam and Campbell [7], describes the role of relational mechanisms in crowdsourcing governance. Altameem [8], highlights the effect on software development of Agile Methodology. In the elicitation and software development stages of software engineering, Alvertis et al [9], conducted a study using crowdsourced and anonymized citizens. The paper proposes the use of a person that speaks to an arrangement of clients with comparable qualities, a pool of personas that Software groups may impart to each other through a shared application, and persona manufacturer as a device to produce such personas through genuine client profiles and information gathered through outsider administrations. Toward the end, a demo application is exhibited, understanding the idea of anonymized, crowdsourced personas. We inspired by the work of Amini [10], towards adaptable assessment of mobile applications through crowdsourcing and automation. Amrollahi et al [11], utilize crowdsourcing platforms to develop a transparent

learning strategy. Anderson [12], defines a framework for distributed learning in crowdsourcing higher education. Auler et al [13], conducted a study to propose gathering web customers execution information to fabricate a crowdsourced compiler flag recommendation framework in the cloud that enables the compiler to play out the suitable improvements for every customer stage. Since this data originates from crowdsourcing as opposed to manual examinations, pointless or hurtful advancements are naturally disposed of. Our approach depends on live estimations done while customers utilize the application on genuine stages, proposing another worldview on how improvements are tried. Babb et al [14], conducted a study to use the consequences of investigations to distinguish normal and new hindrances to learning. Regularly these obstructions to learning exist as indicated by authoritative culture and the degree to which that culture impacts states of mind, standards, and practices relating to learning. They display these boundaries to learning and give knowledge to the causes, impacts, and potential improvements for these hindrances. Bajnaid et al [15], identified an ontological framework to agile software development to support process-driven quality control. Difficulties associated with the part of quality assurance in agile projects appear to be created by developing a prescribed procedure that gives personalized resources to customer inquiries. The planned ontological model involves both measured and functional SQA knowledge of computer types and their prerequisites, including performance characteristics, SQA assessments, SQA measurements, and related SQA techniques and methodology. Bari et al [16], exhibited the status of mechanical software crowdsourcing and features the examination headings distinguished by these practices. Programming crowdsourcing includes individuals, and human are known to act in vulnerability way, it is difficult to accomplish repeatability and dependability for crowdsourcing ventures. In particular, an undertaking frequently needs to convey

administrations at a venture level of steadfastness, consistency, and effectiveness, and this requires a noteworthy administration and organization of a precise group network equipped for delivering the coveted outcomes reliably. Benoit et al [17], demonstrated that our approach works clearly with various sorts of political content, written in various dialects. While discoveries announced here concern content investigation, they have broad ramifications for master created information in the sociologies. Bhatia et al [18], reviewed about clarifying such openings while examining the part of ontologies as a Software Life-Cycle bolster innovation. The paper explores the down to earth and practical use of ontologies in the field of software engineering, which are taken up by the challenges and uncertainties that will keep the area active. Bider [19], conducted an analysis of agile software development from the Web Ontology Language edge transformation perspective. The model has been assembled with the light of the learning change viewpoint, as the creator views it as the most separating points of view when contrasting ASD with customary software development. Bishop [20], conducted a contextual analysis utilizing Big Data from a task called "QPress" that was controlled by an association that is based around unexpected working and between polished skill. Critical things drawn from the information gathered from the examination incorporate the significance of the Cloud to remove working, for example, teleworking; the personality of the association and how specialists identify with it; and also what factors help or restrain laborer inspiration. Bozzon et al [21], conducted a study to display each crowdsourcing application as the creation of basic errand writes and we continuously change these abnormal state specification into the highlights of a responsive execution condition that backings undertaking arranging, task, and consummation and in addition entertainer checking and avoidance. Brabham [22], analyzed crowdsourcing, both its theoretical set-up and prototype instances,

taking into account the acceptance of open source crowdsourcing. In particular, this research discusses the possible results for the template, its ability to abuse a horde of trailblazers, and its potential for previous use in benefit segments. Finally, this analysis suggests a guide for the crowdsourcing test. Breaux and Schaub [23], evaluated outcomes from two pilot thinks about and a third trial to legitimize are applying an assignment disintegration way to deal with prerequisites extraction. The last assessment demonstrates a 60% diminishment in the cost of manual extraction with a 16% expansion in extraction scope. Buchan et al [24], analyzed the desires for two partner gatherings (software development group, and software clients) about client association with a specific end goal to comprehend the desires and survey their arrangement. Buchan utilized RG instrument reflectively in this examination and set that it could likewise be connected from the beginning of a task, or proactively as asymptomatic apparatus all through an undertaking to evaluate and guarantee that desires are adjusted. Buecheler et al [25], analyzed how crowdsourcing could possibly benefit the non-benefit "Exploration Value Chain". Further, an examination plan is suggested that explores a) the pertinence of crowdsourcing to key science and b) the effect of dispersed agent's standards for artificial intelligence inquire about on the vigor of crowdsourcing. Bits of knowledge and strategies from various research fields will be joined, for example, complex systems, spatially inserted associating agents or swarms and dynamic systems. Bücheler and Sieg [26], discussed the usefulness of crowdsourcing and a few Open Innovation techniques to a non-benefit academic approach and basic science (e.g., a conventional university of research). Buettner [27], surveyed the crowdsourcing writing incorporated into top associate assessed diaries and gatherings, and develop a far-reaching picture. In view of this, he recognized experimental and configuration situated research needs. Burguera et al [28], studied for

benefit by preceding methodologies for complex analysis of need for conduct as techniques for detecting malware in the Android level. The framework was illustrated by breaking down the information collected on the focal database using two forms of information collections: artificial malware collected for testing. Challiol et al [29], displayed a novel way to deal with assigning some portion of the exertion in building versatile web software to engineers outside those associations or even to definite clients and demonstrated that this approach is possible, light and handy and present an arrangement of examinations created to confirm cases. Chanal and Caron [30], conducted a study to progressively exhibit the three fundamental periods of the vital investigation completed with the crowdspirit group: (i) elaboration of crowdspirit plan of action; (ii) esteem creation process identified with profiles of crowd spirit network of givers (iii) Theoretical structure on plans of action in view of crowdsourcing. Chatzimilioudis et al [31], presented Smartphones' inborn qualities, and in addition, a scientific categorization that characterizes the developing field of portable crowdsourcing. They likewise displayed three novel applications that streamline area based inquiry and comparability administrations in light of crowd-produced information. Chatzimilioudis and Zeinalipour [32], assessed that propelled class shows the key ideas driving crowdsourcing and its applications to versatile information administration. They reviewed the crowdsourcing research scene from an assortment of points of view, with a specific accentuation on the most recent information administration patterns and center around a top to the bottom scope of rising portable. Chen and Zhang [33], proposed to interface two sorts of documentation by catching engineers' Web perusing conduct with regards to record perusing and incorporating crowdsourced much of the time made inquiries (FAQs) into API archives. Chen and Luo [34], presented three business software items as educational testing projects, which are

crowdsourced by our teaching supportive network and call this "Semi Crowdsourcing Test" (QCT) in light of the fact that the workers are students, who have reliable social relations. The examination results are urging and show to be useful to both the students and industry in QCT projects. Chen and Liu [35], in both types of competitions, find a U-molded link between submission time and winning. Social capital increases the likelihood of winning a crowd-evaluated competition only if the social capital is high enough. Chilana et al [36], introduced LemonAid, another approach to managing help that empowers customers to find help by rather picking a name, gadget, connection, picture or other (UI) component that they accept is important to their concern. Choudhary et al [37], evaluated the after-effects of an experimental investigation directed at Microsoft to find out about Agile Development and its observation by individuals in the development, testing, and administration. Corneli [38], conducted a study to analyze two open internet learning networks, Peer 2 Peer University. Crowston et al [39], examined the performance of information systems in the implementation of free and open source applications. In this article, they rethink what achievement implies inside a FLOSS setting. The first audit existing models of IS achievement and achievement factors utilized as a part of FLOSS examine and evaluate them for their helpfulness, reasonableness and fit to the FLOSS setting. DeLoach and Garcia [40], pose the Methodology Framework for organizational multi-agent software engineering (O-MaSE), which synchronizes a set of concrete technologies to facilitate industrial affirmation. Lucia and Qusef [41], evaluated requirements engineering in agile software development and talks about issues worried about the conduction of prerequisites building exercises in agile software development forms and proposes a few changes to fathom a few difficulties caused by agile necessities designing practices in huge tasks. Nicola and Missikoff [42], presented a lightweight me-

thodology for rapid ontology engineering and proposes a basic, Agile Ontology building strategy planned to put end clients at the focal point of the procedure. Souza et al [43], analyzed agile and customary mix by defining a mixture procedure that exploits the two methodologies. Della et al [44], conducted an experiment to explain both (i) which of the existing crowdsourcing stages are more acceptable for cell phones and (ii) what types of errors are more suitable for cell phones. Despa [45], studied the effects on the current state of information in the philosophy of software development. It intends to set the pace for anticipating the formalization of an advancement-oriented IT software development technique. Devedzic [46], evaluated the involvement of instructing agile software development to understudies of software engineering, software designing, and other related teaches, and remarks on the ramifications of this and the exercises learned and briefly reviews varieties to these courses given somewhere else. Dingsøy et al [47], analyzed productions and references to illustrate how the exploration of agile has advanced in the 10 years following the verbalization of the statement. Specifically, they outline the reasonable structure basic agile grant by playing out an examination of creators who have made striking commitments to the field. Dingsøy and Lassenius [48], spotlighted specifically on two late patterns in examine on agile software development: First, the progress from an attention on agile strategies on group level with accentuation on group execution (outlined by the emphasis on match programming and test first development), to a more extensive hierarchical understanding where more spotlight is put on estimation of the created item. Second, the change from iterative development with beginning suggestions on multi-day emphasis in scrum to the persistent organization of new highlights. Doan et al [49], exhibited crowdsourcing frameworks on the web. They characterize and group such frameworks, at that point portray a wide example of frameworks and ex-

amined essential difficulties, for example, how to enroll and assess clients, and to consolidate their commitments. Dwarakanath [50], assessed a software development philosophy that tends to key difficulties for the use of crowdsourcing to an undertaking application Development. The strategy displays a component to deliberately break the general business application into little undertakings with the end goal that the assignments can be finished freely and in parallel by the group. Dyba and Dingsoyr [51], elucidated the adequacy of agile techniques, they audited the agile development writing and led an efficient investigation of what is known exactly about its advantages and confinements. Elshandidy and Mazen [52], contended that embracing ASD in RE defeats the constraints of the customary development and encourages the per user to (1) get a brisk yet a thorough handle of RE in customary and ASD; (2) comprehend the cooperative energies/shared characteristics between the two methodologies in dealing with RE; (3) perceived the related difficulties of embracing ASD; and (4) recognized the momentum unmistakable agile RE inquire about zones. Estellés and Ladrón [53], studied the meanings of crowdsourcing are investigated to separate normal components and to build up the fundamental attributes of any crowdsourcing activity. Fan et al [54], portrayed an ontology model of the plant's space learning utilizing related organic science information and utilizes Protégé as the apparatus for the plant area ontology depiction and stores the plant space ontology as web ontology language records. Fergis [55], analyzed the impact of an agile methodology on software development costs and investigates the effect of changing to an agile technique on such hazard and related expenses. Fitsilis [56], contrasted PMBOK and agile project management and studied a non-exclusive system of task management types as defined in the information project management body (PMBOK) with various agile enterprise management models. Geerts [57], examined vital discoveries that are important for both hy-

pothesis and practice. Geiger et al [58], conducted a study by arranging the procedures of 46 crowdsourcing illustrations, they distinguish 19 unmistakable process writes. A subsequent bunch examination exhibits general models among these sorts and demonstrates a connection with particular uses of crowd sourcing. Ghobadi and Mathiassen [59], alleged obstacles in agile technology teams to successfully knowledge sharing. They researched how venture supervisors, engineers, analyzers and client agents consider boundaries to viable information partaking in agile development. Golub and Jackson [60], contemplated learning in a setting where specialists get free loud flags about the genuine estimation of a variable and after that convey in a system. Gong et al [61], studied about the building and thinking of movement ontology in view of the techniques for ontology. First, it presented the proceeds of the working of movement ontology in detail, which incorporates characterizing its structure of classes, properties, connections, maxims and filling examples. Graber, M.A. and Graber A [62], focused on world wide web-based crowdsourcing and research morals and has shown that the crowdsourcing research paradigm has the potential to harm participants, manipulates participants into continued involvement, and uses people involved as test objects. Guan et al [63], observed an ontological way to deal with encouraging security information mapping from security prerequisites to their relating arrangements security designs. Happel et al [64], examined a few cases of ontology applications all through the Software Engineering Lifecycle. They talk about the upsides of ontologies for each situation and gave a structure for grouping the use of ontologies in software engineering. Hasteer et al [65], at a crowdsourced stage, investigated the software development process. In an active software development mechanism, the research examines and determines the stage astute expectations. He-ping et al [66], used relational database for research and implementation of ontology automatic construction. Heer

and Bostock [67], examined the appropriateness of mechanical turk from amazon as a level for graphical recognition research. We echo past research into the dynamics of spatial encoding and midtones and look at the development process of our experiments. Hellmann et al [68], analyzed how agile attempts are being used for; what kinds of paper appear to be circulated in this area; how do professionals and scholastics add to look into this field; and what methods are being used to perform agile testing? They demonstrate that scholastics and experts center around various kinds of distribution and, irritatingly, that the quantity of specialist papers in the sources we sought is emphatically down since 2010. Hetmank, and L. [69], carried out a randomized controlled trial of literature on crowdsourcing schemes. They often developed and introduced elements and functions in a crowdsourcing framework that are incorporated. Hirankitti and Mai [70], characterized a demo (.) predicate together with assistant sayings, is proposed and utilized for prevailing upon web ontology language2 ontologies communicated regarding meta-programs. Hneif and Ow [71], evaluated three agile methods, which include agile modeling, extreme programming, and SCRUM, describing their contrasts and prescribing when to use them. Horita et al [72], examined further use of voluntary geographic information (VGI) and crowdsourcing in disaster management their operates lead a widespread systematic review went to survey the ebb and flow status of research in the use of VGI as a source of data to assist in debacle management. Hu and Wu [73], embraced the hypothesis of static recreations with finish data and found that the estimation of effectively difficult likelihood can be utilized to conclude specific aggressive choices of coders in the calculation challenge organize. Hu et al [74], evaluated the software development field in outside nations. Particularly, Scrum will assist us in managing the task all the more productively in light of the fact that it is a versatile procedure. Huberman et al [75], investigated a

huge informational collection from YouTube that the efficiency displayed in crowdsourcing displays a solid positive reliance on consideration, estimated by the number of downloads. Hyland-Wood et al [76], conducted a study to depict progressing exploration to build up a philosophy for software upkeep utilizing Semantic web procedures. They proposed the gathering of software framework metadata which may incorporate utilitarian and non-practical necessities documentation, measurements, the achievement or disappointment of tests and the methods by which different segments associate or were expected to cooperate. Ionel [77], evaluated the primary hypothetical commitments to the field of agile methodologies investigate, by introducing agile methodologies, by dissecting the fundamental papers on social ramifications of utilizing agile, by showing the principle examines on the execution of agile and by orchestrating the examination with respect to correspondence in agile tasks. Ismail et al [78], promoted an efficient 'cross section based' group administration methodology to investigate the space of consents sets. Jammalamadaka and Krishna [79], examined a couple of difficulties with Agile-scrum and gives an understanding to the client whether the nimble is the silver bullet. Joel et al [80], depicted how the laborer populace has changed after some time, moving from a fundamentally direct salary, U.S based workforce towards an undeniably universal gathering with a huge populace of youthful, knowledge Indian specialists. Kåreborn et al [81], studied fieldwork among apple versatile application engineers in Sweden, the UK, and the US. The investigation demonstrates that, albeit a few engineers encounter achievements, financial returns stay tricky and many experience exceptional strains to create and showcase new items in a focused and soaked market. Kazai and MilicFrayling [82], studied the parts that could be utilized to devise such measures. The proposals depend on tries different things with gathering significance appraisals for digitized books, led as a

component of the INEX Book Track in 2008. Kazman and Chen [83], conducted a study on the metropolis model, gives a structure inside which associations can reason about all parts of how they make frameworks, including device bolster, dialects, preparing, asset distribution and administration, and individual inspiration. Kittur et al [84], led an investigation to break down the utility of a smaller scale task grandstand for social occasion customer estimations, and inspect plan considerations for making remote miniaturized scale customer evaluation. Kiran Hiwarkar [85], explored what procedure can be adjusted to a particular undertaking. Correlation between various agile software development techniques will help in the choice of fitting Development show given a specific situation. Knublauch, H. [86], researched Ontology-driven Web semantic software development: an illustration of a Protege / Web ontology language scenario. Depending on a realistic implementation scenario, the technology design and development strategy for network administrations and specialists for the semantic web was displayed. Knublauch et al [87], initiated an open development environment analysis for weekly web services. They portrayed the web ontology language plugin, the Protégé ontology development stage's semantic web overhaul. The web ontology language plugin can also be used to modify ontologies in the language of web ontology, to arrive at depiction reasoners, and to protect phenomena for semantic markup. Košinár [88], proposed a technique for design and adjustment of agile procedures behind human services items development in view of accumulated learning and formal displaying. Their approach permits to help and enhance the procedures with formal strategies for demonstrating and machine learning based recreations. Kosinski et al [89], demonstrated groups made out of laborers of high notoriety accomplish higher execution than low notoriety swarms, and the impact of the measure of the installment is non-monotone - both paying excessively and too little influences execution.

Kozinets et al [90], analyzed the structures and procedures of online aggregate customer development. Kumar and Bhatia [91], conducted a study to distinguish the effects that agile procedure has on software development forms as for quality inside the hierarchical, efficient, and social structure. Lane [92], spotlighted on (a) definitions and uses of aggregate knowledge identified with crowdsourcing and open advancement and (b) production of an upper hand. The ultimate result presents five proposals for how these two kinds of aggregate knowledge might be utilized to help advancement, including the need to make an aggregate insight (CI) framework and utilization of an open development plan of action. Lakhani et al [93], analyzes the applicability of crowdsourcing in software development. LaToza and Hoek [94], studied software building, the group, and whether progress like these can be had in software. In spite of the fact that basic cases are unadulterated fiction today, crowdsourcing obviously is as of now entering software development practice. Leicht et al [95], investigated the current assortment of information with respect to crowdsourcing in software development. Consequently, they recommended a major system with five measurements to structure the existing bits of crowdsourcing knowledge regarding software development and also to deduce an increased inquiry into an excavation plan to tackle. Leimeister et al [96], evaluates the activation-supporting components for the competition for IT-based concepts. Li [97], studied the effect of variables on the quality of code and identified six key elements, including the standard stage quality score, the number of contemporary projects, the duration of the section database, the number of engineers recruited, the highest level of engineers presented and the outline score. Lim and Ncube [98], proposed the utilization of interpersonal organizations and crowdsourcing to distinguish and organize the partners for the arrangement of frameworks ventures. The proposed technique crowdsources the partners to prescribe different

partners, manufactures an informal organization of partners, and organizes partners utilizing interpersonal organization measures. The paper depicts the strategy and examines the qualities and restrictions of applying the technique to the arrangement of frameworks ventures. Lindsjorn et al [99], examined performance of teamwork and software development project achievement. This article researches the impact of cooperation quality on group execution, learning and work fulfillment in agile software groups, and whether this impact varies from that of customary software groups. Franklin [100], focuses on crowdDB: answering queries with crowdsourcing. Mahmud and Aris [101], conducted a study to distinguish the potential categorization of the applications in a way that empowers the analysts to comprehend the ebb and flow circumstance and subsequently decide the heading of future research here. Malone et al [102], conducted a study to list the qualities of aggregate knowledge- the conceivable responses to these key inquiries - and indicate how blends of qualities involve a "genome" that portrays every aggregate insight framework. Mao et al [103], conducted an experiment about the use of crowdsourcing in a computer building, attempting to cover all the writing on this topic. Second, they critique the ideas of crowdsourcing with their conceptual classification and deduce the principle of software engineering crowdsourcing. Mao et al [104], done extensive research to address these issues by presenting 16 cost drivers for crowd-sourced development activities and evaluating 12 popular assessment models using 4 powerful Top-Coder implementation measures. The largest current software development crowdsourcing process and analyzed each of the 5,910 software development errors (for which incomplete data is accessible), using these to eliminate our suggested cost drivers. Martin et al [105], studied about contentions for network totally open source Software development and "big data" good answers for what's to come. For the interim, they have spread out ten best tips for

information handling. With these close by, a first vast scale proteomics examination ideally turns out to be less overwhelming to explore. Marwat et al [106], proposed ontology for software architectural outline in light of IEEE show for engineering depiction and Kruchten 4+1 model for perspectives characterization. The principal point of the proposed ontology is to arrange and find software architectural plan data. Mason and Watts [107], evaluated the impact of remuneration on execution with regards to two examinations, directed on Amazon's Mechanical Turk (AMT). Mehra et al [108], conducted a study on ontology creation by reference to a knowledge corpus. The strategy contains: looking through the corpus to distinguish records with content that matches a seed space portrayal; recognizing further archives inside the corpus that are semantically like the distinguished reports; distinguishing a subgraph of the classification pecking order that incorporates the categories relegated to the separated archives and the further archives; lessening the subgraph to shape the ontology by necessitating that reports in that be demonstrative of a second area depiction, the second area portrayal being in any event as expansive as the seed area depiction. Michael et al [109], to integrate crowdsourced human feedback, integrates design and communication patterns directly into user interfaces. They emphasis on writing and editing, nuanced activities covering many conceptual and practical scales. Mika [110], tested an integrated social and situational networking framework. In this research, the standard bipartite ontology model is extended with social calculation, prompting a unitary system of participants, principles and instances. Missiroli et al [111], published a high school report on the growth of agile software learning. This paper tests I the execution, (ii) the reliability of the code and (iii) the accomplishment of both undergraduates and educators in the use of upgrades. Moniruzzaman and Hossain [112], identified and de-

picted the main considerations, that agile development approach enhances software development procedure to meet the quickly changing business conditions. Additionally, give a short examination of agile development procedures with conventional frameworks development philosophies, and talk about the current condition of receiving agile strategies. They approximate that from a need to satisfy the customer before the schedule and the non-stop transmission of the profitable software, agile software development rises as a contrasting option to a conventional arrangement based Software development techniques. Mujumdar et al [113], examined how proposals can be accommodated dynamic, translated web development dialects. Extremely important interpretation is to cultivate test-driven tools to collect bug fix cases. They show Crowd: Debug, a device that recognizes these advantages for Ruby developers. Murphy [114], studied the effects to find out that the silver bullet to Microsoft's software development has been agile techniques? The overview comes about uncover that regardless of exceptional market weight, the development of agile reception at Microsoft is slower than would be normal. Naik [115], presented the meaning of CROWD (community, remoteness, open-call, web, diversity) in crowdsourcing and specifically software crowdsourcing. It elucidates the significance of these five crowd parts of software crowdsourcing and their commitment to making it effective. Nakatsu and Iacovou [116], examined the hazard elements of outsourced Software development. The very first objective was to make quantitatively produced accommodations of hazard factors for outsourced ventures both locally and at sea, and the second objective was to examine these two configurations: how and why the hazard factors modify which variables are most important each. To address these goals, they directed two Delphi studies to recognize the critical hazard factors from a customer point of view, in residential and seaward settings. Narock et al [117], investigated

large data in the geosciences and unanswered questions about the integration of crowdsourcing and semantics. Noll and Atkinson [118], conducted an analysis endeavoring to contrast, XP with a conventional approach with understudy software designing tasks entrusted two gatherings of understudy groups with creating as an impleroom reservation framework, one gathering following a customary development process, the other utilizing XP and watched minimal real contrast between the items conveyed by either process. Pan and Horrocks [119], conducted research to examine some of the issues behind this paradox, proposing a novel RDF(S) switch as a firm semantic structure for a substantial number of the new logic-based definition ontology dialects, particularly web ontology DL. Papamartzivanos et al [120], outlined a cloud-based framework that works under a crowdsourcing rationale, with the expectation to give i) a constant security flow following administration, ii) a shared foundation for trading data identified with applications' protection introduction level, and iii) conceivably a conduct was driven recognition system with an end goal to exploit the crowdsourcing information to its most extreme efficacy. Parsons et al [121], examined the effect of tools and techniques on the performance of agile software development programs. In this unique situation, our investigation recommends that picking particular mixes of strategies can be especially advantageous. Pastore et al [122], still dreams of semiautomatic software testing: any formed test insight depends largely on a test oracle that decides relativism and, with the exception of generic properties such as ' the program is not to crash, ' such oracles require human input in one form or another. Pedersen et al [123], examined work on crowdsourcing, focusing on top diaries and IS gatherings. As far as anyone is concerned, this writer's main effort in the teaching of the IS. They illustrate how current IS writing applies to the components of this rational model: problem, people (problem holder, person, and crowd), leader-

ship, system, engineering, and consequence. They close with recommendations for future research. Peng, X et al [124], article provides an overview of current technologies for crowdsourcing. Valencia and Terriza [125], studied WEB ONTOLOGY LANGUAGE ontologies improved with open linked data has been proposed as a principal approach to portraying standard communitarian units that can connect in a social setting. The overall Ontology have named IoA-web ontology language considers utilitarian and non-practical viewpoints composed in six profiles, for example, operator, setting, display, administration, social, and protest. Pierre et al [126], introduced an apparatus upheld model-driven approach for crowdsourcing UI adjustments. They assessed proposed instrument and device through an essential primer client think about. Poller et al [127], contemplated how security consultancy influenced authoritative schedules in a software development gathering. Security experts tried their item, revealed vulnerabilities, and conveyed security preparing. They took over the gathering amid and after consultancy occasions. Popli and Chauhan [128], demonstrated the estimation of cost and effort in agile software development. Quan [129], suggested a representation plot of web ontology web ontology language web ontology dialect based on ontology code. Throughout this project, the customer-furnished segment representation information are linked to ontology to show semantic value, which designs and builds the establishment in the division process of recovery for semantic thinking. Rai and Dhir [130], described the effect of examining various conventional systems and new approaches. This paper discusses the reasons behind which software companies have shifted from Traditional RE to Agile RE. This study has carried out comparative investigations of different methodologies of software development, when and where they can be used. Rani et al [131], conducted a survey for distinguishing key elements of agile quality confirmation and the writing audit related to the

quality affirmation markers of lightweight philosophies. Rao et al [132], transmitted with the distinctive goals of looking at and gathering bits of information into current agile objectives and work on, understanding the qualities and weaknesses of agile techniques and various issues of their suitability. Subjective and quantitative research approaches were used to fulfill the goals and priorities set. Apart from a systematized review of writing and audit, four technology organizations in India undertook a qualitative analysis. Riedl et al [133], demonstrated a portion of the potential advantages of crowdsourcing restorative research. The greatest preferred standpoint is the open door it gives specialists to complete examinations considerably quicker and less extravagantly. Crowdsourcing gives researchers a chance to create, test, refine, and retest thoughts at a phenomenal rate. Salay et al [134], established a new model for software development labeled contribution execution (CI) propelled by the manner in which people moderate this trickiness: sources are dealt with as supposition suppliers with fluctuating measures of trust and collecting various conclusions from various sources enhances the nature of the appropriate responses. Sanghvi et al [135], recognized the hidden individuals' components to think about while receiving agile for a group to be compelling. The strategy used is the investigation of 3 totally unique measured agile groups, creating stock supporting indistinguishable advancements and using Scrum. Every goal and subjective measures were utilized and in this way, the outcomes are upheld by a study. The outcomes obviously demonstrate that for agile methodologies to figure well, it's significant to pick the best possible individuals for the best possible group. Sauerwein et al [136], conducted a study to reveal some insight into this by directing an orderly writing survey in view of the snowballing strategy. It conveyed 23 important papers which they broke down as for the accompanying points of view: (a) bibliographic data, (b) connected research procedure, (c) tended to da-

ta security application setting, (d) connected crowdsourcing methodology, and (e) challenges for crowdsourcing based research in data security. Saxton et al [137], defined crowdsourcing that joins "Groups," outsourcing, and social web innovations. They at that point dissect 103 surely understood crowdsourcing sites utilizing content examination techniques and the hermeneutic perusing guideline. In view of their investigation, they build up an "Ordered Theory" of crowdsourcing by arranging the observational variations in nine unmistakable types of crowdsourcing models. Shankhdhar and Darbari [138], deal with genomic data retrieval. O-MaSE is a MaSE development and a more deliberate agent creation approach compared to standard strategies. In sequencing organisms, the biologists produce a large quantity of minimally processed data and the data is circulated and heterogeneous, dispersed in multiple formats and networks. Sheeba [139], analyzed about the semantics connected to learning substance to empower extensive scale coordinated effort of E-Learning exercises over administration situated foundations. Additionally, the hugeness of Semantic Web in E-Learning content and the utilization of ontology in creating E-Learning content are examined. Sherief et al [140], appraised the movement of acquiring assessment criticism from the group and lead two concentration gatherings to comprehend the different parts of such an action. They gave an account of an arrangement of difficulties to address and acknowledge right and effective crowdsourcing systems for Software assessment. In informative exercises, Skaržauskaitė [141], examined the part of crowdsourcing use. In the late decades, the rapid development of imaginative Internet-based data and correspondence technologies gave instructive institutions another field of opportunity to achieve their goals. Sletholt et al [142], analyzed the discoveries of the literature review on the consequences of a formerly directed study. The evaluation shows that projects getting agile practices for logical software devel-

opment see their testing as superior to normal. No distinction to normal tasks was seen with respect to prerequisites related exercises. Snijders et al [143], published a study to suggest Crowd-Centric basic essentials Engineering (CCRE) as a reconsidered requirement engineering technique where customers wind up essential benefactors, creating higher quality prerequisites and expanding customer satisfaction. Soundararajan et al [144], proposed an organized, efficient, and complete way to deal with evaluating the 'integrity' of agile methods. They look at an agile method in light of (1) its sufficiency, (2) the ability of the association to help the embraced standards and practices determined by the strategy, and (3) the technique's viability. To control their evaluation, they propose the framework of goals, ethics, and procedures. Stol et al. [145], published a study to encourage professionals in computing to understand how crowdsourcing would support software systems tomorrow. Stol and Fitzgerald [146], led an investigation to show an exploration structure that includes the diverse mixes of partners and key concerns. The structure can be helpful to control future research on the usage of crowdsourcing as a 'sourcing' technique, and moreover, a way to audit and orchestrate look into findings in order to have the capacity to analyze ponders on crowdsourcing in a Software development setting. Stol et al [147], conducted a study to decide a course of action of key stresses from the crowdsourcing composing as a fundamental demonstrative structure for an exploratory logical examination in a Fortune 500 association. We complement the empirical results of 13,602 crowdsourcing conflicts over a ten-year period on the highly well-understood crowdsourcing stage of Topcoder. Drawing from our accurate findings and the crowdsourcing composing, tells a hypothetical model of group intrigue and genuine cooperation in crowdsourcing rivalries. They evaluate this model using Structural Modeling Equation. Amongst the findings is that the level of prize and duration of rivalries

do not increase swarm enthusiasm for rivalries significantly. Stolee et al [148], indicated how they adjusted an examination to be performed under a framework that not just makes it conceivable to achieve a vast base of clients yet it additionally gives capacities to deal with those clients as the investigation is being led. They examine the exercises learned through this experience, which delineate the potential and tradeoffs of crowdsourcing software designing examinations. A research on the effect of social networking sites on software engineering activities and methods was performed by Storey et al [149]. In this discussion paper, they recommend exploring efforts to understand the advantages, risks, and disadvantages of using web-based life in team, project, and network-level software development. Steered by the implications of the highlights of ebb and flow apps and digital networking. We proposed an arrangement of appropriate research inquiries regarding network participation, project management and administration, as well as special experiments in software development. Störl et al [150], examined the requirement for a construction administration segment equipped for dealing with the Schema, Schema advancement, and in addition information movement inside NoSQL database frameworks. This demo, show Darwin, a middleware for methodical, instrument-based help specifically intended for NoSQL Database frameworks. Darwin does diagram development and information movement assignments. To the best of our insight, Darwin is the first apparatus of its kind that backings both enthusiastic and lethargic NoSQL information movement. Suganthy and Chithralekha [151], researched crowdsourcing regions and explored the effectiveness of crowdsourcing in the production of apps. This research defined the intrinsic attributes of crowdsourcing software curiosities and their related issues. Except for this, the order of the related issues is also given in this document. Tarwani and Chug [152], reviewed agile methodologies in software maintenance. This investigation audits

a few papers with various contextual investigations to assess the execution and nature of software utilizing agile procedures. In this examination, in excess of 30 inquire about examinations are explored which are directed somewhere in the range of 2001 and 2015 and have been ordered by the distribution year, datasets, devices, sort of methods and so on. This will be the principal survey paper on the utilization of the agile in software support which will encourage the specialists and urges organizations and amateurs to embrace these techniques to pick up Software quality. Thomas et al [153], conducted an exploration plan, inspecting the topic of whether similar sorts of accomplishments that micro-task crowdsourcing is having in upsetting different areas can be conveyed to software development. Tripp [154], studied the impacts of agile development methodology use on project success. The exploration outline for the examination used a study that gathered reactions from 83 agile development groups. Tsai et al [155], found that all software development tasks can be crowdsourced, along with requirements, designing, coding, testing, progression, and documentation. Agile software crowdsourcing practices obscure the capability among customers and specialists, and take after the co-creation standard — that is, a general client turns into a co-originator, co-developer, and co-maintainer. Ullah [156], discovered the issues in conventional software development and to indicate how spry techniques lessened those issues in programming improvement. The examination additionally centers the distinctive achievement variables of agile strategies, the achievement rate of agile projects and correlation amongst traditional and agile software development. Vaish and Davis [157], depicted the system in detail, and its usage in the field through two pilots ponders, including progressing work for an expansive scale future venture. In the first pilot ponder they delivered a distributed paper, and in the second, they created better outcomes for an open finished issue being tended to by the best

research gathering. At last, they talk about the examination difficulties and openings in the space. Vijayarathy and Butler [158], conducted a study on Choice of Software development methodologies. Researchers reviewed venture directors and other colleagues about their selection of techniques. The outcomes demonstrate that albeit agile strategies, for example, Agile Unified Process and SCRUM are more common than traditional methods 10 years earlier, including the cascade display, are as yet well known. Vukovic et al [159], studied the effects showing an example crowdsourcing situation in the field of software development to suggest the need for a universally useful cloud-based crowdsourcing management. This suggests a theoretical definition for an order of crowdsourcing stages and tests various current systems against distinguished highlights arrangements. Wang et al [160], Local-based Active Classification (LOAF) suggested organizing true culpability from crowdsourced confirmatory testing. Portion prescribes a few occurrences which are most enlightening inside the nearby neighborhood, and asks client their marks, at that point learns classifiers in view of the nearby neighborhood. Their assessment of 14,609 test reports of 34 business ventures from one of the chinese biggest crowdsourced testing stages demonstrates that our proposed LOAF can create promising outcomes. Wasserman [161], studied software engineering issues for mobile application development. This paper gives a diagram of vital software designing examination issues identified with the development of uses that keep running on cell phones. Among the themes are development forms, instruments, UI plan, application movability, quality, and security. Wexler [162], conducted a study to look at the way in which backers of crowdsourcing reconfigure the established sociological treatment of the group. Wongthongtham et al [163], proposed another way to deal with software building. They sort out Software designing ideas, thoughts, and information alongside Software development philoso-

phies, apparatuses and systems into ontologies and utilize them as a reason for ordering the ideas in correspondence and permitting learning sharing. Wongthongtham et al [164], conducted a study to exhibit an ontology model of software designing to speak to its information. The essential learning identifying with software designing is all around portrayed in the reading material titled software engineering by Sommerville that is currently in its eighth version and the white paper, Software Engineering Body Of Knowledge (SWEBOK), by the IEEE whereupon software building ontology is based. The study offers an inquiry into what is ontology designing code, what it contains, and what it is used to use case scenarios. Wu et al [165], perceived the min-max nature among individuals as a basic outline segment in software crowdsourcing for software ingenuity and quality. Despite the fact that in a min-max diversion, one gathered attempts to increase the discovery of bugs in an arrangement of ancient rarities and other gatherings attempt to limit possible bugs in a similar artifact, crowdsourcing technology may actually be a cooperative and a win - win system for all gathers. Wu et al [166], conducted a study to define an amusement hypothesis model to examine the essential factors in this min-max rivalry decide that affect the idea of support and in addition the software quality. They depict Harvard-TopCoder and AppStori, two methods of crowdsourcing. The system shows the strong contradictions between the two forms of crowdsourcing as participants will have exceptional experiences in conjunction with these two ventures. Ying and Fu [167], proposed a formal setting model in light of ontology utilizing web ontology language to acknowledge semantic setting portrayal, setting thinking and learning sharing, setting arrangement et cetera. Their setting model catches different settings by presenting a grouping plan; catches connections between various setting data by acquainting reliance tag with the property related with a predefined setting class; catches nature of setting by com-

menting on detected setting with extensible quality limitations. Yuan et al [168], observed that when they broke down the product administrations capacity, expanded by the client-driven prerequisites in arranging condition. At that point, the agile development methodology is proposed in light of clients' singularities and setting, which is a "meet in center" strategy for programming advancement. Driven by clients' assorted variety and customized necessities this strategy can be displayed as the arrangement of administration assets on request and the way to deal with the objective steadily. This is a powerful method to meet expanding application Software requirements. Yuen et al [169], overviewed on the writing on crowdsourcing which is classified by their applications, calculations, exhibitions, and datasets. This paper gives an organized perspective of the examination on crowdsourcing to date. Zakariah [170], reviewed business model of crowdsourcing. This investigation investigated one of crowdsourcing critical part which is the crowdsourcing plan of action. Plans of action are exceptionally immediate as it will clarify an arrangement for a company's effective task, distinguishing revenue wellsprings, the proposed customer base, items and financing interest points. Zhao et al [171], conducted a Research on using protege to build Ontology. They present how to utilize the Protégé based web ontology language API to parse Ontology display. In particular, it incorporates how to stack the ontology demonstrate from the web ontology language record, how to gather the class, subclass, singular, question property, information property, and how to discover the space and range that are significant to some protest property. Zogaj [172], demonstrates that testCloud faces three significant challenges: process management, crowd management, and creativity coping. Detail components to address the challenges associated with crowdsourcing projects for each metric that testCloud applies.

CHAPTER 3

INVESTIGATION OF THE RELIEVANCE OF CROWDSOURCING IN AGILE SOFTWARE DEVELOPMENT

Agile software has been a rapidly growing research domain for more than a decade. Nonetheless direct application of agile method in actual practice is weighted down by few confront, with required to be determined in the field of industry and higher academics. One of the main concerns is the scarcity of resource, assets and funds for creation of application and development of agile methodology and techniques. The crowd sourcing methods in this field of software development like in top coder and apple app have shown encouraging and achievable solutions to various problems. Use of crowd methodology in the field of software development is appraised to take its place along with existing methodologies on techniques like service-oriented computing agile. In this research, we put forward an inquisitive and basic conceptual structure of crowdsourcing strategies to scrutinize its minute details of existence and emergence in agile software development. The work has been put forth with the concept of crowdsourcing recognition into the current agile methods along with offering strategies of crowdsourcing to examine the applicability and existence of crowdsourcing strategies like crowd creation, crowd reviews, crowd democracy, crowd wisdom and crowd funding, we achieve quantitative and qualitative research techniques. We also describe the principles of crowdsourcing strategies in agile development process.

Crowdsourcing, an eminent form of externalization software development has restored the way in the field of several leading models of organizing work. The following chapter discusses the perspective of crowdsourcing in agile software development, since it focuses on the key motivations for engaging, and highlights some of the major challenges that lie ahead. Here the author analyzes whether the task presently projected on crowdsourcing platforms have the essential qualities of agile development of software. The major task is to, find a large figure of suitable issues to assess a technique in agile software development and inquire

into use of crowdsourcing as a method of tackling this challenge by helping to develop agile software. In this direction, try to survey both-

(i) The crowdsourcing platforms, which are more worthy to agile software development, among the existing ones.

(ii) The type of tasks, which are more pertinent to crowdsourcing in agile software development.

Altogether, through this work, we have scrutinized several factors where both methodologies can be clubbed in order to make a new product. Furthermore, we need to increase the productivity and provide a better end result and services.

3.1 Introduction

Software engineering is the approach used to determine customer requirements and to model, build and review end-user products with software programming languages to fulfil the requirements. Software engineering is the mechanism of evaluating client needs and designing, designing and testing end-user requests with software programming languages that will accommodate these requirements. It is the pathway for software development of engineering strategies. In contrast to simplistic programming, software engineering is being used for bigger and more complex software programs which could be used as vital systems for businesses and organizations. The mechanism of software development is a generalized term that describes the under-arching mechanism of software product formation. The method can be used to incorporate a single application or a far-reaching ERP framework, often referred to as the computer life cycle. The diverse demands of consumers make it even more difficult. The growing and advanced business requirements cannot be adequately met by conventional

software development methods. Software development techniques, including such agile methodologies of software development, have therefore been developed primarily to solve such a problem.

Agile software development is a structured process for software engineering that facilitates iterative progress all across the project's life cycle, close cooperation between the entire development team and the business side, constant engagement, and closely knit family. In general, agile methodologies facilitate a cycle of project management that requires constant examination and transformation. This theory of leadership encourages teamwork, self-organization, and responsibility. ASD is indeed a collection of best practices in IT (allowing fast delivery of high-quality software) and a wholistic approach (aligning technology with consumer needs and goals).

The founding principles of ASD are base on some existing ethics of software development, despite being a new approach, both from the areas of software engineering, information systems and others such as manufacturing. Not only we have examined the applicability of crowdsourcing in agile software development, but also pinpointed some papers in which organizations have used new agile methods of software development under various circumstances application of big data and cloud computing have been mainly used in handling crowdsourcing.

Agile software engineering was restricted to a small group of developers, but is slowly widely spreading in communities, institutes and organizations involving large groups of people associated with it. There is an increase in globalization with a stand-point on skillfully organized practices and infrastructure. Only evolving method of taking work completed is crowd-

sourcing, a sourcing concept [72], that was found in 1990's. Chosen by web 2.0 technologies, associations with an Internet connection can build a workforce comprising of anybody. Clients can post tasks or works on crowdsourcing platforms, where a crowd of people on individual workers opt for work that is in accordance of their capabilities and skills.

The frequently used methods for solving these difficulties includes carrying out surveys, formation of committees, a more efficient way of overcoming this problem is crowdsourcing.

First of all, various social media platforms like Facebook, Twitter, and Skype etc. have altered meaning of its presence. The web medium allows the aggregate cognitive abilities of the masses in method. Due to time and place restrictions, face-to-face communication is practically impossible at all times. Our research states that crowdsourcing model is a prosperous and achievable web based distributed technique that helps to overcome difficulties and productions method for small business. It is an efficient model for furnishing agile software development processes.

Crowdsourcing has matured as a substantial feature for development of any agile products, as it also allows the candidness and exchanging of communities collected resources and knowledge. Crowdsourcing is a way by which Software developers, designers, testers and others not only develop framework for any software or application, but also manages the collective resources of various contributors and customizing products according to their goal needs. Completing or solving a full project or software related problem can be a difficult job for few people in the organizations. With crowdsourcing, even non-professional enthusiasts have the chance to participate in the development of new types of agile software applications.

Conventionally, the big data technique has also used for crowdsourcing development. As stated by Jeff Howe [72], "Crowdsourcing is a method of taking a job generally executed by an assigned worker, and outsourcing it to a vague, ordinarily, a huge group of individuals as an open call". We have discussed several leaven aging of new technologies, creating a further conjunct corporate culture and emphasizing on an opportunity to distinguish the traditional methodologies. We are focused on the work of Popli and Chauhan [128], focused on analyzing the development and estimation of agile software in spry. They prescribed a strategy for the precise expense and effort estimation, to escape from the problems of current agile practices. The agile methods, which are based on the concepts of versatility and flexibility, constitute an evolving set of software, are currently being used as a gimmick to these recurring problems and make a straightforward way for the future of development. A paper by Altameem [7], explains several methods, in which agile methodology has been, proved to be powerful in software development. It additionally explains the focal points and the confinements of agile technique. This research work motivates us, to adopt this technique, to develop software that proves out to be a remedy for their changing needs. Kumar and Bhatia [91], helps us to recognize the significance of methodology, in terms of its quality within the cultural framework. We have extended the work of Rao et al [132], differentiated between the weaknesses and strengths of agile methods and different predictive validity issues. Wu and Tsai [165], examined the data in detail, assembled on software crowdsourcing and abridge major lessons, adapted, at that point dissect two software crowdsourcing forms, including aesthetic ways. The paper by Rai and Dhir [130], elucidates the influence and comparison of several traditional techniques and a new methodology i.e. Top Coder and App Store processes. In conclusion, they identify the min-max nature among participants as a key

component of software crowdsourcing design for software quality and scrutinize the motives why software enterprises have moved from traditional requirement engineering to agile requirement engineering. The concept by Jamaalamadakil and Krishna [79], underlined couple of confrontations with agile scrum and offers vision to the client whether the agile is wonder drug. Hneif and Ow [71], exhibited their inspection more than three agile methodologies, including agile modelling, extreme programming and scrum differentiate them and advice when to use them and also reviews different agile methodology based on their characteristics, objectives, boons, and banes of using the agile methodology and their unique characteristics.

3.2 Crowdsourcing concept onto Agile Software Development

Crowdsourcing is a process whereby a task, problem or plan is overcome and accomplished through a network of informal and geographically dispersed stakeholders. Crowdsourcing is a collaborative method or strategy of concern-solving that needs support from a network or crowd of people. This network may be linked through the Internet or through a similar website.

Crowdsourcing is a form of collective intelligence, the general idea being that the actions of various individuals can result in the emergence of information processing. The consideration of the forces that drive the current emergence of the current crowdsourcing models, platforms, and environments, particularly in terms of their significance in software development organizations is quite engaging and interesting. Many of the models are novel, and their drawbacks, and corresponding advantages are not really understood well. However, businesses should see the substantial benefits in adopting and using the process of crowdsourcing, even if not to a great extent.

In this chapter, we explore the several motivations that drive software organizations to acquire the crowdsourcing process, along with the forces motivations the developers to participate in these activities. After the need for a simple or less complex way to develop software to meet changing requirements environment, agile methodologies came into reality. Agile methodologies allow these processes that allow communication between developers and consumers, and experience production-delivery-feedback cycles, provide a clear and accurate view of the specifications, and are ready for any form of change at a certain time. The goal of agile methodologies is to deliver at the right time what might be required. The agile scheme features the accompanying:

- Individuals and intercommunication's over procedures and devices.
- Working software over comprehensive documentations.
- Responding to change over after an arrangement.
- Consumer partnership over contract negotiation.

3.3 Strength of Crowdsourcing

Crowdsourcing is an intriguing concept that requires a great deal of care. One has to find some abilities, employed to assure that at the completion of the project he receives what he absolutely needs. It should also be the responsibility of the experts who are the crowd workers to ensure that they do not exercise unacceptable behavior and that they only pick jobs with which they are satisfied. You can't perform unprofessional activities to pick up. Crowdsourcing has already changed the current software development scenario through its various forms. Open source aside, the number of unspoilt crowdsourcing platforms, the no. of corporations actively evaluating with crowdsourcing, the number of employees signing up and

continuously contributing indicate, a process that have stepped into the software development industry through countless ways.

The probable supremacies of adopting the crowdsourcing practices are concrete. The increment in the shift of development, work more dramatic shifts and occurrence to fluid labor markets alone, and does predict the potential for even. Such shifts might raise a much anticipated question and investigate the long-held fundamental beliefs about the software development practices, as in any fundamentally disorderly shift, the eventual spin-off is far from being particular. Now the question of the hour is that, will future developers be half-witted automatons, or will they be operating as freelancers, choosing micro tasks to follow their passion and intensify their skills serious challenges have to be undertaken, if crowdsourcing is going to be that significant in the software enterprises, as it has proved to be in other business fields. The nature of software has much to do with this regard. Software development is compounded and complicated, and cannot be easily decoded into distinctly segmented, self-contained and is not easily grasped. It is important for the group to develop an extensive knowledge of how and when to apply crowdsourcing to software development projects. Crowdsourcing has been widely used as a tool in the agile development of applications to manage several important works. Agile software development crowdsourcing studies can also be used to combine two or more concepts to produce research findings. After inspection on the fulfillment of crowdsourcing, this research can be used as a procurement approach.

3.4 Suitability of Crowdsourcing strategies in Agile Development process

We examine the applicability of crowdsourcing strategies (Crowd Creation, Crowd Wisdom, Crowd Democracy, Crowd Reviews and Crowd funding) in agile software development

process. We find out different characteristics of crowdsourcing strategies, which are more helpful to build small agile projects.

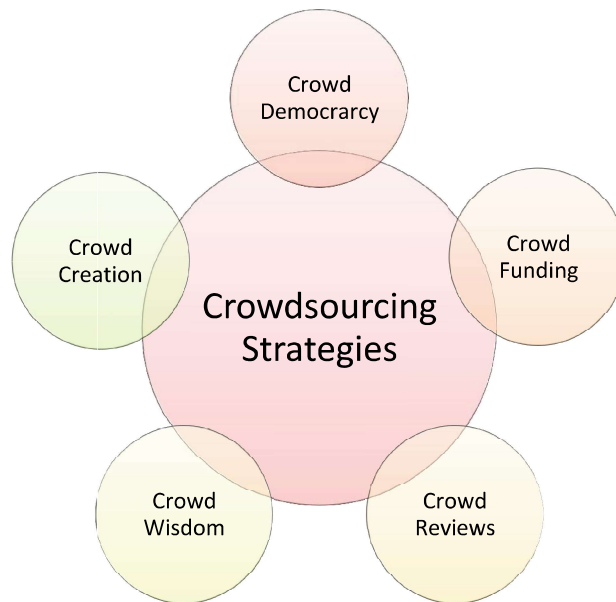


Figure 3.1: Crowd Strategies

3.4.1 Crowd Creation

Crowd creation is the most common form of crowdsourcing. It is related to build up activities of software development process, asking personage and business to clear up a specific state of uncertainty, so that produce an adequate answer to the particular problem. Resulting product is the crowd development production, even if it remains a tangible quality in intellectual capital or physical form (software product). One of the three ways (cattle calling, warehousing process, and collaborative project) is the development of crowds.

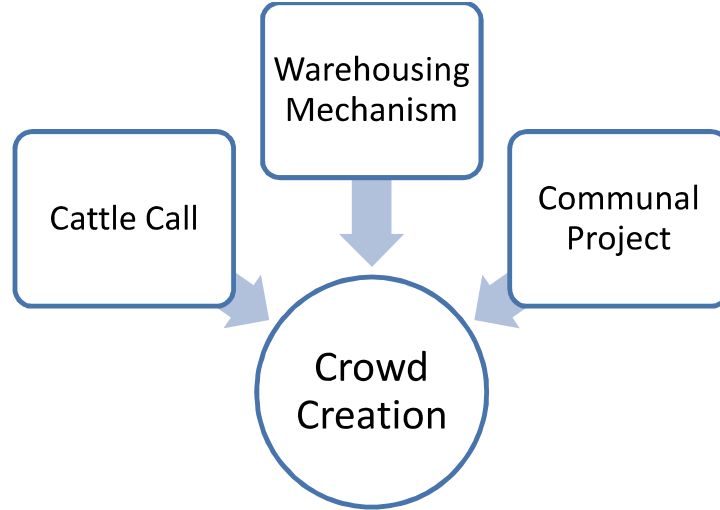


Figure 3.2: Three forms of Crowd Creation

3.4.1.1 Cattle Call

In the first form of crowd creation, to achieve the specific need, company or individuals places a request for submission. Incidences of this are slogan, commercials generated by fans or concepts of design. These have a clear cut task of a product promotion or work completion. Individuals or scrum master convey their complete design to the wishing entity in their complete form. Selection process has followed for choosing either a small group or single selection. This selection process can be done through crowd voting or inside the company professionals.

3.4.1.2 Warehousing Mechanism

As same to the cattle call, scrum master or entity places a requirement on the web for any project or task in warehousing mechanism. The specific content and submission do not

achieve by the objective of the site, that type of restriction proposed by the requester. Software organization or software professionals upload their innovative creation or product to the base entities website to be seeable through the people at large. Dissimilar cattle call, for selection process, there is no constraint followed by the company; those items are displayed that achieve to guidelines or instructions. Customers have a flexibility to choose either the developed software product or the raw content itself. The core company allows the assortment of customers in the form of sales. After completion of the transaction, revenues sale are divided into the software company and the content provider at a fated rate. One another limitation of the form is, possession authority of the image may or may not with the begetter.

3.4.1.3 Communal project

The main contrast among other framework of crowd creation and collective project is that, from the crowd are member of whole project, and creates the end product. In the era of internet, centers basically close to cerebral capital as adverse to physical formation like programming code. Collective project has two forms.

(i) Cattle call of multistage

Contains all these process:

- Forthcoming with the new concept
- Determine either it creates a attainable production candidate
- Examining the completed design of the project
- Generating the final item is break down into individual chunks

(ii) Open source initiative

Generally, application of that type of project drawn under the form of open source initiatives or micro work. Commonly, solutions have to adjust inside the framework of the application.

3.4.2 Crowd Wisdom

Crowd Wisdom is indeed the mechanism taking into account. A group of individual consensus opinion, having their own point of view than a single individual. This theory states that answers gathered and discussed with large groups that involve estimation, reasoning, worldly affairs are usually better.

Crowd Wisdom aims at less, but better output and aggregates individual knowledge. It needs explicit command and motivation.

The Key conditions for Crowd Wisdom are:

- Decentralization: Cognition, Co-operation
- Diversity of opinion: Co-ordination

3.4.2.1 Retrospectives

Retrospectives, in the perspective of crowdsourced agile software development, are the conversations that are held after frequent intervals or iterations. During these conversations, the team reflects on what happened in the previous iterations and identifies the areas where more improvement is desired or needed, and takes or plan future actions accordingly. During these conversations an environment of trust and honesty are necessary needed so that each member can share their thoughts without hesitating .This also boosts their confidence levels.

3.4.3 Crowd Democracy

It is used in policymaking; it also takes into account other professional's opinions and collects information. The US federal government has been using crowdsourcing for several years as a way of a citizen's participation in national open government. Openness and collaboration being at its core.

Crowdsourcing is used in every stage of agile software development. Although crowdsourcing is a new and is becoming a prominent technology, it has been used by a number of global software enterprises in their projects.

With the aid of crowd democracy companies or an individual can get direct critical assessment from its customers. It is also used in many non-profit organizations. After developing project under crowdsourcing method, crowd democracy plays an important part to guarantee their quality.

3.4.4 Crowd Reviews

Main outlook of crowd reviews is taking job requirement from the server for developing the software. G2-crowd is a peer-to-peer business solution review. The company was established in 2012 and furnishes with real time and in partial reviews based on user rating and other salient data so as to provide with assessment of what is most beneficial for the business.

A few problems continue to rise because team members are unable to understand the meaning of previous meetings and are unwilling to engage in them and thus doubt their meaning.

Now, here are a few difficulties faced along with their elucidation and solutions:

- Thinking of one's team to be flawless for retrospective: It is impossible for any team to be proficient, that there is no need of improvements. Even if it happens it shows that the team that is not professing. Hence no team is completely skilled and perfect.
- At times members find the meetings monotonous. So, it is nowhere mentioned that they should be enthusiastic about the meetings and it is carried out in accordance of all the rules and regulations. And appropriate behavior should be maintained.
- There are some who dislike retrospectives. It is impossible to like all parts of jobs but it is important to partake in all meetings for the sake of professionalism.

3.4.5 Crowd Funding

In the field of software development, every person is trying to earn more for their creative ideas, in the past capital funds and kind investors were options for getting large pay. These days, crowd funding has become a realistic resource for developers to gain assets but do not wish to forfeit investment or go through the exhausting procedure of being questioned by the financier. Crowdsourcing is a new method for entrepreneurs or software developers to raise capital by tapping the masses rather than a small pool of financiers. Some instances might have heard of a few, like kick starter helps to find the resources and support they desire to form their creative ideas and bring them to reality. According to our deduction, "Kickstarter is the best way to connect the people who can actually help you."

Crowd funding works by allowing anyone to provide relatively small help to a project in exchange for real goods, rather than financiers investing and risking huge amount of contribution funds differ from campaign to campaign but usually ranges from \$1 to \$10000. Crowdsourcing sites radically ill fame in the creative communities of groups.

3.5 Conclusion

The chapter investigates the present state of research in terms of agile process of development and crowdsourcing. The current chapter will be used as a starting point for applying agile crowdsourcing development process from a research perspective.

Although one of the assumptions that can be taken from the above research is that the use of agile methodologies and crowdsourcing would certainly bring significant benefits for companies. Crowdsourcing strategies (crowd funding, crowd creation, crowd reviews, crowd democracy, and crowd wisdom) will be pertained by various fields of agile software development and specification will be strict and bounded.

CHAPTER 4

ANATOMIZE THE SUITABILITY OF CROWDSOURCING IN AGILE SOFTWARE DEVELOPMENT

3.5 Conclusion

The chapter investigates the present state of research in terms of agile process of development and crowdsourcing. The current chapter will be used as a starting point for applying agile crowdsourcing development process from a research perspective.

Although one of the assumptions that can be taken from the above research is that the use of agile methodologies and crowdsourcing would certainly bring significant benefits for companies. Crowdsourcing strategies (crowd funding, crowd creation, crowd reviews, crowd democracy, and crowd wisdom) will be pertained by various fields of agile software development and specification will be strict and bounded.

CHAPTER 4

ANATOMIZE THE SUITABILITY OF CROWDSOURCING IN AGILE SOFTWARE DEVELOPMENT

The present chapter defines the purpose of thorough examination of the crowdsourcing application, and to learn the behavior, thus reflected and analyze the suitability in agile software development. Agile software crowdsourcing is the exit domain of software engineering. The determination of suitability of crowdsourcing in agile software development is done via examination of allied literary works on crowdsourcing structure along with related case study material, and drawing out the characteristics which are referred by them as a part of the development of agile software.

The ideology in this chapter strongly stresses a research agenda; however, exploratory facts reflect that this concept of crowdsourcing can support agile software development with high quality performance.

4.1 Introduction

From the crowdsourcing research view, where the main emphasis is on one of three aspects: the supplier aspect, the customer aspect and the system aspect (crowdsourcing platform) are the procedure of collecting ideas, services or information by collecting earnings from a large group of people, including the online community, despite customary staff or service providers. Crowdsourcing industry is growing rapidly, not for graphic design only, but also for various other services.

A study shows the numerous benefits which have been connected with the utilization of crowdsourcing in agile software development.

4.1.1 Deduction of cost

In the software development community, crowdsourcing is generally sought as a cost effective manner to collect labels for training data. Within crowd, unlabeled instances are outsourced to a large group of people due to this bypassing of the extra cost burden commonly exhibited in hiring software developers.

4.1.2 Reduction of actual development time

The time consumed in the development phase is reduced through crowdsourcing, in agile software development. Various decomposed tasks are developed parallel to each other, hence reducing the actual time of software development. Flexibility in working time (specialized talent or ability who can accomplish take after the-sun advancement crosswise over time zones and who will work at ends of the week plays a key role for faster delivery of the software product.

4.1.3 Broad participation means higher quality

Crowdsourcing provides the platform to a huge pool of development professionals to showcase their talents and expertise in various fields. They take part in competitions where the winning solution for any question is selected for the development process. They take part in competitions where the winner solution for any problem is selected for the development process.

4.1.4 Open innovation and creativity

The enormous range of professionals and experts ensures very innovative solutions, clarifications, explanations, which more often than not avoid the fixed mindsets hovering within par-

ticular organizations or companies. This concept is basically referred as “Near field repurposing of knowledge”.

We are inspired by the work of Buecheler et al [25], explained 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. The research and works of Maddalena et al [44], enlightened the possibilities that if a task proposed on crowdsourcing platforms is appropriate two mobile devices. The 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 research work [141], “The application of crowdsourcing in educational activities”, explain the use of crowd sourcing in educational activities. That soul motive of this publication is to give up brief explanation of how crowdsourcing was chosen by educational organizations as part of their activities in the current scenario time to investigate how outsourcing use would soon be spread over time to other educational activities. According to Hoek's [14], “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. The paper [21], “Reactive Crowdsourcing”, by Alessandro et al proposers and approach to crowd Sourcing which provides find level, powerful and ex-controls and transform these hi-tech requirements into features of a responsive execution environment that facilitates task scheduling, execution and completion as well as performer evaluation. The paper by Stolee and Elabaum [148], explored the utilization of crowdsourcing as a procedure to notify that test by aiding in subject enlistment. Further, to be more ex-

act, with the assistance of his work they can express how study can be performed under foundation that guarantees the potential outcomes to approach a vast base of employments and furthermore permits dealing with those clients while the examination is being carried out. Vaish [157], 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. We were motivated by Humberman's work [75], researched from a large set of data from YouTube that the crowdsourcing output provides a strong positive dependency on publicity measured by the 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. Yuan et al's research paper [168], suggested a smooth, service-oriented software development method for a research perspective. The history of software development process is primarily viewed when reflecting on the driving forces of computer technology. In addition, the reliability of the software services was observed, strengthened by user-driven network environment needs. Narock and Hitzler [117], supposed that implementing search algorithms with crowdsourcing can prove to be a reliable solution. To be precise, look at big data in the geosciences and explain big questions about the convergence of crowdsourcing and semantics.

4.2 Crowdsourcing process

There are a variety of steps to be undertaken at the stage of crowdsourcing a project or goal. The procedure associated with crowdsourcing an undertaking is portrayed in figure 4.1. The procedure delineated in the figure centers around three noteworthy roles engaged with any

crowdsourcing applications. Included with the procedures are clarified as follows: the business owner introduces the task throughout platform to be crowdsourced. The task depiction ought to unmistakably say the necessities of the solution and indicate the conditions to be trailed by the worker. A platform's enlisted workers are permitted to select the assignment based on their interest. By applying their educated skills, the staff understands the mission and provides the answer to the board. The platform manager should figure out how to analyze the nature of the submitted solutions by assessing the tasks given by every one of the specialists. In specific cases quality checking is likewise done by crowdsourcing technique. It ensures that the plan put together by each of the experts can be installed in a standard vault and the staff is allowed to vote in support of the optimal solution regardless of anybody else.

The exact role of the platform here is that it acts as a middle person here between employer and the employee by giving the company a clear place to express their job and questions and holding a crowd network to find the assignment posted on the platform.

Crowdsourcing



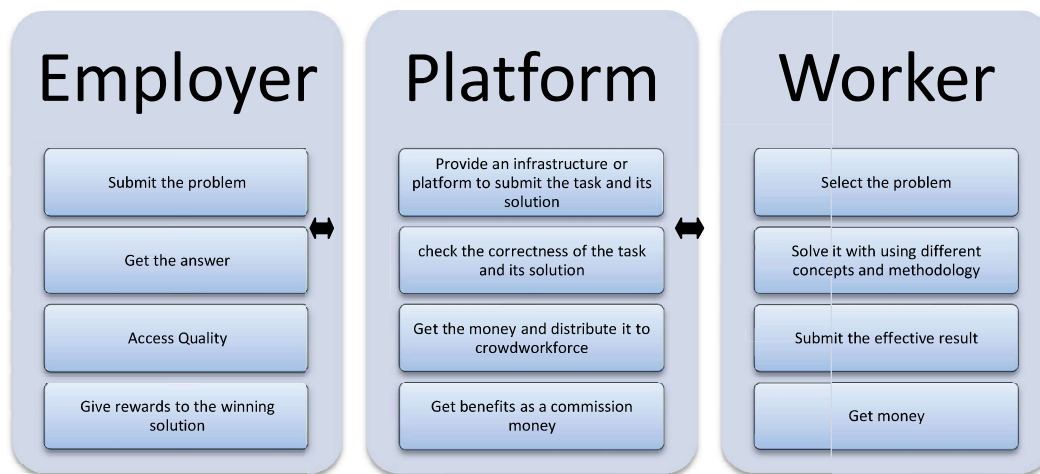


Figure 4.1: Crowdsourcing Process

4.3 Agile Software Crowdsourcing Process Flow

The non-exclusive method correlated with crowdsourcing any activity as per the results of this research work is as continues to follow: any crowdsourcing process starts with new requirements. After assembling the basic requirements, the next step is to characterize the basic needs in the documentation of the task. In view of the company's own rules, the assignment that can be crowdsourced is then recognizable and crowdsourced. There are some rules to be checked before picking up any crowdsourcing task and it relies on the company's hierarchical imperatives that detail the crowdsourcing assignment. If the assignment to be crowdsourced passes into subprojects before being crowdsourced, then at that point the reaction to the sub-task should be accrued to get the answer to the task. Figure 4.2 demonstrates the process flow of agile software crowdsourcing.

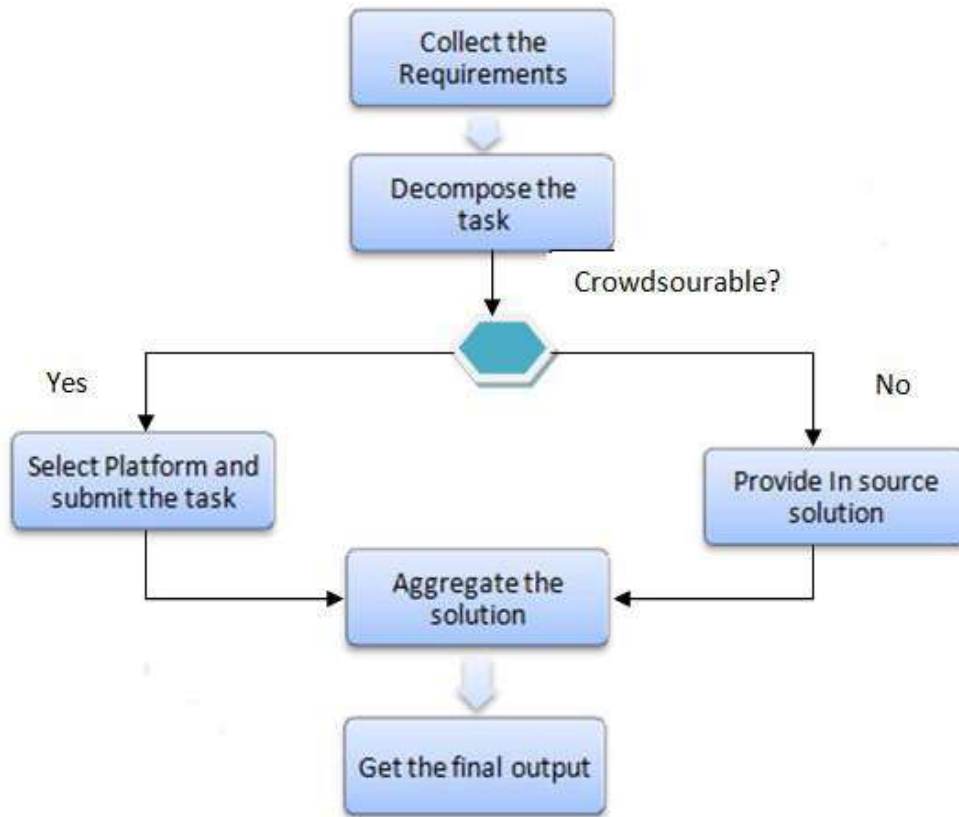


Figure 4.2: Process flow of Agile Software Crowdsourcing

4.4 Key concerns regarding the use of Crowdsourcing in Agile software development

The concern regarding the use of crowdsourcing in agile software development cannot all be covered in boundaries, but some general concerns can be drawn out considering the general literature of crowdsourcing.

Few of these concerns have been mentioned ahead relevant 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

4.4.1 Task Partitioning or Decomposition Problem

Crowdsourcing basically means that the whole is divided into smaller partitions and sub tasks and distributed as an individual task to the crowd. The issue that arises here are the modules defined by Herbsleb and Grinter as “*a responsibility assignment rather than a subprogram*”. A major concern here is that developers designing agile development technology have accurate knowledge of how certain code suits and works into the resulting software being developed based on the assumptions they have already made and the research they have done on the interoperability of the modules. As we know, the tasks are split into smaller independent modules in the general crowdsourcing sense. Nonetheless, the role being created is complex

in terms of agile software development, and there are much greater interdependencies between modules.

4.4.2 Coordination & Communication gap between the organization and crowd

A team can only find success when there is proper coordination and planned execution. With reference to agile software development, there is a dire need of coordination for proper flow of execution.

Co-ordination as defined by Malone and Crowston [102], “The process of managing dependencies among activities”. According to management, coordination is characterized as coordinating efforts of people towards a typical achievable objective and connecting diverse parts of an association for accomplishing a general set of tasks.

All the definitions defined above assume that the task is carried out within an organization, leaving any outsider out of the task. But in agile software development, there is a continuous contact with the client so as to know the interests of the client and make amends accordingly. In crowdsourcing however, the crowd who ‘submits the solution’ is not the part of the organization and different task from different people have interdependencies among them, where, the existence of one is not known to the other, which causes incompatibilities while summing up all the different modules as a whole single unit.

The more the general population included, the higher the correspondence overhead is. This in crowdsourcing context relies on the components like if the work is done in a community oriented way or in a focused way.

4.4.3 Planning, Scheduling and Execution

As with crowdsourcing, different tasks are spread out to various unknown workforces for the job to be done, which, to some extent, means that the organization lets go of the control over various tasks. Advantage in an agile development environment is the timely delivery and review of the tasks which are being completed independent and parallel to each other.

This, however, brings up a certain level of unproductiveness as to whether all the tasks are submitted in the provided time space.

In an organization, the deadline and work pressure compels the employees to complete the task provided to them in time. In crowdsourcing, however, there is this uncertainty of untimely submission of the job took up by any workforce. Crowdsourcing promises to make the product development cycle short.

4.4.4 Quality Concerns

Many advocacies have been made regarding crowdsourcing heaping praises over the high quality of different submissions. But, the probability of the presence of noise in submissions is also considerable that means low quality submissions may also be present. In the context of agile development, crowdsourcing, the wide pool of ideas from a number of developers assist in detection of defects and anomalies. *“Given enough eyeballs, all bugs are shallow”* better explains this. This hints that the crowd has both experts and amateurs, and sometimes amateurs can spot a bug which passed unnoticed by the experts. Problem solving becomes much more efficient, but attracting a handsome number of willing contestants is the main challenge, assuming that the presence of enough contestants will ensure the presence of required expertise. Transparency has quite a role in assuring better quality and improvement.

Sites like Top Coder have a high transparency where the contestants are required to register prior to any competition. Increasing the number of participants, however, the quality of the product will be lower. The crowd is considered to be full of amateurs mostly, which primarily suggests of a lower quality output. But Brabham [22], stressed this as a total myth. The main aspect of software development is quality assurance, doesn't matter if the software is being developed in-house or by external parties and customers having no knowledge of the developer or their methods. The solutions coming from the crowd can bring up disagreements.

Either a set of requirements is fulfilled or not, there is always a chance of disagreements rising up regarding certain functionalities or the scope of a task.

4.4.5 Knowledge and intellectual property

Intellectual property is a major concern for crowdsourcing projects in software development. Consequently, leakage of confidential information and loss of advantage was a big concern in the implementation of crowdsourcing in the development of agile software. Sharing every specification and too much detail about a certain task to be crowdsourced may not be easy for an organization because of the security and privacy clauses of the project. The developers, on the other hand, require sufficient detail to proceed with their job and get a hang of what the organization is requesting them to do. Ownership and copyright issues are another prime concern, as the tasks are highly creative and require new methods for efficient implementation and even development in the coming future. Thus, organizations want to be completely sure that they can claim as theirs and patent potential inventions that emerge during the whole development process with absolutely no perplexity over ownership.

4.4.6 Motivation and Renumeration

The success of a project in software development is achieved with motivation. Motivation can be physical (environment) or mental like a sense of achievement. The profit of a crowd-sourced task relies upon the complexity and duration of the task. Tasks could be simple such as a simple coding competition or complex like hackathons and competitions spanning over 4 weeks. There is a reduction of cost to a greater extent as the professionals are being replaced by expert crowd very economically, but, the problem here is to set-up an appropriate price for a particular task in crowdsourcing or even in general software development specifically.

4.5 Conclusion

The last decade has witnessed crowdsourcing as an emerging practice by organizations hoping to take up new ideas and solutions from the crowd. Crowdsourcing has been a popular topic of discussion among scholars which are very keen to investigate the practice thoroughly.

With crowdsourcing emerging rapidly, it can be discussed that in terms of agile software development, the applicability in future will spread over the following spaces:

- Segmentation of crowdsourcing processes, i.e., professional organizations will come up and occupy the market for each subtask which might create a monopoly.
- Strictness and boundations will be a priority for specifications with crowdsourcing being applied to various fields of software development.

- Attraction of large developer audience towards the method due to lack of boundations like deadline pressure, which will surely enable the developers to do their best and produce quality solutions.

CHAPTER 5

AMPLIFYING STRENGTH OF THE AGILE SOFTWARE BY USING CROWDSOURCING

The agile system has developed as a solid swap for Rapid Application Development (RAD) software programming building methodology. As the complements enlarge the disperse quality related with the Daily Scrum additionally increments with acting inconveniences in the accompanying emphasis. This builds up the benefit use to the extent that execution starts to slump at a wild pace. The developers have tried to devise a structure that helped the deft approach with crowdsourcing. The effectiveness of the agile process has been essentially upgraded as the requirements of the agile strategy are currently scattered on crowdsourced deal sort out through the favorable circumstances required like CPU time, memory, battery or power are given by moderately unending gathering professional.

5.1 Introduction

To agile technology development science, crowdsourcing spreads significant ideas. Crowdsourcing was implemented as a powerful technique to performance testing, GUI testing, and even as a way to deal with people in coding model empirical tests. There is a broadening level of regard for social affiliations and systems inside agile software designing research, close by crowds are an essential bit of this. This chapter especially focused over utilizing crowds as an elective sort of sourcing, developing it from different structures, for example, internal sourcing, opensourcing, and standard software outsourcing. In a manner of speaking, by what

means can a crowd or 'crowd workforce' successfully add to transmute of a software framework? Abundant research has concentrated on mostly accommodating crowdsourcing stages, for example, Amazon's Mechanical Turk (AMT). In any case, by no examination exists on crowdsourcing agile programming improvement, instead of the reason for crowdsourcing in an all the more wide sense. This chapter portrays there is noteworthy potential in programming advancement through crowdsourcing, at any rate that much research is required to all the more probable see how to ideally do this. This chapter also shows the examination custom for our legitimate examination on crowdsourcing agile software development. While the lean methodology may appear as though something that is essentially relevant to new companies, the innovation business is stacked with instances of associations utilizing improvements in items and iterative progression as development perspective. Google is one of the better-known instances of an association that is determinedly fixated on accentuation through examination. Looking Google makes things and manages the system for development is illuminating. The research work by Stolee and Elabaum [148], taken a gander at the upside of crowdsourcing as a framework to talk that oppose by helping in subject selection. Further, by help of his examination, can reveal how study can be master underneath establishment that attest the nearness to characterize a generous base of vocations and besides yield managing those customers while the examination is running out. Vaish [157], has demonstrated an exploration course which analyzes the reasonableness of master outsourcing by consolidating guide with understudy crowd. This method will allow guides to on the other hand utilized administrators. We are motivated by Humberman [75], analyzed an extensive informational index from YouTube that the potential showed in crowdsourcing gives a solid belief on mindfulness, made by number out of downloads. Research work by Yuan et al [168], pro-

pounded a deft improvement strategy of administration arranged programming. Narock and Hitzler [117], demonstrated dependable arrangement actualizing seeks calculations with crowdsourcing. They watch Big Data inside the geosciences and characterize coherent inquiries identified with the merger of crowdsourcing. Popli and chauhan [128], centered on agile software development exploration work and agile estimation. Supported a method for the real cost, to maintain a strategic distance from the issues of current coordinated practices. Al-tameem [8], proposed a few strategies, in which agile ideas has been, turned out to be persuasive in programming development. It additionally portrays the focal points and the confinements of agile technique. This examination spurs designers, to accept this system, to create programming that ends up being a solution for their evolving necessities. Kumar and Bhatia [91], settled the centrality of this strategy, regarding its quality inside the social structure. Rao et al [132], executed with the genuine aims of look at and pick up keenness. Into, the most recent light-footed ideas and techniques, identifies the strengths and limitations of organized plans and their materiality. Wu and Tsai [166], assembled on programming crowdsourcing and shorten real lessons, adapted, at that point examined two Software crowdsourcing forms, including imaginative ways. Rai and Dhir [130], elucidated the impact and examination of a few conventional systems and another procedure. Jamaalamadakai and Krishna [79], accentuated on couple of encounters with agile->scrum and offers vision to the client whether the agile is wonder drug. Hneif and Ow [71], exhibited their survey more than three agile methodologies including extreme programming, agile modeling, and scrum recognizes them and guidance, when to utilize them.

The association starts by asking for contemplations from a broad assortment of partners; these thoughts are then situated using different criteria. Once a thing enters the accentuation

phase, it is made by a small group that plans to dispatch a product as appropriate on time as possible remembering the true objective to get customer input and make it further a considerable measure of this feedback is overcome computerized examination of customer collaborations. Google is an immense customer of A/B testing (also known as split testing or bucket-testing), a strategy whereby two varieties of an item, A and B, are erratically attempted on customers. With A/B testing, changes to a page or item setup can be attempted against the present layout. Evaluating the impact of even minor changes empowers Google to ensure that the items it makes have the most amazing possible levels of transformation and commitment. Clearly minor complexities like the establishment shade of a site have been found to have any sort of impact to customer change. This A/B testing concept isn't constrained to sites. A/B testing is used comprehensively to trial particular evaluating frameworks, collaboration methodologies and displaying to tailor messages to grow their profitability. A/B testing is an investigative methodology that affiliations use to tailor the impact of their items and organizations. Another methodology that online associations, including google, use to breakdown their items is customer following, a procedure whereby the beginning of a customer's association with an item, the experience which customer encounters while using an item, and the objective the customer heads to a brief time later are altogether put away (secured) for examination. By dismembering designs transversely over various customers, legitimate instruments can intuit customer relationships, objectives, and tendencies to make a particular move. The majority of this useful data would then have the capacity to be utilized to tailor a customer's inclusion to support a final product that examination has demonstrated is likely. Finally, agile companies use assessment examination as an interpretive device. Estimation examination uses trademark lingo planning, content examination, and semantics to perceive

abstract information from a person's cooperation's. Once in a while called data mining, strategy is used by development associations to tailor the customer experience to the customer's tendency and idea. Final examination is in like manner separated, assessed, and sold to various contributed people that need to measure state of mind patterns

5.2 Crowdsourcing Innovation: Agility from the outside

In a circumstance in which the making of isolated improvement is hampered by the extending homogenization of products (caused, to a constrained degree, by globalization), crowdsourcing is gradually being utilized as a way to amplify the speed of progression and enunciate new products and services beneficially and feasibly.

Crowdsourcing is the demonstration of getting product ideas by asking for alms from the massed public rather than just from specialists or suppliers. It's a fundamental idea, but it is at times arduous conception for existing enterprises to adopt, as they are generally anxious over IP ownership.

As the expense of research raises and the capability to pass on research quickly from inside the organization is decreased, crowdsourcing is continuously look as a smart direction, notwithstanding for the most traditional organizations.

Gamble and Proctor, the multinational shopper products organization, set out on a crowdsourcing project when it expected to figure out how to print pictures onto Pringles cans. P&G's inquiry driven it to a little Italian bakery shop that had figured out how to print pictures onto baked goods. P&G authorized the innovation and could roll the idea out to the commercial center in less than a year. This methodology toward R&D was compelling to the point that P&G significantly expanded its crowdsourcing activity and now utilizes crowd-

sourcing for 50 percent of the developments it conveys to market. The P&G move has been mirrored by others in the buyer products space. Johnson and Johnson, Clorox and 3M, all have crowdsourcing activities set up. Next instance is Kimberly-Clark, which picked up a 30 percent decrease in the time it took to present a new product using crowdsourcing. The figure 5.1 portrays the connection between Requester, Platform and Worker segments of crowdsourcing.

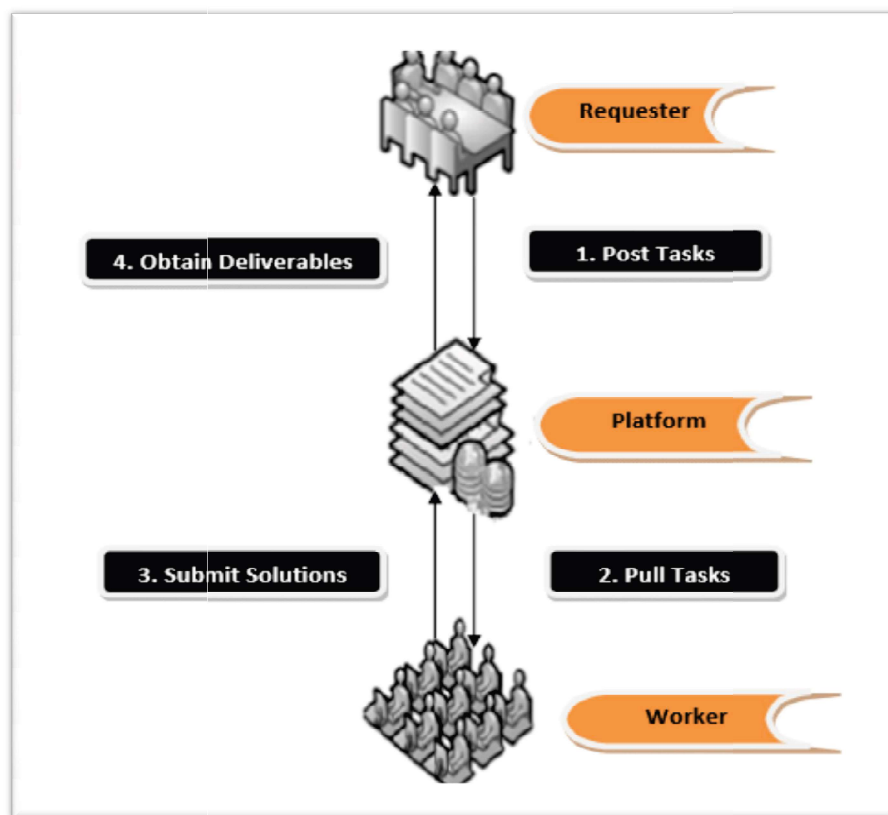


Figure 5.1: Crowdsourcing Components



Figure 5.2: Classification of Crowdsourcing

5.3 Iterative Crowdsourced Software Engineering

Most of the latest approaches to handling crowdsourced software engineering include a lone team mode, using crowdsourcing to handle a given overall mission. Seeing the cascade as a template is incredibly interesting, making such a fast comeback as a sensible concept that supports a lot of crowdsourced development research. This examine might be brief; as it generates; crowdsourced software engineering will most probably wind up resilient and iterative to design the core software engineering frames it facilitates even more likely. Multi-crowdsourced software engineering is a conventional iterative process whereby each team responds to the possible outcomes of various group activities and affects them.

5.4 Research Work

5.4.1 Overview of Scrum

Scrum has a place with the group of agile software development strategies that have pulled in critical consideration among software professionals amid most recent five years. Though the outrageous programming technique that has been generally acknowledged as a standout amongst the most essential agile approaches has a distinct programming flavor (match programming, coding guidelines, test driven advancement, refactoring, continuous integration), SCRUM focuses on arranging software projects.

Scrum introduces that Software development is excessively perplexing and erratic, making it impossible to be arranged precisely ahead of time. Alternatively, it is important to attach experimental process control to ensure transparency, analysis, and transition. For particular, the distinctive environmental and technological variables, time span, performance, requirements, resources, application developments, and methods, including improvement strategies, need to

be constantly monitored with a specific end goal in order to answer to changes. This is done by means of an iterative and gradual method of progression.

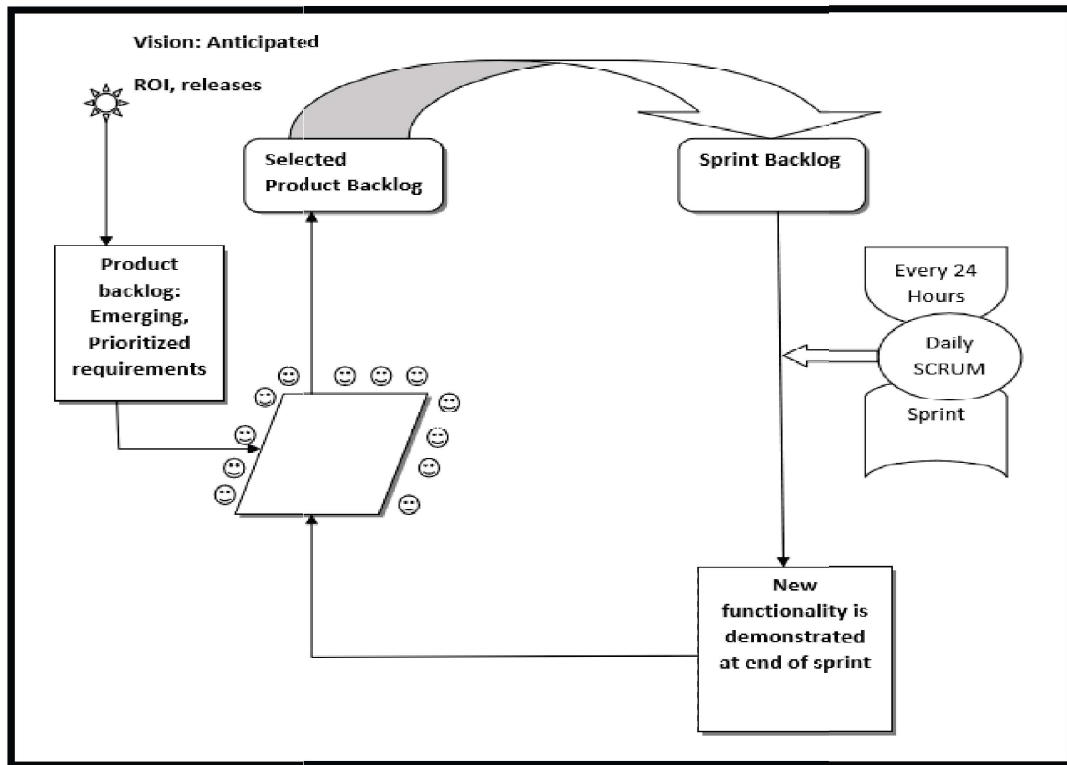


Figure 5.3: Detailed Scrum flow

5.4.2 Efficiency and Velocity in Agile Development

The collection of actual factual data during most of the task is the only practical approach to quantify the competence of the project and agile procedures are linked to the collection and description of this concrete information, making it apparent to the entire association so that there will never be any inquiry where there is an enterprise classification. One of the main real data measurements of agile methods is the efficiency of the project, the estimate of the project that the group will achieve in a prescribed time / incarnation period. Using the last

speed of the project, the community will predict the pace for the next iteration. By also evaluating their velocity patterns, you will be able to settle on some smart data-based decisions while graphing the team's velocity for the task. For helpful hints on offering the most exciting opportunity to your agile team to achieve a fast, stable pace.

Nonetheless, even with all the considerable data velocity graphs give, I ended up needing something more than simply the velocity outline. When resource days are restricted due to holidays, holidays or sick days, the teams ' performance can fall completely. It comprehends course:

Less working days = diminished work finished for that iteration.

In any scenario, how could we prove that the fall in speed was due only to the estimated amount of resource days? How could we show that the group worked as hard since they would in case they were fully staffed for the whole iteration?



Figure 5.4: Velocity and Efficiency in agile methodology

$E = V/R$ [<http://www.adoptionofagile.com/2009/07/efficiency-must-have-agile-metric-as.html>]

E=Efficiency for the iteration

V=Velocity for the iteration

R=Total number of person days worked on the team for the Iteration

Ideally, we have as of now track velocity (V) for finished iterations and furthermore need to begin following aggregate real individual days (R) work for your finished iterations too. For instance, if there is a team of 5 and iteration is 10 days, at that point the most extreme conceivable R esteem for team would be $5 \times 10 = 50$ days. The real R esteem would along these lines be the most extreme conceivable R esteem for your team minus the number of absent days per colleague amid the iteration (e.g. $50 \text{ max} - 2 \text{ sick day} = 48$).

5.5 Crowdsourcing in agile project

In crowd sourcing an agile project mainly involves:

- Requester
- Platform
- Crowd

The requester is itself the SCRUM entity that deals with the agile process development.

Here the iterations, similar to a chronograph are intermittently doled out to variable and fluctuating crowds. Crowds are sources to some sort of computational resources that can be CPU cycles, memory, battery power or something different that in any capacity can be pulled from

the crowds. The thought is to meet few or a considerable lot of the necessities to be met by the SCRUM in light of the fact that ‘Sprinting’ in agile software development requires numerous iterations and when the project load surpasses the most extreme limit of the development Servers, it turns into a no choice alternative else to crowdsource the project.

5.6 Derivation of Agile efficiency by Crowdsourcing

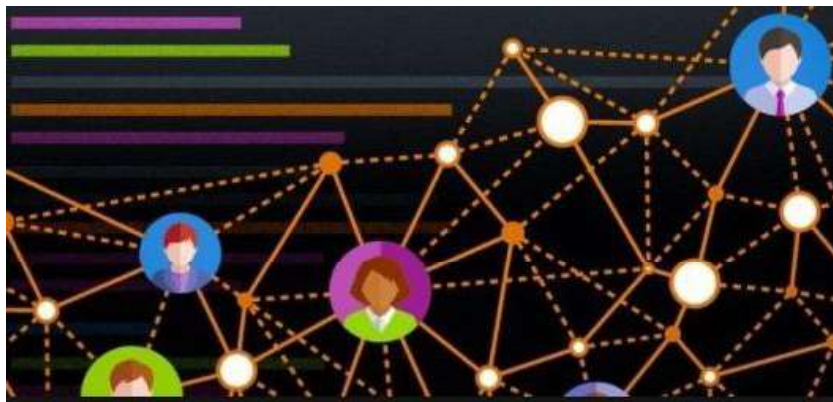


Figure 5.5: Crowd Sourced Agile product

The figure 5.6 demonstrates a depiction of a crowd sourced agile project. The network is a blend of people and machines which can be cell phones, laptops, hand held gadgets or some other resourceful object. By taking a gander at the splendor of the conjugated round nodes and their interrelatedness, it winds up evident that there source use in crowd is auto adjusted. This implies everything moves and waves as indicated by the power of the resource highlight required.

The ‘workers’ go back and forth as per the ISP of that zone.

Assume, there are N numbers of ‘Workers’ in a crowd at one moment of time processing a single monolithic request R,

$p_1, p_2, p_3, p_4, \dots, p_n$ represents processing times used by W_i worker.

Where

$$i=1 \rightarrow n \quad \dots\dots\dots 1$$

$$\text{Eff} = \sum p_i / N \quad \dots\dots\dots 2$$

5.7 Conclusion

The chapter shows the updated efficiency of the Agile Development Process by using the crowdsourcing thought. The contortion and exponential terrorizing of the necessities from the customers is treated with the frequently growing and every so often settled or invariable transport of asset qualities like CPU time, memory, control and diverse attributes.

CHAPTER 6

PROPOSED METHODOLOGY FOR CROWDSOURCING AND AGILE DEVELOPMENT

In this chapter, we proposed a phenomenon for 'crowdsourcing' in light of consolidating these strategies to agile development in software engineering. The entire software development life cycle miniature for crowdsourcing is produced and attempted to be clubbed with agile. This is done with a specific end goal to accelerate the development process of agile. Agile responds as a rapid development of applications (RAD) based on the philosophy of software engineering. The crowdsourcing process's 'workforce' is assigned separately to the positions of agile strategies employed on agile tasks.

6.1 Introduction

- Throughout this study, we introduce a framework for the use of crowdsourcing in agile software development in which crowdsourced specialists carry out the deployment process
- Crowdsourcing has different potential benefits over the existing techniques.
- Speed to market quicker due to simultaneous implementation and the ability to quickly scale up the group.
- Low cost because of access to the correct ability or right talent
- High quality software because of the imagination and the focused nature of the individuals from the group.

Such a framework incorporates many open source software development techniques, including a basis for splitting a complicated business request into team (crowd) transmission-friendly nuclear (atomic) programming tasks. All tasks can be completed within a specified period of time, are largely independent, and to frame the entire application, the response to all tasks can be tested and incorporated.

The proposed methodology is as per the following:

- Skilled experts design Hierarchical Component Architecture (HCA) on the practicality of the process using the concept of Work Breakdown Structure (WBS).
- Constituents of the hierarchical component engineering are indicated as contracts comprising of test cases, interfaces and a textual portrayal. Ideas of design by interface, test driven advancement and dependency injection are utilized here.
- Attributions are made to the team to enforce the sections from the contracts.
- The programmed validation is conducted using the internet ontology language as the team (crowd) presents the script.

Crowdsourcing agile software development commonly includes three kinds of actors: specialists (workers), who take an interest in developing software; clients, who have software development work that should be done; and stages (platforms), who give an online market place where clients and specialist can meet. Every one of these three actors will have an alternate point of view on crowdsourcing agile software development. We briefly examine them underneath.

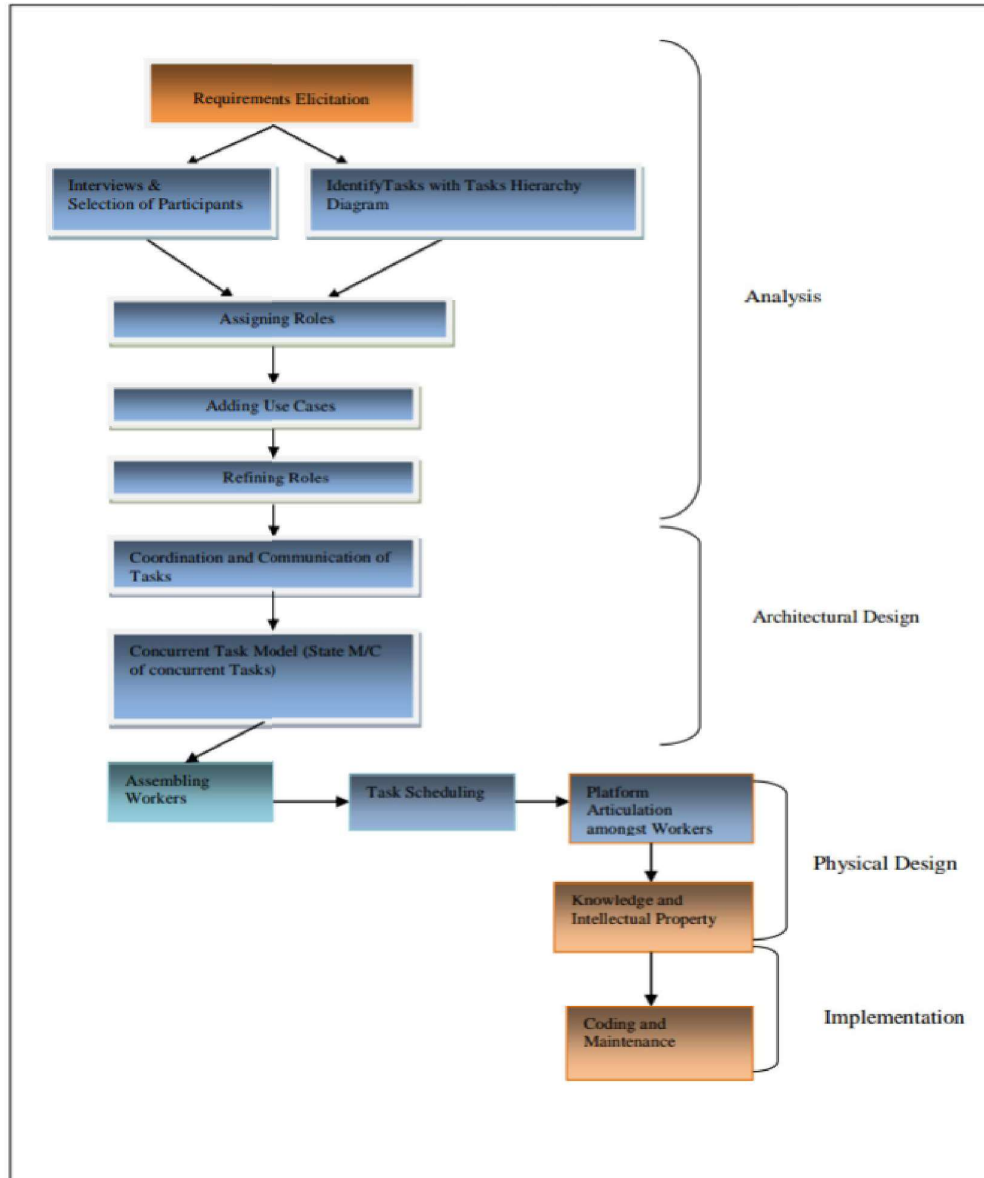


Fig-

ure 6.1: Proposed Methodology

- **Clients**

Clients, or participants, are groups or persons with code assignments (tasks) which must be completed. Clients may have unique crowdsourcing inspirations; for instance, interestingly, a

business will need to increase its workforce. An affiliation can turn out to be very adaptable in scaling up and down their production capacity by 'outsourcing' a portion of the work to the 'group.'

- **Workers**

Employees are the people who are doing the work— they are making the pieces of software that a company outsources. 'Who these employees are, and there are many open investigations which would be of great interest, is hardly thought of. Some fundamental research on the motivation of workers has been done, but we need a superior understanding of why participants are taking part in this form of 'insecure' study. However, while 'common' occupations of growth normally require specific levels of formal education, little is known about the foundation of specialists in crowdsourcing. Different contemplations incorporate the unwavering quality of specialists, both in form of clients having the capacity to deliver items in an auspicious manner, and additionally watchful consideration of product particular information and IP.

- **Platforms**

Stages (platforms) give an online 'commercial center' for workers and clients to meet. The biggest stage for crowdsourcing agile software development is Top Coder, with software professionals of something beyond than 600,000 as of January 2014. Be that as it may, there have likewise been signs just a little part of its enrolled clients are really taking an interest in software development. Stages (platforms) may have unique participation models. Top Coder for example utilizes a competition based model— in Top Coder's model, the champion gets paid a certain.

6.2 Crowdsourcing Methodology (CRM)

Agile software development is not any more a typical practice among small groups of software developer professionals, however progressively takes place in associations and specialty including various people. Crowdsourcing, a form of procurement that grows in the 1990s, is an innovative idea to get work done. Because of the boundless chances of the World Wide Web (WWW), associations can take advantage of a workforce comprising of anybody (even remotely separated) with the help of an Internet connection.

Software development applicants or clients can advertise work fragments, activities or functions on a crowdsourcing stage, where providers (i.e., individual specialists) select those commitments or tasks that fit their capabilities and interests. Crowdsourcing have been received in a wide assortment of domain: social context, education and medicinal services; yet for this examination, we will lay accentuation on crowdsourcing in agile software development.

Crowdsourcing is the concept used to combine the terms "outsourcing" and "crowd." Crowdsourcing is first described as a modern web-based business strategy that saddles through what adds up to an open call for proposals the creative ideas of a decentralized network of people. This definition is very obscure. A second more broad meaning of sets that "Crowdsourcing presents to the demonstration or act of an organization or foundation taking a software work/duties once performed by representatives and outsourcing it to an unknown workforce, topographically dispersed network of people with various abilities as an open call". This can appear as peer development (team work or collaborative), however it is sometimes often at-

tempted by sole people. The major requirement is the use of the open call system and the extensive network of potential employees." This second definition fits superior to the first, since it states clearly that the activity can be performed cooperatively, yet in addition separately and that an organization utilizes an open call to achieve the substantial network of potential specialists.

To construct the initial methods of CRM, 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 achieve and solutions are the bottom up builders for the final product. For example, taking Research Journal Paper Submission Process (RJSP) as the case study, first the Requirements are elicited. Then the process of Analysis begins that incorporates Interviews and selection of Participants. The participants later are assigned to workers. A task hierarchy diagram involving minsky's model of task reduction. After task reduction is done, roles are assigned and use cases are elaborated. Roles are further refined in that the specific task is allocated to specific role(s). The process of design and implementation of CRS then proceeds ultimately converging to the final crowd sourced product. We have extended the concept of Alam and Campbell [7], discovered an interpretive contextual investigation research of a non-benefit crowdsourcing activity inside the Australian Newspapers Digitization Project (ANDP). Amrollahi et al [11], presented an in-advance contextual investigation of utilizing the crowdsourcing model to execute the open methodology idea in an Australian college. Amini et al [10], analyzed about the expansive and developing number of portable applications. Imagined benefit expands on crowdsourcing, virtualization, and computerization to empower substantial scale examination of applications. Anderson [12], outlined the real difficulties that obstruct to successfully consolidate crowd sourcing into existing business con-

dition to develop another type of participation. They lay out research challenges in most vital territories: the procedure plan and check outline of run of the mill examples to help business portions finished by nonconcurrent swarm contribution through unidentified accomplices, improvement of work processes administration framework usefulness to suit swarm sourcing exercises. The article by Brabham [22], contended that the crowdsourcing model, an effective, Web-based, disseminated critical thinking and generation display for business, is a fitting model for empowering the subject interest process out in the open arranging ventures. Bücheler and Sieg [26], examined the relevance of crowdsourcing and a few strategies from Open Innovation to the logical strategy and fundamental science in a non-benefit condition. Chen and Liu [35], found a U-formed connection between the accommodation time and winning in the two kinds of challenges. Chen and Luo [34], reported a preparatory report on crowdsourcing testing for instructive ventures. They present three business-programming items as instructive testing ventures, which are crowdsourced by our showing emotionally supportive network. Geiger et al [58], centered only around an authoritative point of view and on the systems accessible to these associations. Graber [62], showed that the crowdsourcing model of research can possibly make hurt members, controls the member into proceeded with support, and uses members as exploratory subjects. Hetmank [69], endeavored to pick up a superior comprehension of what crowdsourcing frameworks are and what regular plan angles are considered in the improvement of such frameworks. In this paper, the creator led a deliberate writing audit in the space of crowdsourcing frameworks. Metropolis Model of Kazman and Chen [83], depicted the standards encompassing how such frameworks may be made and supported. Kittur et al [84], explored the utility of a smaller scale undertaking market for gathering client estimations, and talk about outline contemplations for

creating remote miniaturized scale client assessment assignments. Peng et al [124], consolidated best practices from open source improvement furthermore, outsourcing, crowdsourcing influences and animates vitality toward disseminated esteem creation. Through reviewing the existing research environment, Zhao and Zhu [171], introduced a simple analysis of the crowdsourcing research system, including hypothetical establishments, inquire about techniques, and research foci, and distinguishes a few imperative research bearings for IS researchers from three points of view - the member, association, and framework - and which warrant additionally think about. Zogaj et al[172], demonstrated that testCloud faces three fundamental difficulties, these are: dealing with the procedure, dealing with the group and dealing with the innovation. For each measurement, they diagram instruments that testCloud applies for confronting the difficulties related with crowdsourcing ventures.

6.3 Analysis

6.3.1 Requirements Elicitation

It becomes an unavoidable activity to elicit the user requirements before moving to the next steps of the CRM. Interviews, level of participation and platform on which the CRM has to run are brought on paper. The process consists of

- Requirement Discovery
- Requirements Classification and Organization
- Requirements Prioritization and Negotiation
- Requirements Specification.

These steps are shown in the figure 6.2.

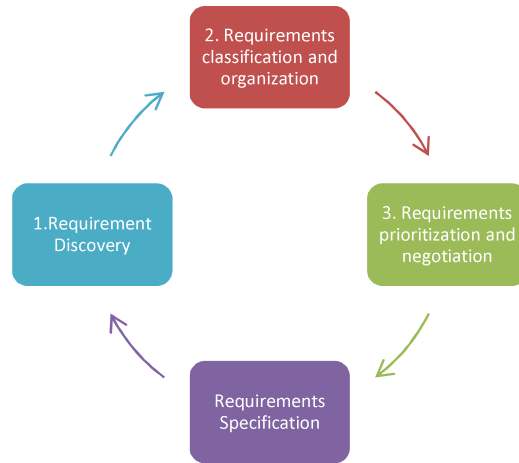


Figure 6.2: Requirement Elicitation in CRM

6.3.2 Identify Tasks with Tasks Hierarchy Diagram

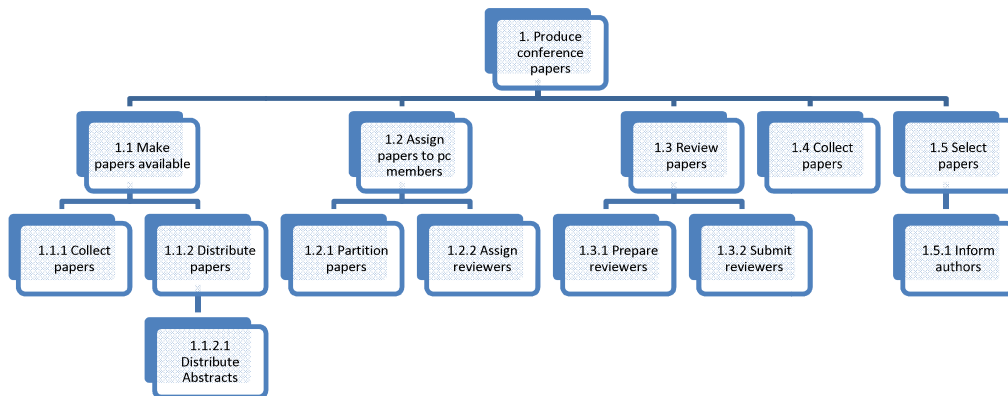


Figure 6.3: Tasks Hierarchy Diagram

6.3.3 Assigning Roles

The transformation of goals into roles in the general case is one - to-one, with each goal being mapped to a role. However, there are situations where it is useful to have a single role, including convenience or efficiency, to be responsible for multiple goals. Below is a mapping of the objectives of our previous example to a series of functions.

- PaperDB
- Partitioner
- Assigner
- Reviewer
- Collector
- Decision Maker

6.3.4 Adding Use Cases

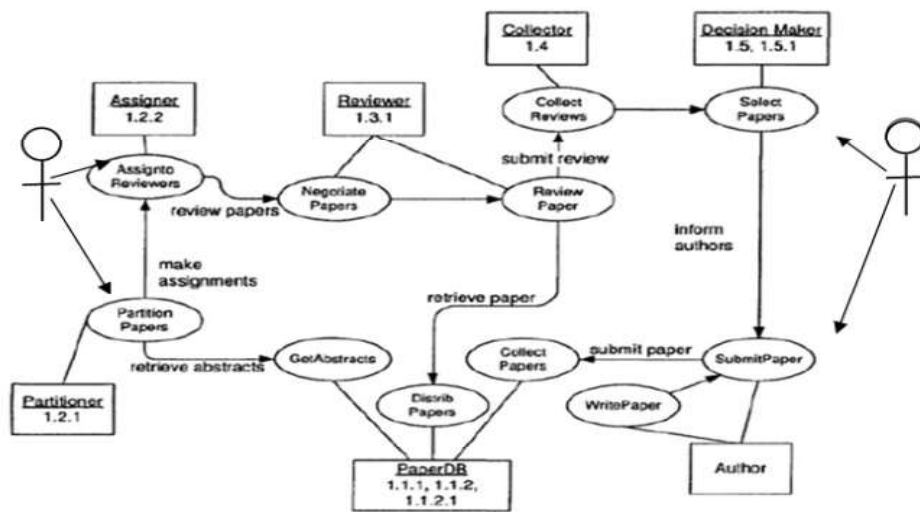


Figure 6.4: Use Case Development

6.3.5 Refining Roles

Roles are refined in this stage. There are already many roles, which are application specific and have been assigned to entities. Now in this stage grouping of roles in the order of similarity and action, identified roles are refined in groups.

6.4 Design

6.4.1. Architectural Design

6.4.1.1. Coordination and Communication of Tasks

The tasks allotted to the worker are streamlined and prepared to be distributed to them.

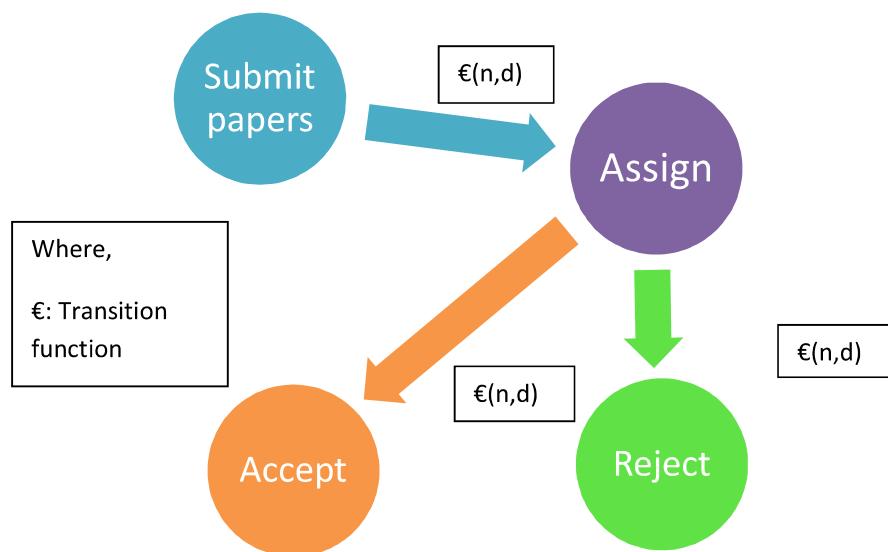


Figure 6.5: Concurrent task model

6.4.2 Physical Design

6.4.2.1 Task Scheduling

Task Scheduling is done by extracting meaning from the state machines. Tasks are assigned to the workers in accordance with the roles. Platform Articulation amongst Workers is what the basic need of an agile project.

In crowd sourcing an agile project mainly involves:

- Requester
- Platform
- Crowd

The requester is itself the SCRUM entity that manages the agile process development.

Here the iterations, like a chronograph are periodically assigned to variable and fluctuating crowds. Crowds are sources to some kind of computational resources that can be CPU cycles, memory, battery power or something else that in any way can be pulled from the crowds. The idea is to meet few or many of the requirements to be met by the SCRUM 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 no choice option else to crowdsource the project.

6.5 Implementation of SCRUM in Crowdsourcing

Scrum belongs to the agile software development methodology family, which over the past five years has gained significant attention from software practitioners. Even though the Extreme programming process, widely understood as one of the most important agile methods,

has a definite programming texture (pair programming, coding standards, test driven development, and continuous integration), SCRUM focuses on project management technology.

Scrum commences with the idea that the design of technology is too complicated and unpredictable for preparation. Alternatively, to ensure transparency, examination, and adaptation, scientific process management must be implemented. The various environmental and technical variables (such as time frame, efficiency, requirements, resources, implementation of technology and equipment, and even design methods) must also be constantly monitored in order to be able to adapt flexibly to changes. This is achieved through an iterative and incremental development process.

6.6 Coding and Maintenance

OSS (Open Source Software) like Python is used for coding. Python 3.2.3 is preferred with DJANGO as front end and SQLite as backend. Security is provided by user / role basis.

6.7 Conclusion

In this chapter, a model has devised for agile software development using crowd-sourcing. This is particularly done in order to act the software development life cycle phases through workers in crowdsourcing. A methodology has been designed for the same. Agile is based on rapid application development, so in order to increase the velocity of the development of any agile product, crowdsourcing is utilized. As future scope efficiency and velocity of agile software development is aimed to be maximized.

CHAPTER 7

has a definite programming texture (pair programming, coding standards, test driven development, and continuous integration), SCRUM focuses on project management technology.

Scrum commences with the idea that the design of technology is too complicated and unpredictable for preparation. Alternatively, to ensure transparency, examination, and adaptation, scientific process management must be implemented. The various environmental and technical variables (such as time frame, efficiency, requirements, resources, implementation of technology and equipment, and even design methods) must also be constantly monitored in order to be able to adapt flexibly to changes. This is achieved through an iterative and incremental development process.

6.6 Coding and Maintenance

OSS (Open Source Software) like Python is used for coding. Python 3.2.3 is preferred with DJANGO as front end and SQLite as backend. Security is provided by user / role basis.

6.7 Conclusion

In this chapter, a model has devised for agile software development using crowd-sourcing. This is particularly done in order to act the software development life cycle phases through workers in crowdsourcing. A methodology has been designed for the same. Agile is based on rapid application development, so in order to increase the velocity of the development of any agile product, crowdsourcing is utilized. As future scope efficiency and velocity of agile software development is aimed to be maximized.

CHAPTER 7

APPLICATION OF CROWD SOURCING IN AGILE SOFTWARE ENGINEERING TO IMPROVE EFFICIENCY AND VALIDATE REQUIREMENTS USING ONTOLOGICAL FRAMEWORK

Ontological support to software development life cycle provides the necessary seeds for validating the agile requirements. It is necessary to modify the existing agile model so that organizational productivity runs simultaneous to the inherent enlargement in the number of user demands in agile development. The crowd-sourcing of the programming activities to the crowd workforce deals with this increase in specifications or guidelines in agile software development. Sprinting demands agility and so the authors have devised a solution by emulsifying crowdsourcing and agile software development. A semantic approach is accompanied by an ontology called RAgile being developed and preserved. The crowd workforce and agile development continually interact with each other. The requirements that the crowd meets up are first converted to rules in semantic web rule language and then validated by the DROOLS reasoner. The invalid rules are recycled and the valid rules are saved.

7.1 Introduction

The stakeholder model suggested in the emergence is a source of confusion linked to the consumer and their device(s) between internal and external performance. If the QA / QC paradigm is intended to generate metadata on the consistency of cognitive data in the context of ISO 19157 (the quality model of the producer), this method requires certain varieties of quality elements. Table 7.1 provides a summary of the quality components recognized as part of a process of quality assurance, with an emphasis on involved volunteers.

Table 7.1: Table showing various Quality elements and their description

Quality element	Definition
Vagueness	Inability to make a clear-cut choice (<i>i.e.</i> , lack of classifying capability)
Ambiguity	Incompatibility of the choices or descriptions made (<i>i.e.</i> , lack of understanding, of clarity)
Judgement	Accuracy of choice or decision in a relation to something known to be true (<i>i.e.</i> , perception capability and interpretation)
Reliability	Consistency in choices / decisions (<i>i.e.</i> , testing against itself)
Validity	Coherence with other people's choices (<i>i.e.</i> , against other knowledge))
Trust	Confidence accumulated over other criterion concerning data captured previously (linked to reliability, validity and reputability)

Crowdsourcing is a data collection method that has generated previously unavailable data resources and enhanced current ones, but can also vary greatly in performance. This introduces a state where difficulties to potential end clients are worried about the V&V and QA Activities of the information gathered. Having the capacity to evaluate the vulnerability, characterize and measure the diverse quality components related with crowdsourced information, and permits strategies for dynamic task and implementation of agile rules utilizing the idea of blended ontology which is the extent of this paper. Sorts of crowdsourcing range from very composed strategies for saddling the aggregate energy of the group, for instance amazon mechanical turk and other monetary reward based schemes to volunteered geographic information (VGI) such as open streetmap.

We at that point talk about how outlines can be executed utilizing progressed and conventional coordination models. We analyzed that institutionalization endeavor completed by the FIPA methodology technical committee and after that displays the re-see exercises we did beginning from that proposition; these brought about a little unique meaning of probably the most essential components of the approach so as to help a multi-see portrayal of the section (the perspectives are process, reuse, putting away, and execution). The association based multi-operator programming building (O-MaSE) system [40], which coordinates an arrangement of solid innovations went for encouraging. We propose an arrangement of documentations and a methodological structure to help the originator to fabricate MAS utilizing AGR. We at that point demonstrate that it is conceivable to outline multi operator frameworks utilizing just OCMAS models. ASPECS advancement process and gives an arrangement of methodological rules for each procedure action. An entire contextual investigation is likewise used to show the plan procedure and the related documentations. ASPECS utilizes UML as a demonstrating dialect. As a result of the particular needs of specialists and holonic authoritative plan, the UML semantics and documentation are utilized as reference focuses, yet they have been stretched out by presenting new particular profiles.

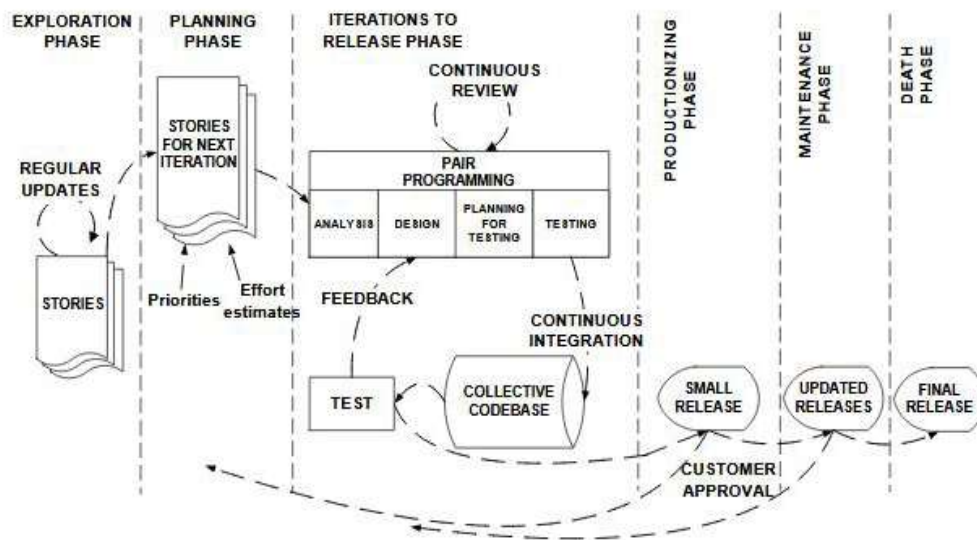


Figure 7.1: Agile Phases with special focus on Pair Programming

7.2 Crowdsourcing Emulsified with Agile software development

The process of crowdsourcing agile software development begins with the construction of work break-down structure, also called task reduction. Now when the work is decomposed into tasks, goals are identified by grouping similar tasks together. It is not always the case that multiple tasks unite to form a goal. A standalone task can result into a single goal as well.

7.2.1 Task Decomposition

Numerous resources are needed in a crowd to build hierarchical software systems to profit from a potentially significant crowd, the project should be separated into several small chunks which various developers in the crowd may build in parallel. This raises an intriguing question in software engineering, namely how to decompose the system into smaller modules

without causing problems in putting it back together once it is developed. Common software engineering concerns relate to definitions, interfaces and dependencies throughout the context of decomposition.

7.2.2 Coordination and Communication

While the primary issue of task decomposition would be how to decompose a project to be divided into manageable pieces of work, the role of coordination is to preserve the dependencies between these tasks. Coordination is important to maintain that tasks are carried out in a timely manner, while at the same time achieving the main objective of building a system.

7.2.3 Planning and Scheduling

For crowdsourcing, simultaneous distribution of software development becomes far more complicated than what can be set up internally or in traditional outsourcing circumstances where production is subject to a bilateral agreement. Faster implementation is a significant benefit of crowdsourcing, as when the job can be split into fewer assignments that can then be done in parallel. As well as, and from the other hand, because it's not possible to kind of speed up crowdsourcing contests until the due date has been set, achieving better customer service is not imaginable to intrude. Fundamental issues in the development of crowdsourcing systems are therefore important to how to ensure timely delivery of a software project when parts are crowded to an unknown workforce.

7.2.4 Quality Assurance

Many proponents for crowdsourcing argue that some applications will be of superior quality due to a large number of responses (from a large enough crowd), thus answering a central concern in software engineering. Also, closely related to the law of Linus, Which is that that a sufficiently huge number of people is likely to be someone who knows how to handle a certain flaw, a similar line of thought would suggest that a wide range of experience is still available in the crowd.

7.2.5 Knowledge and Intellectual Property

Software engineering is a demanding task requiring specialization, and information exchanging and management plays an important role throughout the software development life cycle. A fundamental difference between first-house production and traditional outsourcing cases, on the one side, and crowd sourcing, in the other, is that the latter is characterized by a possibly continuing turnover of employees.

7.3 Problem Statement

Major issues in agile development and crowd sourcing are:

Concern 1: Companies ought to learn exactly what they are looking business rules and procedures and whether they are correctly following them.

Concern 2: Organizations will have to define the business rules contained throughout their information systems in such a way that can also be interpreted by all stakeholders, and need to have a way to make sure traceability among these rule definitions and the rules ' actual implementation.

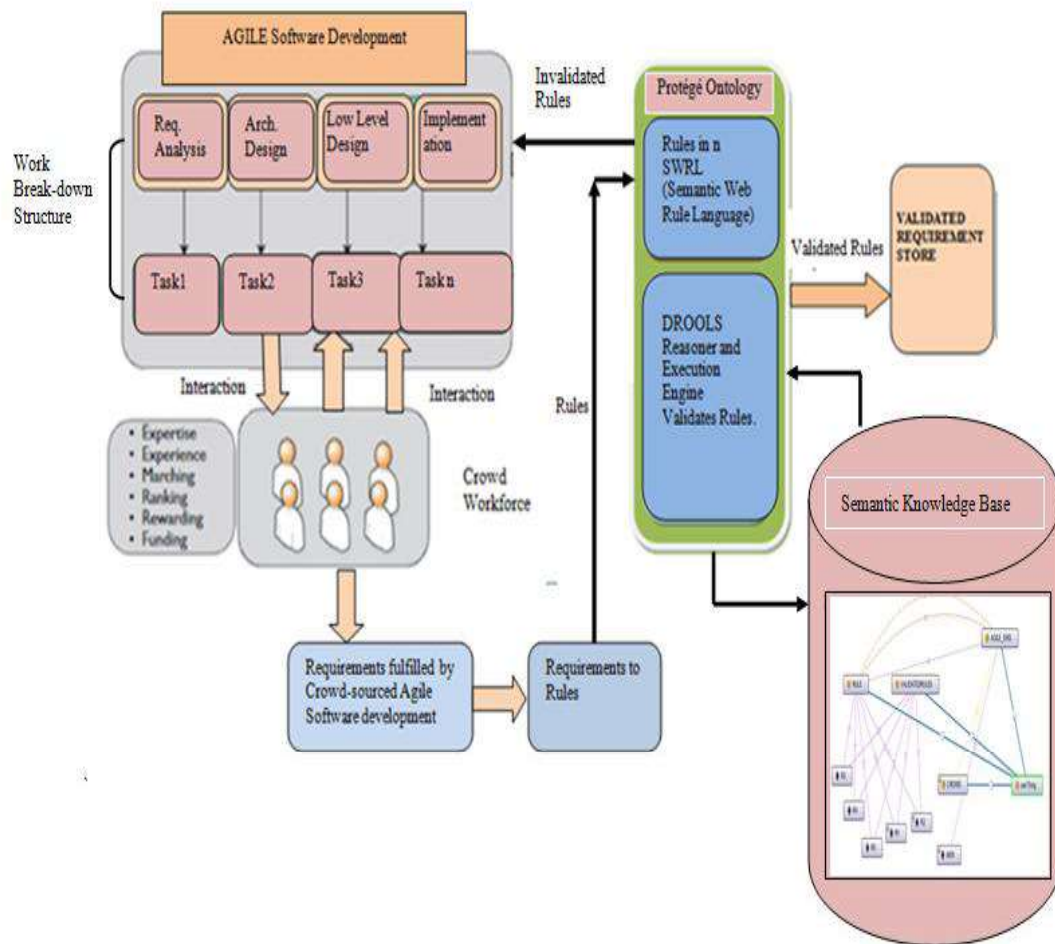


Figure 7.2: Agile in combination with Crowd sourcing

The above figure (Figure 7.2) states that agile development uses Work-break down structure where each and every phase of software engineering model is realized by WBS. It is then send to Crowd sourced based workforce which is governed by the help of Protégé Ontology storing the semantic Knowledgebase. The entire process is supported by DROOLS. The above engine is verified by the help of a Case on Genomics & DNA Sequencing.

7.4 Case Study: Genomics & DNA Sequencing

As a case study for implementing inter-agent communication, DNA pattern search in existing varied and heterogeneous Genome Repositories is chosen.

From the viewpoint of a computer science researcher the important concerns regarding sequencing a DNA are:

The Genome contains the DNA and the whole genetic structure. This genetic structure keeps the complete information necessary for an organism to live its life. This genetic material is similar in many organisms. Biologists and Life Science's experts sequence DNA in the form of sequences of four characters, A, C, T and G. This is done in order to represent a DNA programmatically.

1. Thymine (T)
2. Cytosine (C)
3. Guanine (G)
4. Adenine (A)

A DNA structure is made up of the combinations of these four elements. Since the DNA of an organism is similar to other organisms, conditions arise when biologists look for similarity in DNAs like in areas like pharmacy. There are requirements when a particular extract of a DNA has to be searched in disparate and heterogeneous data sources ranging from plain text files to plethora of databases acting as repositories of fully sequenced DNAs. The full discussion on DNA sequencing is out of the scope of this paper. As a researcher, my quest deals

only with inter-agent communication focusing validated policies and verified message exchange between agents.

The above case study is used to implement the agile rules as shown below:

As part of the architecture design, policies stating the rules that the agents must stick to, are also derived. The policies through requirements stated in this work can be stated as:

***R1:** Maximum Time Limit: 30ns*

***R2:** ONLY A, C, T, or G characters can be used to represent a DNA Pattern.*

***R3:** The search string cannot contain white spaces, hyphens or any special character or numbers.*

***R4:** Max Length of a search pattern can be set but assumed to be 1024 chars.*

***R5:** Break-up size of search pattern can be set but assumed to be 11 chars.*

***R6:** Patterns not conforming to P1, P2, P3 and P4 will be immediately discarded.*

***R7:** Total time from query submission to result display can be maximum 8.00 secs.*

***R8:** Queries from Agile to crowds failing to meet R6 will be held for resubmission.*

This chapter implements the issues discussed above as part of the framework developed by the authors and implements the Agile Rules for crowd sourcing.

Issue1 (Organizations will have to know what business processes they use and whether they regularly use them.) This Issue is Implemented Using Requirements V&V.

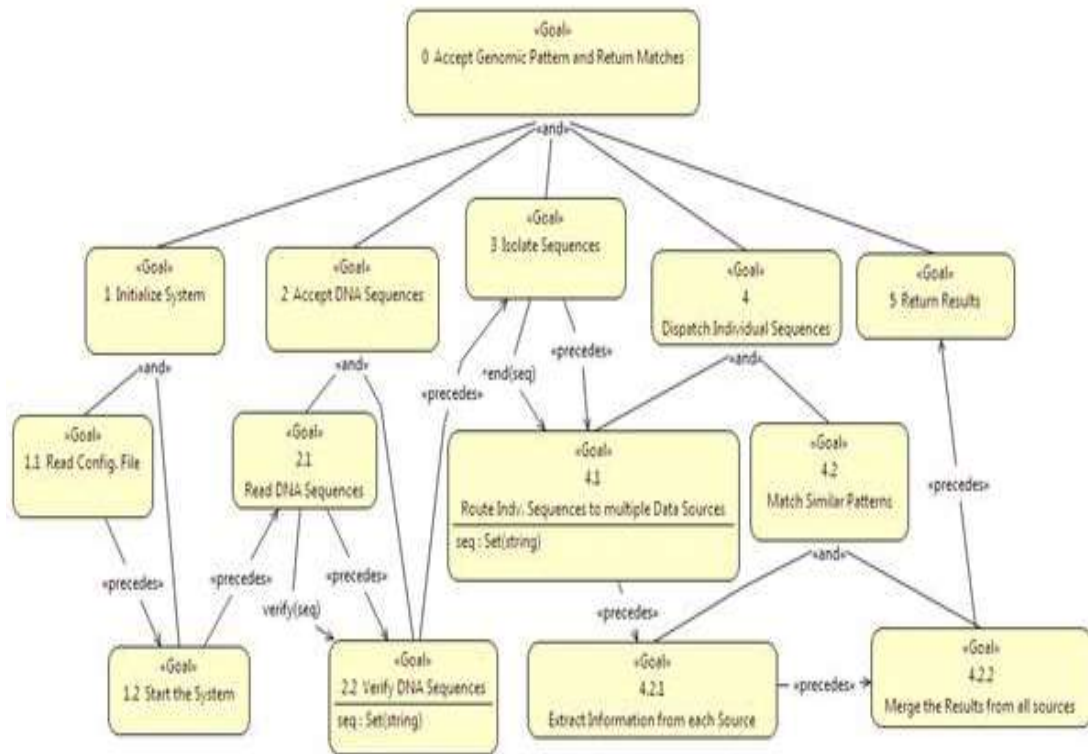


Figure 7.3: Task or Goal Model

7.5 Requirement Validation

R1: Maximum Time Limit: 30ns

$RULE(?R) \wedge hasCPUTIME(?R, ?t) \wedge swrlb:lessThan(30^{xsd:short}, ?t) \rightarrow valRULES(?R, ?t) \wedge VALIDATEDRULES(?R)$

In the above code SWRL formal language is used to test, verify and validate the agile rules/requirements in crowd outsourcing. R, one of the instances of RULE class is testified with any range of valid and invalid rules and then only the successfully verified and validated rules are selected and pushed to the val_RULESproperty in the ontology designed specifical-

ly for this. The above RULE example R1 only considers a single rule that: “One of agile processes need 30 nano seconds as maximum to get executed”.

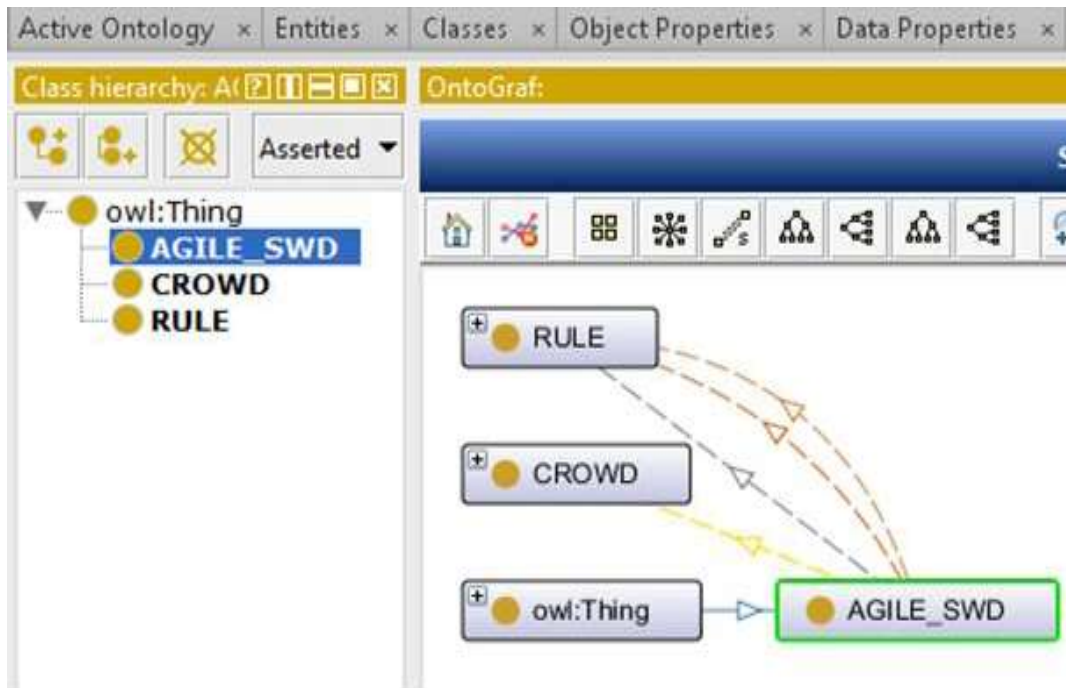


Figure 7.4: Basic Rule of Ontology

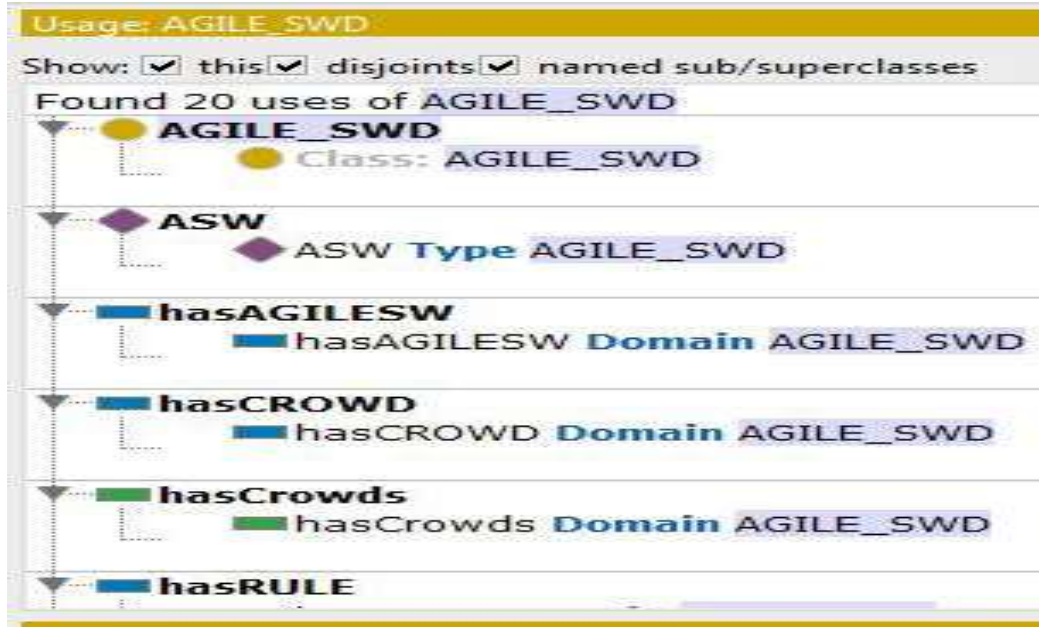


Figure 7.5: Agile SWD Class with basic Rule Ontology

The above ontology is expanded to include Agile Rules and its relationship with Rules. It is pertinent to mention that these Rules are in fact derived from the requirements of the end users and hardcoded in the ontology for validation purpose. These requirements are extracted from the software requirements specification itself and verified first at the manual level then injected as rules in to the ontology for validation purpose. Rule R1 is one of the 8 Rules that are inferred by the DROOLS reasoner that comes along with Protégé 5. The reasoner continuously checks for any discrepancy in the formulated rules. Only the individuals that abide by the Rule Engine Inference mechanism are moved to the VALIDATEDRULES class. This parallel checking is conducted like a chronograph and new rules might be coming on the fly and validated accordingly as in the figure below.

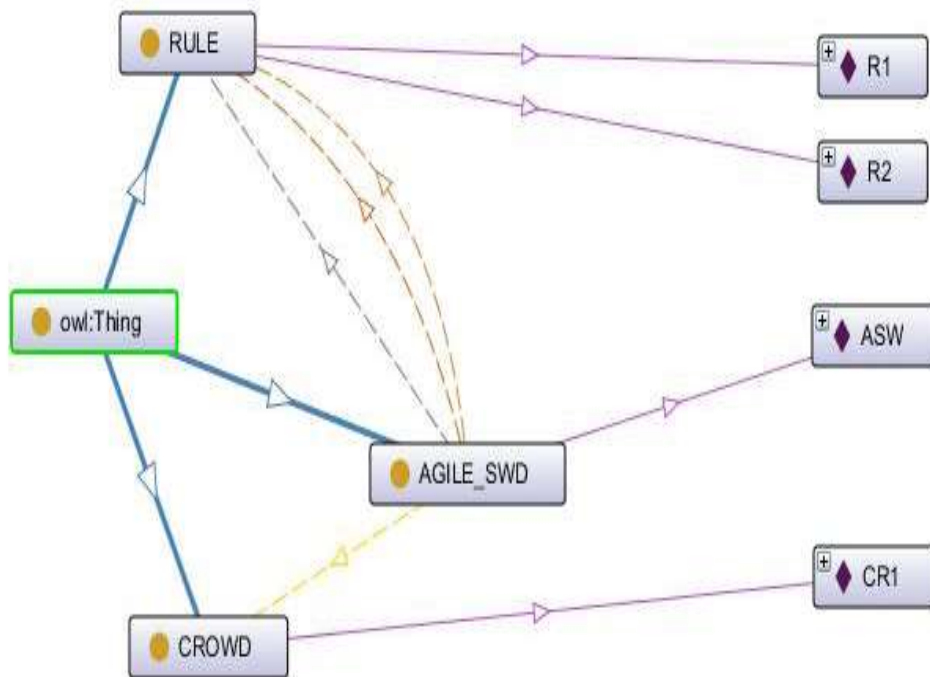


Figure 7.6: Ontology is expanded to include Agile Rules and its relationship with Rules

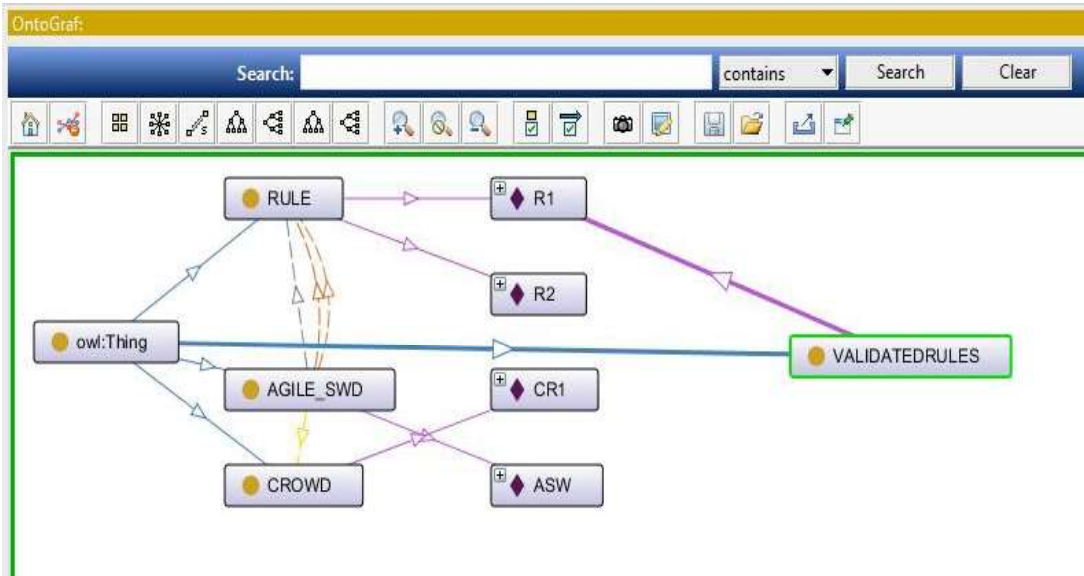


Figure 7.7: Validation of Rules

7.6 Merged ontology

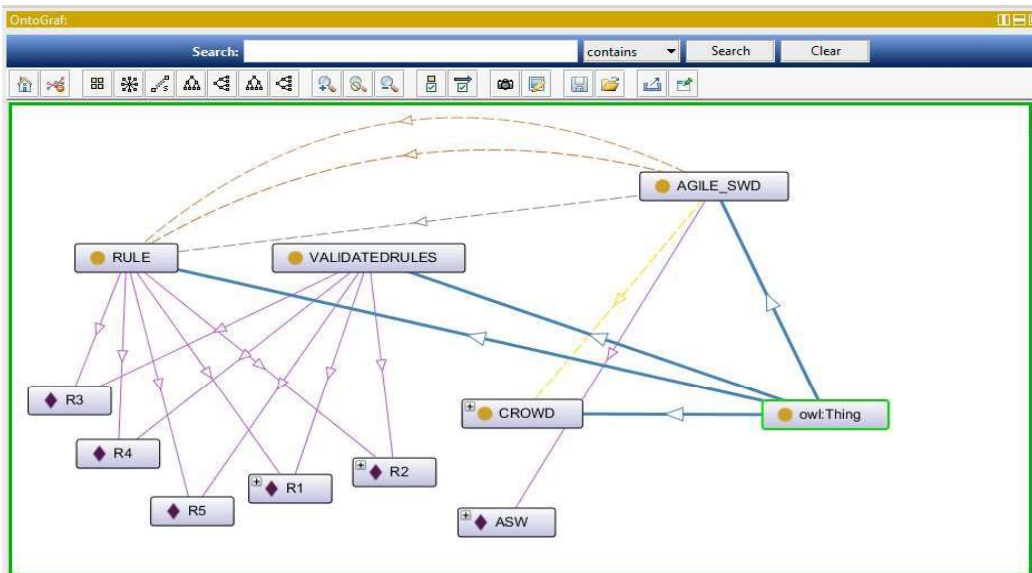


Figure 7.8: Merged ontology with newly discovered and validated rules

Rules represented as requirements are dynamic entries by the software requirements specification (modification) from end user, namely, R3, R4, R5 are added to the ontology. Onto graph is used for representing the relationships between classes, properties and individuals.

7.7 Implementation

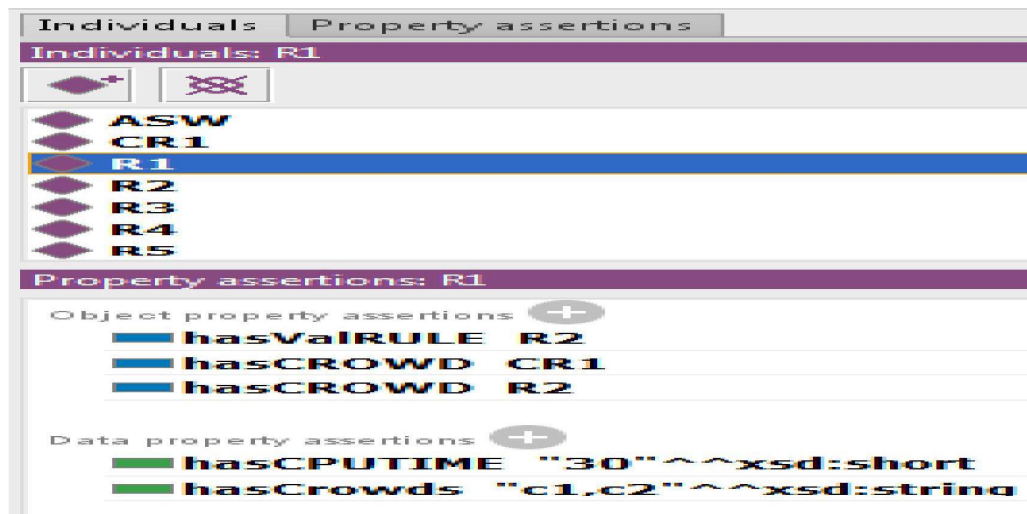


Figure 7.9: Rules Derived from First Requirement

The above snap shows that Rule R1, derived from first requirement does not have “valRule” Property set for itself. This is because the Estimated CPU Time (t) by the Requirement R is greater than 30 ns.

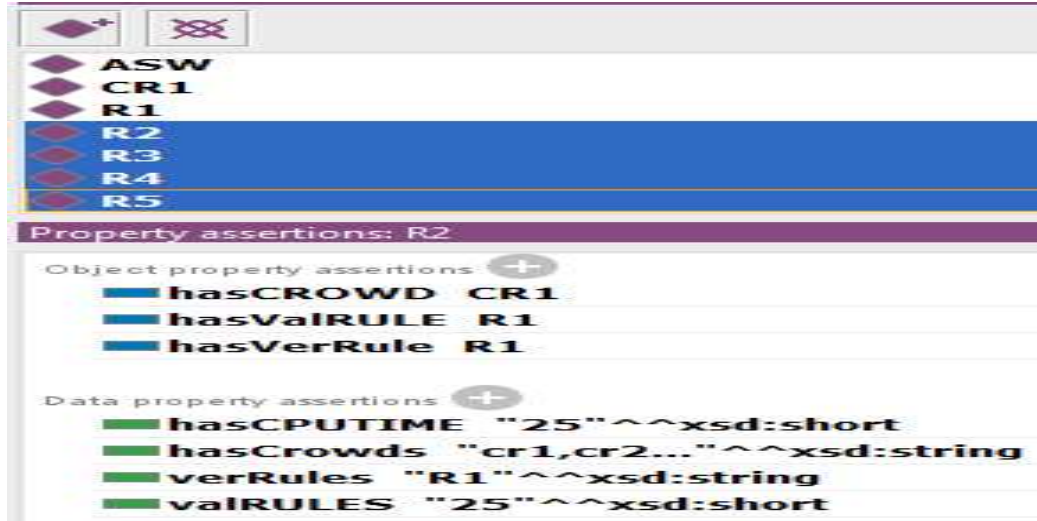


Figure 7.10: R2, R3, R4, R5 requirements / rules and has valRULES

On the contrary, the above figure displays R2, R3, R4, R5 requirements / rules and has valRULES property set because they come under the criteria of t being greater than 30. In short, this implementation selects only those valid rules that have CPU Time less than 30 ns.

7.8 Conclusion

The core concept of this work is to construct a meta-model that with the help of semantics and ontological merging proves to implement, verify and validate the agile rules. DROOLS reasoner with Protege 5.2 is used to model and verify the crowd sourcing for satisfaction and attainment of the rules in distributed environment. Merging of the ontology in different versions of the same original RDF/web ontology language file is done by the Protégé's inbuilt merging tool. As the future work authors intend to work upon context sensitive requirements that are actually manifested by changing the 'Rules' or even deeper the 'Policies' that are the actuation of Rules by building ontology that are sensitive and adaptive to the changing environments.

CHAPTER 8

COMPARISON OF USAGE OF CROWDSOURCING IN TRADITIONAL AND AGILE SOFTWARE DEVELOPMENT METHODOLOGIES ON THE BASIS OF EFFECTIVENESS

This chapter focuses to forecast effort in person months for development of agile software with the use of crowdsourcing compared to traditional (prototyping) software development incorporating crowd sourcing.

DNA Matcher (DNAM) acts like a case study and its development has been done using agile+crowdsourcing and prototyping+crowdsourcing. COCOMO II model is applied in steps to agile development and then to traditional (prototyping) methods. When both methods use crowdsourcing, agile design was found to be considerably economical. The research paper shown here is DNAM, built again using traditional methods of prototyping. Costing is performed during its review. Then, with the crowdsourcing used in parallel to the prototyping method, costing was done. The effort and time in Person Months (PM) was calculated. The agile development methodology is used in the development of DNAM. The crowdsourcing paradigm was attached to the agile process. As soon as the analysis phase is completed, Simple build-up approach forecasts the time and effort in terms of number of iterations and individual months and we compared the results in both techniques ' actions and costs. It is found that the agile approach is rational, as well as in cost and effort, thereby increasing the value and productivity of software development progression.

8.1 Introduction

The DNAM System comprises of 4 Modules that are refined through roles that came into existence through the goals which are at first identified and for the most basic step in prototype design. The DNAM (Genomic Information Retrieval) System takes as input a DNA pattern that has to be searched in the varied heterogeneous databases. Some even consisting of plain text files or excel sheets. We examine the relevance of crowdsourcing and a few procedures ranging from open innovation to smart technique and significant science in a non-advantage condition discover a U-shaped association between convenience time and winning in both sorts of challenges. Social capital lifts the probability of winning a gathering assessed challenge just if the social capital is enough high report a preliminary give an account of crowdsourcing testing for informational endeavors. We introduce three business programming things as enlightening testing wanders, which are crowdsourced by our indicating candidly strong system. We call this "Semi Crowdsourcing Test" (QCT) in light of the fact that the contender workers are understudies, who have certain social relations. The examination is encouraging and show to be favorable to both the understudies and industry in QCT wanders. The consequent estimations are pre choice of supporters, accessibility of partner responsibilities, collection of duties, and pay for responsibilities. By gathering the strategies of 46 crowdsourcing outlines, we perceive 19 unmistakable processes create. A resulting group examination exhibits general cases among these sorts and demonstrates an association with particular usages of crowdsourcing will tell that the crowdsourcing model of research can make hurt individuals, controls the part into continued with help, and uses individuals as exploratory subjects. We assume that traditions relying upon this model require Institutional Review Board (IRB) examination attempts to get an unrivaled appreciation of what crowdsourcing

systems are and what general arrangement points are considered in the change of such structures. This section identified the strengths of some modern and agile methods widely used as component of software development. I have likewise talked about the qualities and shortcomings between the two contradicting procedures and furnished the difficulties related with actualizing agile procedures in the software industry. This episodic confirmation is rising with respect to the adequacy of agile methodology in specific situations; however there have not been much gathering and investigation of observational proof for agile products. Notwithstanding, to help my exposition I led a poll, requesting criticism from programming industry professionals to assess which approach has a superior achievement rate for various sizes of software development. As per our discoveries dexterous procedures can give great advantages to little scaled and medium scaled undertakings yet for substantial scaled activities customary strategies appear to be overwhelming.

UI Module deals with taking the DNA pattern (Consisting of four letters, A,C,T and G sequences). It forwards this to the module which by looking at the routing information forwards the segmented sub-patterns to the various “source modules”. The source agents construct a local ontology and the patterns are searched in those ontologies. Then the results are combined to form a merged ontology and the links to the data sources are sent back to the user.

8.2 DNA Sequencing

Genomics contains DNA and, along with all other genes, the genome analysis includes the complete collection of DNA. It includes all the knowledge needed to build, maintain and sus-

tain the organism. Understanding the relationship between an unknown gene or DNA sequence and known sequences is essential for diagnosing such diseases.

The key knowledge in the field of genetics and biotechnology can only be given as below for a computer science researcher.

The bases of Nucleotides and their sequences are being used to classify all species qualities.

The Nucleotide bases are comprised of the following elements:

- Thymine (T)
- Cytosine (C)
- Guanine (G)
- Adenine (A)

We have a large, monolithic plenty of homogeneous and diverse databases that encompass many organisms' nucleotide base patterns. But the enormous size and scope of this massive evidence makes trawling for a "DNA pattern" in these databases very difficult for the biologists. Biologists or experts in life sciences need to scan the existing patterns contained in the databases for a particular sequence. To discover homologous patterns, this pattern matching is accomplished. This is useful because when a mutation occurs in an organism's DNA and this mutated DNA is the cause for some question, for example. Although if this mutated pattern streams several other species and the organism has been treated with certain enzymes, the test can be guided directly to that hormone that can be used to cure the cancer in the originating organism. The DNAM project is used as a research study to equate Agile with crowdsourcing and also helped by crowdsourcing conventional prototyping.

8.3 Crowdsourcing with Traditional Prototyping

The task of prototyping with crowd sourcing begins with goal identification that needs to be accomplished. The pictorial representation for what needs to be done at different levels of abstraction is goal diagram. The goals are broken down into sub tasks and this phenomenon is called task reduction resulting into Work breakdown Structure (WBS). The sub-divided goals need someone to take the responsibility to put them into action. Once the goal diagram is drawn, we assign different roles to the identified goals as shown in the figure 8.2.

Now we assign roles to various identified crowd sourcing workers.

The various crowd sourcing workers work on these segmented modules:

- **Initiator**

This module deals with the starting up of the system eg: reading the configuration file etc.

- **User Interface**

Deals with sub-modules like the way the end user will interact with the system.

- **Wrapper**

Deals with modules for routing the DNA Pattern over the network.

- **Source**

The sub-modules in Source module deal with the issues of platform transparency such that information from heterogeneous data sources can be extracted and the DNA pattern is searched. This is shown in figure 8.1.

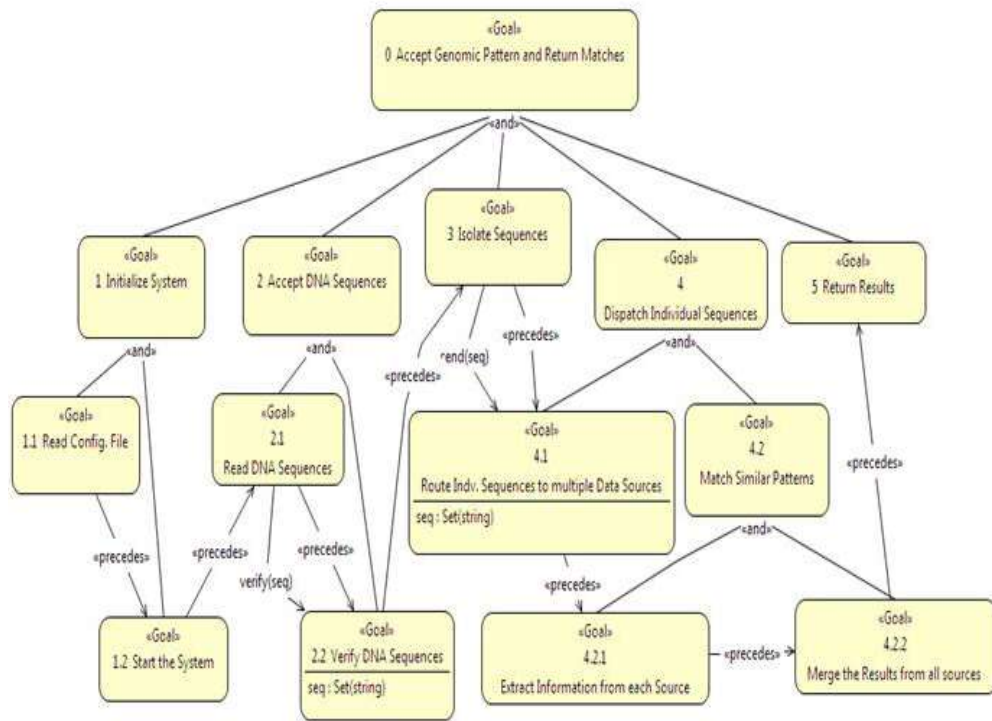


Figure 8.1: DNAM Goal Diagram

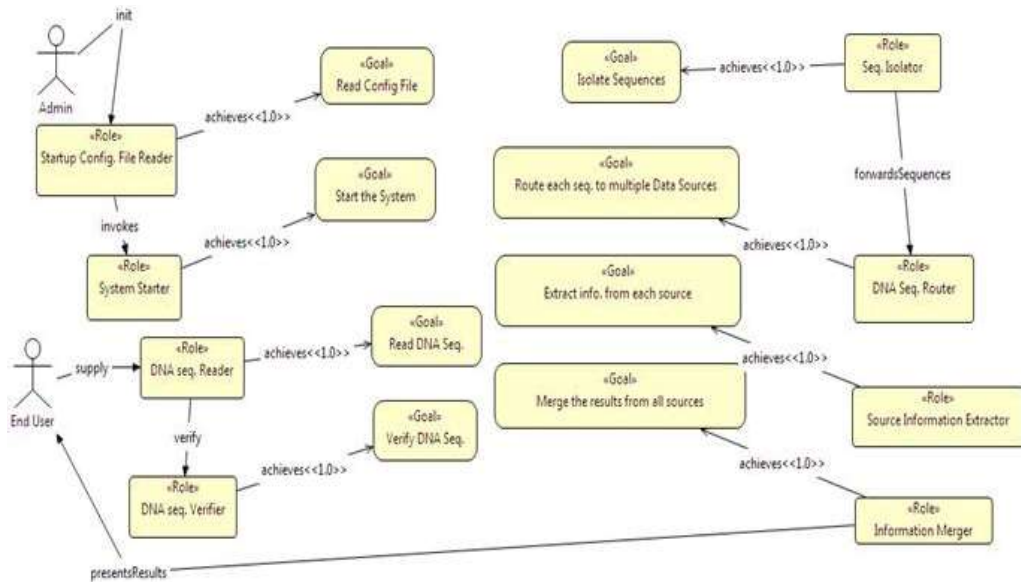


Figure 8.2: DNAM Role Diagram

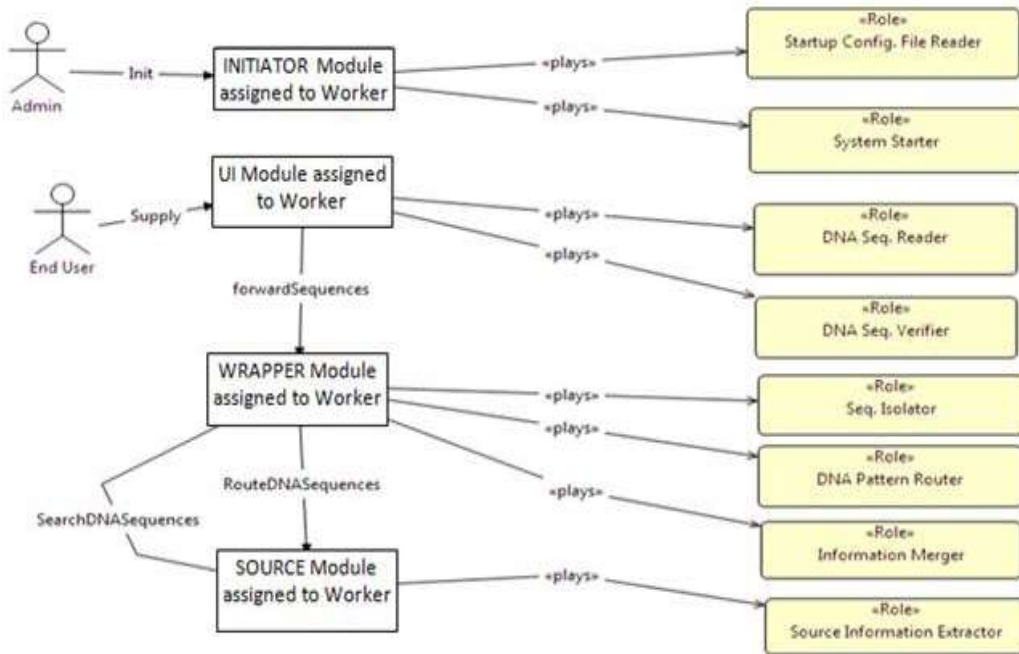


Figure 8.3: DNAM CrowdSourced Worker Task Assignment Diagram

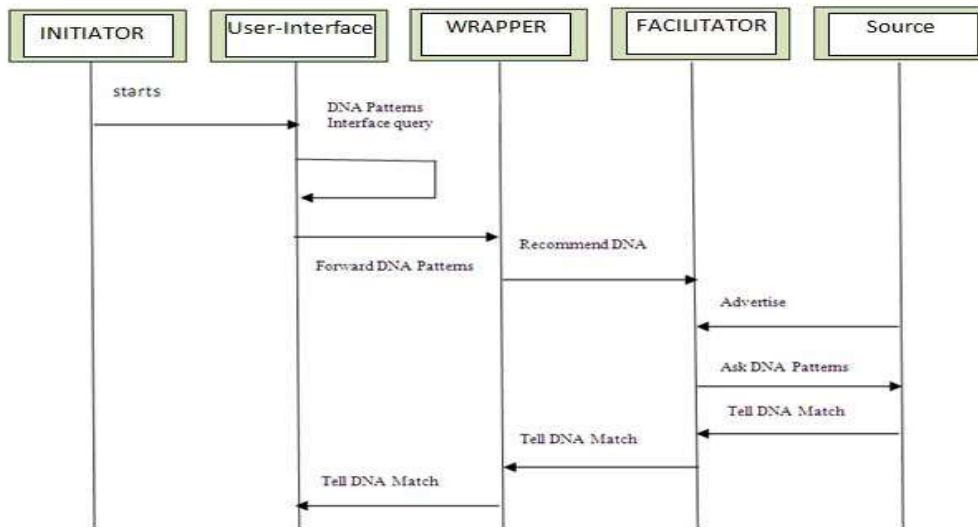


Figure 8.4: Interactions and Messages Exchanged Between Modules

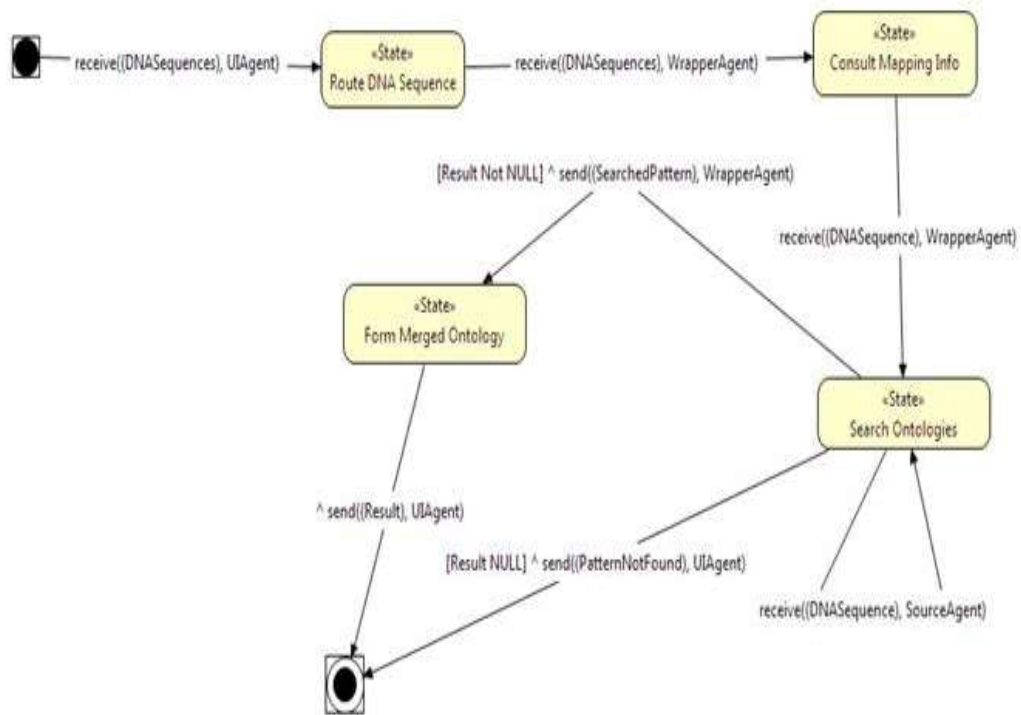


Figure 8.5: DNAM State Diagram

8.4 COCOMO II in Traditional (Prototyping) Software development approaches + Crowdsourcing

The development of a software system in different methodologies such as waterfall, spiral or incremental and agile approaches requires quite a few mathematical formulas for forecasting effort. The research uses COCOMO II early design model using 5 scale factors and 7 effort multipliers to project size (sloc), effort (person months) and time (months). The scale factors and effort multipliers are derived from the analysis done in a traditional case study involving the same project DNAM(Genomic Information Retrieval) accompanied by crowdsourcing.

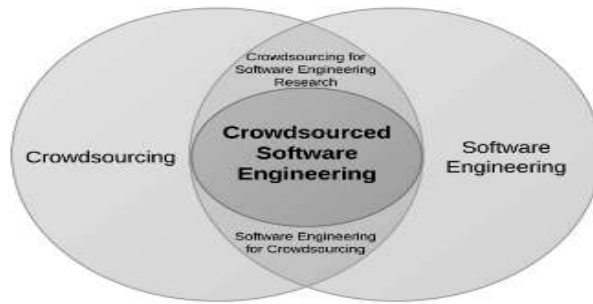


Figure 8.6: Coagulation of Traditional Software Engineering and Crowdsourcing

The diagrams are refined hierarchically from highest to lowest level of abstraction. High abstraction involves goals and plans and lowest abstraction involves with the state machines, coding and implementation.

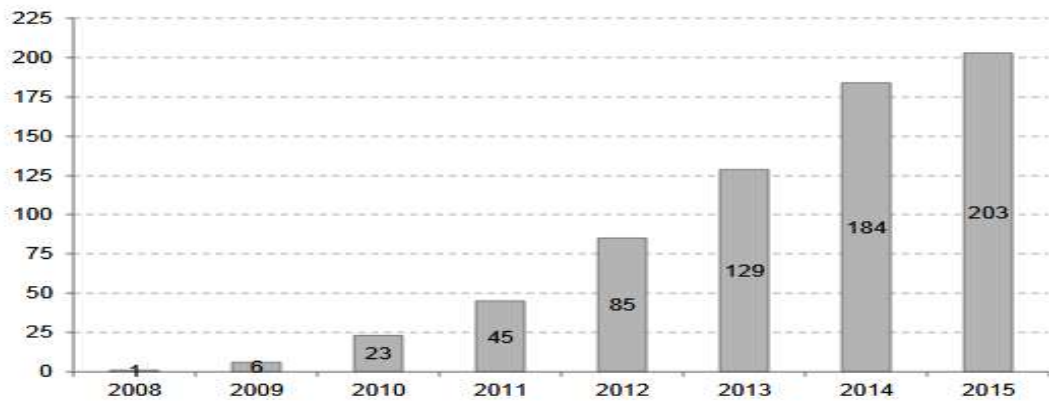


Figure 8.7: Cumulative growth of Crowdsourced Software Engineering studies published before April 2015.

8.5 Effort Estimation

The values for A, B, EMI and SFj in COCOMO II in Agile are standardized values taken from the effort of 161 projects in the model database. The formula for effort can be referred from Boehm (2000:13). In case of DNAM only 7 effort multipliers are used to remove confusion.

8.6 DNAM through Prototyping (Traditional SDLC)

$$PM = A \times \text{Size}^E \prod_{i=1}^{17} EM_i$$

$$A = 2.94 \text{ (for COCOMO II.2000)}$$

(1)

$$E = B + 0.01 \sum_{j=1}^5 SF_j$$

$$B = 0.91 \text{ (for COCOMO II.2000)}$$

(2)

The charts shown under represent the analysis done on Agile Development techniques involving crowdsourcing.

Table 8.1: Details identified of Agents through UML Design Toolkit in Prototype Model

Element	DNAM (Prototype)
Total No. of Interactions with other modules	6
Tot. No. of messages inter-changed	9
Tot. No. of Roles	8
Tot. No. of Requirements	8
Tot. No. of Events	9
Total no. of state machines applied	4
Tot. No. of states in every machine.	4

Rules identified in DNAM

R1: Maximum Time Limit: 30ns

R2: ONLY A, C, T, or G characters can be used to represent a DNA Pattern.

R3: The search string cannot contain white spaces, hyphens or any special character or numbers.

R4: Max Length of a search pattern can be set but assumed to be 1024 chars.

R5: Break-up size of search pattern can be set but assumed to be 11 chars.

R6: Patterns not conforming to P1, P2, P3 and P4 will be immediately discarded.

R7: Total time from query submission to result display can be maximum 8.00 secs.

R8: Queries from Agile to crowds failing to meet R6 will be held for resubmission.

Table 8.2: Statistical data about functional behavior

Element	DNAM (Prototype)
Tot. No. of Roles R	8
Tot. No. of Events E	9
Tot. No. of State Machines S	4
Total number of types of entities (R+E+S)	21
Rules dedicated to management of entities	8

Table 8.3: Scale Factors applied to the projects (Both in Agile and Traditional S/W Development)

Scale Factors	Range (DNAM Prototype)	Value (DNAM Prototype)
Precedentness	Nominal	3.72
Development Flexibility	Low	6.24

Architecture/ Risk Resolu- tion	Low	6.24
Team Cohe- sion	Low	2.19
Process Matur- ity	Low	6.24
		18.110

8.7 Scale Factors

The incremental aspect of the assessment is governed by five scale factors (SF) describing comparative efficiencies of scale or disease economies. Whether the approximation is less than 1. A project has economies of scale. If Exponent=1, then economies of scale and diseases are in equilibrium. If the exponent is more than one the project has diseconomies (Boehm 2000:30).

Table 8.4: Cost Drivers

Cost Drivers	Range(Prototype)	Value(Prototype)
Product Re- liability and Complexity	High	1.10
Reusability	Nominal	1.00
Platform dif- ficulty	Low	1.10
Personnel Capability	Nominal	1.00

Personnel Experience	Nominal	1.00
Facilities	Nominal	1.00
Required Development Schedule	Low	1.14 PIVal=1.38

Certain relevant factors, in addition to the scale factors, impact the developer's activities, called cost drivers. These are as follows:

1. Product Reliability and Complexity (RCPX) is a mixture of Software Reliability (RELY), Database Size (DATA), Documentation (DOCU) and Software Complexity (CPLX).
2. Reusability (RUSE)
3. Platform Difficulty (PDIF) is the mixture of Time Constraint (TIME), Main Storage Constraint (STOR) and Platform Volatility (PVOL).
4. Personnel Capability (PERS) is Analyst Capability (ACAP), Programmer Capability (PCAP) and Personnel Continuity (PCON).
5. Personnel Experience (PREX) clubbed Analyst Experience (AEXP), Programmer Experience (PEXP) and Language and Tools Experience (LTEX).
6. Facilities (FCIL) clubbed Uses of Software Tools (TOOL) and Site Environment (SITE)
7. Required Development Schedule (SCED)

The values for scale and cost factors are taken from the COCOMO Manual http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

Table 8.5: Equivalence of each element into Sloc

Element	Element Count DNAM(Prototype)	SLOC / Element (Prototype)	SLOC(Prototype)
Event	9	100	900
Rule	8	80	640
Goal	8	200	1600
Task	8	100	800
State Machines	4	150	600
TOTAL SLOC	-----	-----	4540
			4.5 KSLOC

KSLOC decreases in agile development due to the modern languages like PYTHON, RUBY, etc where we just need the required Tools (.dll files) that get easily integrated with our project.

Now Through Eq. 2:

$$PM = A \times \text{Size}^E \prod_{i=1}^{17} EM_i$$

$$A = 2.94 \text{ (for COCOMO II.2000)}$$

$$E = B + 0.01 \sum_{j=1}^5 SF_j$$

$$B = 0.91 \text{ (for COCOMO II.2000)}$$

$$E_{\text{prototype}} = 0.91 + 0.01 * (18.110) = 1.091$$

$$PM_{\text{prototype}} = 2.94 * (4.5)^{1.091} * (1.38) = \mathbf{20.94PM}$$

This means that a single crowdsourced worker can complete the whole process in nearly 21 months.

If there are n workers the job can be accomplished in $21/n$ months.

8.8 Analysis and Estimation of DNAM project using Agile with Crowdsourcing

8.8.1 Terminology

8.8.1.1 User Story

A "User Story" is a generic statement about how much a user plans to do with an asset included and the benefit that even the user would derive from that feature:

- What the user wants to do with a feature
- The User
- Value gained by the user
- Acceptance Criteria
- Front and back of a card

8.8.1.2 Story Points

A story point is the effort involved for the entire story to be implemented. It's a number which represents the size of the character based on how hard it is to incorporate a story.

8.8.1.3 Story Point Velocity

Velocity tests how much can be done over a specified period of time in the amount of a particular event; e.g. how many Story Points are reached during a sprint.

8.8.1.4 Simple Buildup Approach for Agile Project Cost Estimation

Velocity (V) = Iteration Duration / Completed Total SP.....1

Iterations needed = Total SP / Velocity2

A story point utilizes one of several preceding sizing scales in certain cases:

- 1, 2, 4,8,16
- X-Small, Small, Medium, Large, Extra-Large (known as “T-Shirt Sizing”)
- Fibonacci sequence: 1,2,3,5,8,13,21

8.8.1.5 Crowdsourcing the Agile software development

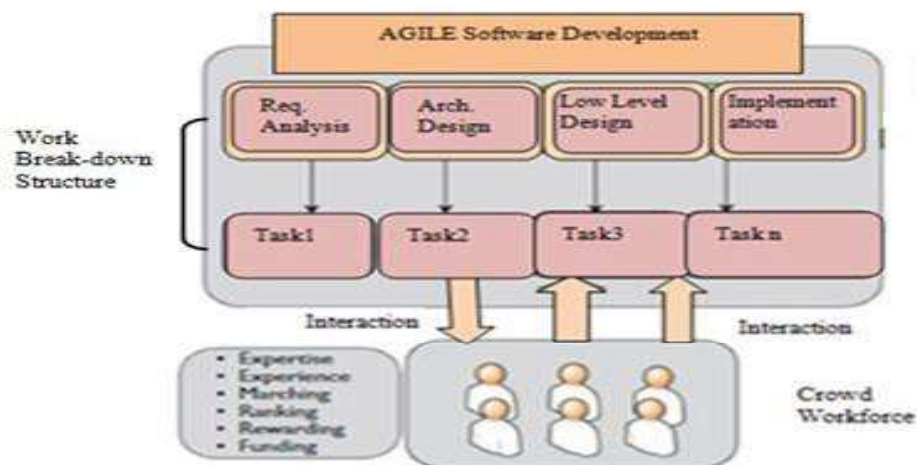


Figure 8.8: Agile with Crowdsourcing

Work break-down structure is the first process of crowdsourcing agile software development that begins with its construction, also called task reduction. Now when the work is decomposed into tasks, goals are identified by grouping similar tasks together. It is not always the case that multiple tasks unite to form a goal. A standalone task can result into a single goal as well.

8.8.1.6 Identifying Stories

<p style="text-align: center;">Login SP=1</p> <ul style="list-style-type: none"> > Registered user can log into the system. > Unsuccessful Login should be prompted with an Error Message. > User should be allowed login attempts upto 3 times 	<p style="text-align: center;">Accept DNA pattern SP=1</p> <ul style="list-style-type: none"> > DNA Input must only contain 'A', 'C', 'T' or 'G' characters. > No whitespaces, special symbols or digits are permissible. > The input DNA Pattern's length must only be 11 characters.
<p style="text-align: center;">Forward Pattern to Wrapper module SP= 2</p> <ul style="list-style-type: none"> > Route patterns to source modules. <ul style="list-style-type: none"> > Refer ontology > Merge results > Return matched results back to UI module 	<p style="text-align: center;">Receival of Pattern by Source Module SP=2</p> <ul style="list-style-type: none"> > Convert local database to XML <ul style="list-style-type: none"> > search local repository > Find matches > Pass matched DNA content to Wrapper Module
<p style="text-align: center;">User Interface SP=1</p> <ul style="list-style-type: none"> > Accept matched DNA sources 	

<p>> Return matched DNA sources list to the end user</p> <p>> Ready to accept new DNA patterns as query</p>	
---	--

The authors have developed first two user stories.

By equation 1 and 2,

Velocity (V) = 2/2=1;

Total iterations needed=Total SP/V=7/1=7;

if 1 iteration is assumed to be 2 weeks, then there are total 14weeks=**3.11months** and team size is 4.

In case of Prototype Model used along with crowdsourcing, we got the total time needed for development= $21/n$

Since in agile with crowdsourcing we took the team size to be equal to 4,

Time consumed by prototype with crowdsourcing= $21/4=$ **5.25 months**

It is obvious to figure out the level of increase in efficiency of the agile developers in crowd-sourced projects. As regarding the quality of these types of methods a brief theoretical comparison is shown under:

Costs & Benefits attached with Crowdsourcing:

- Concerns with regard to cost

- Reduced overhead Cost (OC)
- Increased Development Cost (DC)
- Schedule
- The problem with scheduling traditional software development is project completion dates vary with different developers. The cool undertaking does sometimes incur greater cost and time.

8.9 Concerns with regard to Quality

Practical quality of the software is thought to be met, when the predetermined necessities are met. The prerequisites ought to be successfully assembled. In conventional software development, the dispersal of assignments amid the arranging stage is less demanding when contrasted with crowdsourcing software development. Likewise coordinating distinctive modules after their advancement is finished is moderately less complex. Quality confirmation in crowdsourcing software development can be extremely testing. It turns out to be exceptionally troublesome and wrong to expel the bugs in the later phase of the software lifecycle. A portion of the difficulties experienced while keeping up the nature of software are: partitioning expansive work into little assignments, sorting out and conveying successfully. Plainly the general population associated with crowdsourcing is not a piece of any comparable association subsequently there is an issue of contradiction.

8.10 Conclusion

This chapter tries to apply effort forecasting in DNA pattern matching from vast heterogeneous sets of Data Sources (DNAM) taken as case study for comparing traditional and agile system currently developing under crowdsourcing and hypothesized that agile is better in re-

ducing development time thereby increasing efficiency greater as compared to traditional software development life cycle models.

CHAPTER 9

CONCLUSIONS AND FUTURE SCOPES

Over the past several years, crowdsourcing has emerged as a suitable method for companies that want to introduce new workable solutions from the crowd for micro-tasks. In tandem with the proliferation of this exercise among inventive organizations, scholars have become more and more interested in investigating crowdsourcing, making it a prominent topic among scholars. With the rapidly increasing purview of crowdsourcing, we can draw the conclusion that in the coming years, the relevance will extend its hand in all of these aspects in terms of agile crowdsourcing software development:

- 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.

The knowledge tools available on the Internet are being used slowly to support programming both at the time of design and at the time of execution. These appear as traditional administrations and, moreover, human learning works directly on the World Wide Web through crowd sourcing as well as data.

Crowdsourcing has already changed the current software development scenario through its various forms. Open source aside, the number of pristine Crowdsourcing platforms, the no. of corporations actively evaluating with Crowdsourcing, the number of employees signing up and continually contributing indicate, a process that have stepped in the software development industry through innumerable ways. The probable supremacies of adopting the crowdsourcing practices are concrete. The increment in the shift of development, work more dramatic shifts and occurrence to fluid labor markets alone, and does prognosticate the potential for even. Such shifts might raise a much awaited question and suspect the long-held fundamental beliefs about the software development practices, as in any fundamentally disorderly shift, the eventual spin-off are far from being particular. Now the question of the hour is that, will future developers be halfwitted automatons, or will they be operating as qualified workers and freelancers, choosing micro tasks to follow their passion and enhance their skills. Serious challenges have to be undertaken, if crowdsourcing is going to be that impactful in software industry, as it has proved to be in other industries. The nature of software has much to do with this aspect. Software development is compounded and intricate, and cannot be easily decoded into distinctly segmented, self contained and is not easily comprehended. Evolving an extensive cognizance of how and when to apply crowdsourcing in software development projects is valuable for the community. Crowdsourcing has been used as a technique to process some important works within agile software development. Crowdsourcing as a sourcing strategy, very less research exit; whereby the real development of agile software is crowdsourced and clubbed into a final product. Studies of crowdsourcing agile software development can also be used to combine two or more concept to produce research findings.

This research can be used to conduct hereafter exploration on the utilization of crowdsourcing like a sourcing tactics.

Crowdsourcing strategies 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 strategies (crowd funding, crowd creation, crowd reviews, crowd democracy, crowd wisdom) 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. The following are the important concluding remarks over the research work:

- In this study, by using the crowdsourcing method, we have increased the efficiency of the agile development process. The distortion and exponential intimidation of the necessities from the clients is treated with the regularly expanding and now and again settled or changeless conveyance of asset characteristics like CPU Time, memory, control and different attributes.
- In this research, the authors have devised model for Agile software development using Crowdsourcing. This is particularly done in order to duct the SDLC phases through workers in crowdsourcing. A methodology has been designed for the same. Agile is based on RAD, so in order to increase the velocity of the development of any agile product, crowdsourcing is utilized. As future scope efficiency and velocity of Agile software development is aimed to be maximized.

- This research has developed a meta-model named, OPFACS that with the help of semantics and ontological merging proves to implement, verify and validate the Agile Rules. DROOLS Reasoner with Protégé 5.2 is used to model and verify the crowd sourcing for satisfaction and attainment of the Rules in distributed environment. Merging of the ontologies in different versions of the same original RDF/web ontology language file is done by the Protégé's inbuilt merging tool.
- This research tries to apply Effort Forecasting in DNA pattern matching from vast heterogeneous sets of data Sources (DNAM) taken as case study for comparing Traditional and Agile System currently developing under crowdsourcing. The authors hypothesized that Agile is better in reducing development time thereby increasing efficiency greater as compared to traditional SDLC Models. The estimation in prototyping is done using COCOMO II Model merging crowdsourcing. The estimation of Agile with crowdsourcing is done using Simple WBS and Buildup Approach. In this paper the authors have tried to calculate the Effort required for a developer in developing the DNAM project thereby projecting the overall time taken in completion of the project by both software designs.

REFERENCES

- [1]. Ab Aziz, N.S.N. and Abdullah, R., 2015. "An Ontological Model of Experience-Based Knowledge Management in Agile software development Environment". *Lecture Notes on Software Engineering*, 3(1), p.26.
- [2]. Abbas, J., 2016. "Quintessence of traditional and agile requirement engineering". *Journal of Software Engineering and Applications*, 9(03), p.63.
- [3]. Afuah, A. and Tucci, C.L., 2012. "Crowdsourcing as a solution to distant search". *Academy of Management Review*, 37(3), pp.355-375.
- [4]. Aitken, A. and Ilango, V., 2013. "A comparative analysis of traditional software engineering and agile software development". In *System Sciences (HICSS), 2013 46th Hawaii International Conference on* (pp. 4751-4760). IEEE.
- [5]. Akerman, A. and Tyree, J., 2006. "Using ontology to support development of software architectures". *IBM Systems Journal*, 45(4), pp.813-825.
- [6]. Al-Zewairi, M., Biltawi, M., Etaiwi, W. and Shaout, A., 2017. "Agile software development methodologies: survey of surveys". *Journal of Computer and Communications*, 5(05), p.74.
- [7]. Alam, S.L. and Campbell, J., 2013. "Role of relational mechanisms in crowdsourcing Governance: an Interpretive Analysis".
- [8]. Altameem, E., 2015. "Impact of Agile Methodology on Software development". *Computer and Information Science*, 8(2), p.9.

- [9]. Alvertis, I., Papaspyros, D., Koussouris, S., Mouzakitis, S. and Askounis, D., 2016. “Using crowdsourced and anonymized Personas in the requirements elicitation and software development phases of software engineering. In Availability, Reliability and Security (ARES)”, 11th International Conference on (pp. 851-856). IEEE.
- [10]. Amini, S., Lin, J., Hong, J., Lindqvist, J. and Zhang, J., 2012. “Towards scalable evaluation of mobile applications through crowdsourcing and automation” (cmu-cylab-12-006).
- [11]. Amrollahi, A., Ghapanchi, A.H. and Talaei-Khoei, A., 2014. “Using crowdsourcing tools for implementing open strategy: A case study in education”. In 20th Americas Conference on Information Systems, AMCIS 2014.
- [12]. Anderson, M., 2011. “Crowdsourcing higher education: A design proposal for distributed learning”. MERLOT Journal of Online Learning and Teaching, 7(4), pp.576-590.
- [13]. Auler, R., Borin, E., Halleux, P. D., 2014. “Addressing JavaScript JIT Engines Performance Quirks: A Crowdsourced Adaptive Compiler”, LNCS 8409, pp. 218–237, 2014. Springer-Verlag Berlin Heidelberg.
- [14]. Babb J.R., Hoda, R. and Nørbjerg, J., 2013. “Barriers to Learning in Agile software development Projects”, H. Baumeister and B. Weber (Eds.): LNBIP 149, pp. 1–15, 2013. Springer-Verlag Berlin Heidelberg.
- [15]. Bajnaid, N.O., Benlamri, R., Pakstas, A. and Salekzamankhani, S., 2015. “Towards Ontology-based SQA Recommender for Agile software development”. J Inform Tech SoftwEng, 5(160), p.2.

- [16]. Bari. E, Johnston. M and Tsai. W-T, 2016. “Software Crowdsourcing Practices and Research Directions”, IEEE Symposium on Service-Oriented System Engineering.
- [17]. Benoit. K, Conway. D, Lauderdale. B.E and Laver. M. 2015. “Crowd-sourced text analysis: Reproducible and agile production of political data”. European Research Council grants ERC2011-StG 283794-QUANTESS.
- [18]. Bhatia, M.P.S., Kumar, A. and Beniwal, R., 2016. “Ontologies for software engineering: Past, present and future”. Indian Journal of Science and Technology, 9(9).
- [19]. Bider, I., 2014. “Analysis of agile software development from the knowledge transformation perspective”. In International Conference on Business Informatics Research (pp. 143-157). Springer, Cham.
- [20]. Bishop, J., 2015. “Supporting crowd-funded agile software development projects using contingent working: Exploring Big Data from participatory design documentation and interviews”. In Proceedings of the International Conference on Information and Knowledge Engineering (IKE) (p.43). The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- [21]. Bozzon. A, Brambilla. M, Ceri. S, Mauri. A, 2013. “Reactive Crowdsourcing”, International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author’s site if the Material is used in electronic media, Rio de Janeiro, Brazil. ACM 978-1-4503-2035-1/13/05.
- [22]. Brabham, D.C., 2008. “Crowdsourcing as a model for problem solving: An introduction and cases”. Convergence, 14(1), pp.75-90

- [23]. Breaux, T.D. and Schaub, F., 2014. "Scaling requirements extraction to the crowd: Experiments with privacy policies." In Requirements Engineering Conference (RE), 2014 IEEE 22nd International (pp. 163-172). IEEE.
- [24]. Buchan, J., Bano, M., Zowghi, D., MacDonell, S. and Shinde, A., 2017. "Alignment of Stakeholder Expectations about User Involvement in Agile software development". In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (pp. 334-343). ACM.
- [25]. Buecheler, T., HenrikSieg, J. M., Fuchslin, R. and Pfeife, R., 2010. "Crowdsourcing, Open Innovation and Collective Intelligence in the Scientific Method: A Research Agenda and Operational Framework", Proc. of the Alife XII Conference, Odense, Denmark.
- [26]. Bücheler, T. and Sieg, J.H., 2011. "Understanding science 2.0: Crowdsourcing and open innovation in the scientific method". *Procedia Computer Science*, 7, pp.327-329.
- [27]. Buettner, R., 2015. "A systematic literature review of crowdsourcing research from a human resource management perspective". In System Sciences (HICSS), 2015, 48th Hawaii International Conference on (pp. 4609-4618). IEEE.
- [28]. Burguera, I., Zurutuza, U. and Nadjm-Tehrani, S., 2011. "Crowdroid: behavior-based malware detection system for android". In Proceedings of the 1st ACM workshop on Security and privacy in smart phones and mobile devices (pp. 15-26). ACM.
- [29]. Challiol, C., Firmenich, S., Bosetti, G.A., Gordillo, S. and Rossi, G., 2013. "Crowdsourcing Mobile Web Applications". ICWE 2013 Workshops, LNCS 8295, pp. 223–237, © Springer International Publishing Switzerland.

- [30]. Chanal, V. and Caron-Fasan, M.L., 2008. "How to invent a new business model based on crowdsourcing: the Crowdsprite case". In *Conférence de l'Association Internationale de Management Stratégique* (pp. 1-27).
- [31]. Chatzimilioudis, G., Konstantinidis, A., Laoudias, C. and Zeinalipour-Yazti, D., 2012. "Crowdsourcing with smartphones". *IEEE Internet Computing*, 16(5), pp.36-44
- [32]. Chatzimilioudis, G. and Zeinalipour-Yazti, D., 2013. "Crowdsourcing for mobile data management. In *Mobile Data Management (MDM)*". *IEEE 14th International Conference on (Vol. 2, pp. 3-4)*. IEEE.
- [33]. Chen, C. and Zhang, K., 2014. "Who asked what: integrating crowdsourced FAQs into API documentation". In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 456-459). ACM.
- [34]. Chen, Z. and Luo, B., 2014. "Quasi-crowdsourcing testing for educational projects". In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 272-275). ACM.
- [35]. Chen, L. and Liu, D., 2012. "Comparing strategies for winning expert-rated and crowd-rated crowdsourcing contests". *First findings*.
- [36]. Chilana, P.K., Ko, A.J. and Wobbrock, J.O., 2012. "LemonAid: selection-based crowdsourced contextual help for web applications". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1549-1558). ACM.
- [37]. Choudhary, B. and Rakesh, S.K., 2016. "An approach using agile method for software development". In *Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, *International Conference on* (pp. 155-158). IEEE.

- [38]. Corneli, J., Mikroyannidis, A. 2012. "Crowdsourcing education on the Web: a role based analysis of online learning communities". In: Okada, Alexandra; Connolly, Teresa and Scott, Peter eds. Collaborative Learning 2.0: Open Educational Resources. Hershey, PA: IGI Global, pp. 272–286.
- [39]. Crowston, K., Howison, J. and Annabi, H., 2006. "Information systems success in free and open source software development: Theory and measures". *Software Process: Improvement and Practice*, 11(2), pp.123-148.
- [40]. DeLoach, S.A. and Garcia-Ojeda, J.C., 2010. "O-MaSE: a customizable approach to designing and building complex, adaptive multi-agent systems". *International Journal of Agent-Oriented Software Engineering*, 4(3), pp.244-280.
- [41]. De Lucia, A. and Qusef, A., 2010. "Requirements engineering in agile software development". *Journal of emerging technologies in web intelligence*, 2(3), pp.212-220.
- [42]. De Nicola, A. and Missikoff, M., 2016. "A lightweight methodology for rapid ontology engineering". *Communications of the ACM*, 59(3), pp.79-86.
- [43]. De Souza Carvalho, W.C., Rosa, P.F., dos Santos Soares, M., da Cunha Jr, M.A.T. and Buiatte, L.C., 2011. "A Comparative Analysis of the Agile and Traditional Software development Processes Productivity". In *Chilean Computer Science Society (SCCC), 2011, 30th International Conference of the* (pp. 74-82). IEEE.
- [44]. Della Mea, V, Maddalena, E, Mizzaro, S, 2013. "Crowdsourcing to Mobile Users: A Study of the Role of Platforms and Tasks", *DBCrowd: First VLDB Workshop on Databases and Crowdsourcing*.
- [45]. Despa, M.L., 2014. "Comparative study on software development methodologies". *Database Systems Journal*, 5(3), pp.37-56.

- [46]. Devedzic, V., 2011. "Teaching agile software development: A case study". IEEE Transactions on Education, 54(2), pp.273-278.
- [47]. Dingsøyr, T. and Lassenius, C., 2016. "Emerging themes in agile software development: Introduction to the special section on continuous value delivery". Information and Software Technology, 77, pp.56-60.
- [48]. Dingsøyr, T., Nerur, S., Balijepally, V. and Moe, N.B., 2012. "A decade of agile methodologies: Towards explaining agile software development".
- [49]. Doan .A, R. Ramakrishnan, and A. Y. Halevy 2011. "Crowdsourcing systems on the world wide web". Commun. ACM, 54(4):86–96.
- [50]. Dwarakanath. A, Chintala. U, Shrikanth N. C., Viridi. G, Kass. A, Chandran. A, Sengupta. S and Paul. S, 2015, "CrowdBuild: A Methodology for Enterprise Software development using Crowdsourcing". Conference Paper- DOI:10.1109/CSI-SE.2015.9.
- [51]. Dyba, T. and Dingsoyr, T., 2009. "What do we know about agile software development"? IEEE software, 26(5), pp.6-9.
- [52]. Elshandidy, H. and Mazen, S., 2013. "Agile and traditional requirements engineering: A survey". International Journal of Scientific & Engineering Research, 4(9), pp.473-482.
- [53]. Estellés-Arolas, E. & González Ladrón-de- Guevara, F. 2012. "Towards an integrated crowdsourcing definition". Journal of Information Science (JCR: 1, 41).
- [54]. Fan, J., Zhang, X.P. and Dong, T.Y., 2008. "Research of Plant Domain Knowledge-Model Based on Ontology. In Innovative Computing Information and Control". ICIC-IC'08. 3rd International Conference on (pp. 108-108). IEEE.

- [55]. Fergis, K., 2012. "The impact of an agile methodology on software development costs". Tata Consultancy Services Ltd, 2016. Recommendation system for agile software development. U.S. Patent 9,262,126.
- [56]. Fitsilis, P., 2008. "Comparing PMBOK and Agile Project Management software development processes". In *Advances in Computer and Information Sciences and Engineering* (pp. 378-383). Springer, Dordrecht.
- [57]. Geerts, S. 2009. "Discovering crowdsourcing: theory, classification and directions for use". Master thesis. Technische Universiteiten Eindhoven.
- [58]. Geiger, D., Seedorf, S. and Schader, M. 2011. "Managing the Crowd: Towards Taxonomy of Crowdsourcing Processes". In: *Proceedings of the Seventeenth Americas Conference on Information Systems*, Detroit, Michigan August 4th- 7th 2011.
- [59]. Ghobadi, S. and Mathiassen, L., 2016. "Perceived barriers to effective knowledge sharing in agile software teams". *Information Systems Journal*, 26(2), pp.95-125.
- [60]. Golub, B. and Jackson, M.O., 2010. "Naive learning in social networks and the wisdom of crowds". *American Economic Journal: Microeconomics*, 2(1), pp.112-49.
- [61]. Gong, H., Guo, J., Yu, Z., Zhang, Y. and Xue, Z., 2008. "Research on the building and reasoning of travel ontology". In *Intelligent Information Technology Application Workshops, 2008. IITAW'08. International Symposium on* (pp. 94-97). IEEE.
- [62]. Graber, M.A. and Graber, A., 2013. "Internet-based crowdsourcing and research ethics: the case for IRB review". *Journal of medical ethics*, 39(2), pp.115-118.
- [63]. Guan, H., Yang, H. and Wang, J., 2016. "An ontology based approach to security pattern selection". *International Journal of Automation and Computing*, 13(2), pp.168-182.

- [64]. Happel, H.J. and Seedorf, S., 2006. "Applications of ontologies in software engineering". In Proc. of Workshop on Semantic Web Enabled Software Engineering"(SWESE) on the ISWC (pp. 5-9).
- [65]. Hasteer, N, Bansal, A and Murth. B.K. 2015. "Crowdsourced Software development Process: Investigation and Modeling through Markov Decision Theory". International Journal of Software Engineering and Its Applications Vol. 9, No. 9, pp. 41-54.
- [66]. He-ping, C., Lu, H. and Bin, C., 2008. "Research and implementation of ontology automatic construction based on relational database". In Computer Science and Software Engineering, 2008 International Conference on (Vol. 5, pp. 1078-1081). IEEE.
- [67]. Heer, J and Bostock.M ,2010. "Crowdsourcing graphical perception: using Mechanical Turk to assess visualization design". In Proceedings of the 28th international conference on Human factors in computing systems.
- [68]. Hellmann, T.D., Sharma, A., Ferreira, J. and Maurer, F., 2012. "Agile Testing: Past, Present, and Future-Charting a Systematic Map of Testing in Agile software development". In Agile Conference (AGILE), 2012 (pp. 55-63). IEEE.
- [69]. Hetmank, L., 2013. "Components and Functions of Crowdsourcing Systems-A Systematic Literature Review". Wirtschaftsinformatik, 4, p.2013.
- [70]. Hirankitti, V. and Mai, T.X., 2012. "A meta-logical approach for reasoning with an web ontology language 2 ontology". Journal of Ambient Intelligence and Humanized Computing, 3(4), pp.293-303.
- [71]. Hneif, M. and Ow, S.H., 2009. "Review of agile methodologies in software development". International Journal of Research and Reviews in Applied Sciences, 1(1), pp.p1-8.

- [72]. Howe, J., 2006. "The rise of crowdsourcing". *Wired magazine*, 14(6), pp.1-4.
- [73]. Hu, Z. and Wu, W., 2014. "A game theoretic model of software crowdsourcing". In 2014 IEEE 8th International Symposium on Service Oriented System Engineering (SOSE) (pp. 446-453). IEEE.
- [74]. Hu, Z.G., Yuan, Q. and Zhang, X., 2009. "Research on agile project management with scrum method". In *Services Science, Management and Engineering, 2009. SSME'09. IITA International Conference on* (pp. 26-29). IEEE.
- [75]. Huberman B. A, Romero D.M, Wu. F, 2009. "Crowdsourcing, attention and productivity". *Journal of Information Science*, 35 (6) 2009, pp. 758–765.
- [76]. Hyland-Wood, D., Carrington, D. and Kaplan, S., 2006. "Toward a software maintenance methodology using semantic web techniques". In *null* (pp. 23-30). IEEE.
- [77]. Ionel, N., 2009. "Agile software development methodologies: An overview of the current state of research". *Annals of Faculty of Economics*, 4(1), pp.381-385.
- [78]. Ismail, Q., Ahmed, T., Kapadia, A. and Reiter, M.K., 2015. "Crowdsourced exploration of security configurations". In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 467-476). ACM.
- [79]. Jammalamadaka, K. and Krishna, V.R., 2013. "Agile software development and challenges". *International Journal of Research in Engineering and Technology*, 2(08), pp.125-129.
- [80]. Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson., 2010, "Who are the crowdworkers?: shifting demographics in Mechanical Turk". In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*.

- [81]. Kåreborn .B.B, Howcroft.D, 2013. “The Apple business model: Crowdsourcing mobile applications”. *Accounting Forum* · DOI: 10.1016/j.accfor.2013.06.00.
- [82]. Kazai, G. and Milic-Frayling, N., 2009. “On the evaluation of the quality of relevance assessments collected through crowdsourcing”. In *SIGIR 2009 Workshop on the Future of IR Evaluation (Vol. 21)*.
- [83]. Kazman, R. and Chen, H.M., 2009. “The metropolis models a new logic for development of crowdsourced systems”. *Communications of the ACM*, 52(7), pp.76-84.
- [84]. Kittur, A., Chi, E.H. and Suh, B., 2008. “Crowdsourcing user studies with Mechanical Turk”. In *Proceedings of the SIGCHI conference on human factors in computing systems*(pp. 453-456). ACM.
- [85]. KiranHiwarkar. K., 2016. “Comparative Analysis of Agile software development Methodologies-A Review”. *Int. Journal of Engineering Research and Applications*. Vol. 6, Issue 3, (Part - 6) March 2016, pp.80-85.
- [86]. Knublauch, H., 2004. “Ontology-driven software development in the context of the semantic web: An example scenario with Protege/Web Ontology Language”. In *1st International workshop on the model-driven semantic web (MDSW2004)* (pp. 381-401).
- [87]. Knublauch, H., Fergerson, R.W., Noy, N.F. and Musen, M.A., 2004. “The Protégé WEB ONTOLOGY LANGUAGE plugin: An open development environment for semantic web applications”. In *International Semantic Web Conference* (pp. 229-243). Springer, Berlin, Heidelberg.
- [88]. Košinár, M., 2017. “Knowledgemodeling of agile processes in healthcare systems development”. *Lékař a technik* Clinician and Technology, 43(4), pp.32-35.

- [89]. Kosinski, M., Bachrach, Y., Kasneci, G., Van-Gael, J. and Graepel, T., 2012. "Crowd IQ: Measuring the intelligence of crowdsourcing platforms". In Proceedings of the 4th annual ACM web science conference (pp. 151-160). ACM.
- [90]. Kozinets, R. V., Hemetsberger, A., and Schau, H. J. 2008. "The Wisdom of Consumer Crowds: Collective Innovation in the Age of Networked Marketing". *Journal of Macro-marketing*, 28 (4): 339-354.
- [91]. Kumar, G. and Bhatia, P.K., 2012. "Impact of Agile methodology on software development process". *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 2(4).
- [92]. Lane, S. 2010. "Collective Intelligence for Competitive Advantage: Crowdsourcing and Open Innovation". *Applied Information Management Master's Capstone Projects and Papers*.
- [93]. Lakhani, K., Garvin, D. and Lonstein, E., 2010. "Topcoder (a): Developing software through crowdsourcing".
- [94]. LaToza, T.D. and van der Hoek, A., 2016. "Crowdsourcing in software engineering: Models, motivations, and challenges". *IEEE software*, 33(1), pp.74-80.
- [95]. Leicht, N., Durward, D., Blohm, I. and Leimeister, J.M., 2015. "Crowdsourcing in Software development: A State-of-the-Art Analysis".
- [96]. Leimeister, J.M., Huber, M., Bretschneider, U. and Krcmar, H., 2009. "Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition". *Journal of management information systems*, 26(1), pp.197-224.
- [97]. Li, K., Xiao, J., Wang, Y. and Wang, Q., 2013. "Analysis of the key factors for software quality in crowdsourcing development: An empirical study on topcoder. Com".

- In 2013 IEEE 37th Annual Computer Software and Applications Conference (COMP-SAC) (pp. 812-817). IEEE.
- [98]. Lim, S.L. and Ncube, C., 2013. "Social networks and crowdsourcing for stakeholder analysis in system of systems projects". In System of Systems Engineering (SoSE), 2013 8th International Conference on (pp. 13-18). IEEE. .
- [99]. Lindsjörn, Y., Sjøberg, D.I., Dingsøy, T., Bergersen, G.R. and Dybå, T., 2016. "Teamwork quality and project success in software development: A survey of agile development teams". *Journal of Systems and Software*, 122, pp.274-286.
- [100]. M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin., 2011. "Crowddb: answering queries with crowdsourcing". In ACM SIGMOD 2011, pages 61–72. ACM, 2011.
- [101]. Mahmud, F. and Aris, H., 2015. "State of mobile crowdsourcing applications: A review". In Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on (pp. 27-32). IEEE.
- [102]. Malone, T. W., R. Laubacher, C. Dellarocas. 2009. "Harnessing Crowds: Mapping the Genome of Collective Intelligence". MIT sloan research paper (4732-09).
- [103]. Mao, K., Capra, L., Harman, M. and Jia, Y., 2015. "A survey of the use of crowdsourcing in software engineering". *Rn*, 15(01).

- [104]. Mao.K, Yang.Y, Li.M 2013. "Pricing Crowdsourcing-Based Software Development Tasks". San Francisco, CA, USA New Ideas and Emerging Results 978-1-4673-3076-3/13/\$31.0 IEEE.
- [105]. Martin, S.F., Falkenberg, H., Dyrland, T.F., Khoudoli, G.A., Mageean, C.J. and Linding, R., 2013. "PROTEIN CHALLENGE: crowd sourcing in proteomics analysis and software development". *Journal of proteomics*, 88, pp.41-46.
- [106]. Marwat, M.I., Jan, S. and Shah, S.Z.A., 2013. "Software Architectural Design Ontology". *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 7, p.12.
- [107]. Mason, W. and Watts, D.J., 2009. "Financial incentives and the performance of crowds". In *Proceedings of the ACM SIGKDD workshop on human computation* (pp. 77-85). ACM.
- [108]. Mehra, P., Brooks, R. and Thomas, C., 2010. "Ontology creation by reference to a knowledgecorpus". Hewlett-Packard Development Co LP. U.S. Patent Application 12/432,492.
- [109]. Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, Katrina Panovich. 2015. "Soylent: A Word Processor with a Crowd Inside". *Communications of the ACM*, Vol. 58 No. 8, 85-94.
- [110]. Mika, P., 2005. "Ontologies are us: A unified model of social networks and semantics". In *International semantic web conference* (pp. 522-536). Springer, Berlin, Heidelberg.

- [111]. Missiroli, M., Russo, D. and Ciancarini, P., 2016. "Learning agile software development in high school: an investigation". In Proceedings of the 38th International Conference on Software Engineering Companion (pp. 293-302).
- [112]. Moniruzzaman, A.B.M. and Hossain, D.S.A., 2013. "Comparative Study on Agile software development methodologies". arXiv preprint arXiv:1307.3356.
- [113]. Mujumdar, D., Kallenbach, M., Liu, B. and Hartmann, B., 2011. "Crowdsourcing suggestions to programming problems for dynamic web development languages". In CHI'11 Extended Abstracts on Human Factors in Computing Systems (pp. 1525-1530). ACM.
- [114]. Murphy, B., Bird, C., Zimmermann, T., Williams, L., Nagappan, N. and Begel, A., 2013. "Have agile techniques been the silver bullet for software development at microsoft?". In Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on (pp. 75-84). IEEE.
- [115]. Naik, N., 2017. "Software CROWD-sourcing. In Research Challenges in Information Science (RCIS)". 2017 11th International Conference on (pp. 463-464). IEEE.
- [116]. Nakatsu, R.T. and Iacovou, C.L., 2009. "A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study". *Information & Management*, 46(1), pp.57-68.
- [117]. Narock, T.W. and Hitzler, P., 2013. "Crowdsourcing semantics for big data in geoscience applications". In AAAI 2013 Fall symposium series, semantics for Big Data, November (pp. 15-17).
- [118]. Noll, J. and Atkinson, D.C., 2003. "Comparing extreme programming to traditional development for student projects: A case study". In International Conference on Extreme

Programming and Agile Processes in Software Engineering (pp. 372-374). Springer, Berlin, Heidelberg.

[119]. Pan, J.Z. and Horrocks, I., 2007. “Rdfs (fa): connecting rdf (s) and Web Ontology Language dl”. IEEE Transactions on Knowledge & Data Engineering, (2), pp.192-206.

[120]. Papamartzivanos, D., Damopoulos, D. and Kambourakis, G., 2014. “A cloud-based architecture to crowdsource mobile app privacy leaks”. In Proceedings of the 18th Panhellenic Conference on Informatics (pp. 1-6). ACM.

[121]. Parsons, D., Ryu, H. and Lal, R., 2007. “The impact of methods and techniques on outcomes from agile software development projects”. In IFIP International Working Conference on Organizational Dynamics of Technology-Based Innovation (pp. 235-249). Springer, Boston, MA.

[122]. Pastore, F., Mariani, L. and Fraser, G., 2013. “Crowdoracles: Can the crowd solve the oracle problem”. In International Conference on Software Testing, Verification and Validation (ICST) (pp. 342-351).

[123]. Pedersen, J., de Vreede, G.J., Kocsis, D., Tripathi, A., Tarrell, A., Weerakoon, A., Tahmasbi, N., Xiong, J., Deng, W. and Oh, O., 2013. “Conceptual foundations of crowdsourcing: A review of IS research. In 2013 46th Hawaii International Conference on System Sciences (pp. 579-588). IEEE.

[124]. Peng, X., Babar, M.A. and Ebert, C., 2014. “Collaborative software development platforms for crowdsourcing”. IEEE software, 31(2), pp.30-36. article provides an overview of current technologies for crowdsourcing.

- [125]. Pico-Valencia, P. and Holgado-Terriza, J.A., 2016. "Semantic agent contracts for internet of agents". In Web Intelligence Workshops (WIW), IEEE/WIC/ACM International Conference on (pp. 76-79). IEEE.
- [126]. Pierre A. Akiki, Arosha K. Bandara, and YijunYu , 2013. "Crowdsourcing User Interface Adaptations for Minimizing the Bloat in Enterprise Applications" EICS'13, June 24–27, 2013, London, United Kingdom.
- [127]. Poller, A., Kocksch, L., Türpe, S., Epp, F.A. and Kinder-Kurlanda, K., 2017. "Can Security Become a Routine?: A Study of Organizational Change in an Agile software development Group". In CSCW (pp. 2489-2503).
- [128]. Popli, R. and Chauhan, N., 2014. "Cost and effort estimation in agile software development". In Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on (pp. 57-61). IEEE.
- [129]. Quan, L., Xinjuan, J. and Yihong, L., 2007. "Research on ontology-based representation and retrieval of components". In null (pp. 494-499). IEEE.
- [130]. Rai, P. and Dhir, S., 2014. "Impact of different methodologies in software development process". International Journal of Computer Science and Information Technologies, 5(2), pp.1112-1116
- [131]. Rani. G, Bajwa .J.K., 2017. "Jan MAPPING AND ANALYSIS OF AGILE QUALITY ASSURANCE METRICS". International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106, Volume-5, Issue-1.
- [132]. Rao, K.N., Naidu, G.K. and Chakka, P., 2011. "A study of the agile software development methods, applicability and implications in industry". International Journal of Software Engineering and its applications, 5(2), pp.35-45.

- [133]. Riedl, J. and Riedl, E., 2013. "Crowdsourcing Medical Research". IEEE Computer, 46(1), pp.89-92.
- [134]. Salay, R., Dalpiaz, F., 2015. "Integrating Crowd Intelligence into Software". IEEE/ACM 2nd International Workshop on Crowdsourcing in Software Engineering
- [135]. Sanghvi, A., Sehgal, P., Kapur, A. and Singh, S., 2017. "Influence of the number of people in agile software development". International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 04 Issue: 01.
- [136]. Sauerwein, C., Gander, M., Felderer, M. and Breu, R., 2016. "A systematic literature review of crowdsourcing-based research in information security". In Service-Oriented System Engineering (SOSE), 2016 IEEE Symposium on (pp. 364-371). IEEE.
- [137]. Saxton, G.D., Oh, O. and Kishore, R., 2013. "Rules of crowdsourcing: Models, issues, and systems of control". Information Systems Management, 30(1), pp.2-20.
- [138]. Shankhdhar, G.K. and Darbari, M., 2016. "Building custom, adaptive and heterogeneous multi-agent systems for semantic information retrieval using organizational-multi-agent systems engineering, O-MaSE". In Advances in Computing, Communication, & Automation (ICACCA) (fall), International Conference on (pp. 1-7). IEEE.
- [139]. Sheeba, T., Begum, S.H. and Bernard, M.J., 2012. "Semantic web to e-Learning content". International Journal, 2(10), pp.58-66.
- [140]. Sherief, N., Jiang, N., Hosseini, M., Phalp, K. and Ali, R., 2014. "Crowdsourcing software evaluation". In proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (p. 19). ACM.
- [141]. Skaržauskaitė, M., 2012. "The application of crowd sourcing in educational activities". Social Technologies, 2(1), pp.67-76.

- [142]. Sletholt, M.T., Hannay, J., Pfahl, D., Benestad, H.C. and Langtangen, H.P., 2011., “A literature review of agile practices and their effects in scientific software development”. In Proceedings of the 4th international workshop on software engineering for computational science and engineering (pp. 1-9). ACM.
- [143]. Snijders, R., Dalpiaz, F., Hosseini, M., Shahri, A. and Ali, R., 2014. “Crowd-centric requirements engineering. In Utility and Cloud Computing (UCC)”. IEEE/ACM 7th International Conference on (pp. 614-615). IEEE.
- [144]. Soundararajan, S., Arthur, J.D. and Balci, O., 2012. “A methodology for assessing agile software development methods”. In Agile Conference (AGILE), 2012 (pp. 51-54). IEEE.
- [145]. Stol, K.J., Caglayan, B. and Fitzgerald, B., 2017. “Competition-Based CrowdsourcingSoftware development: A Multi-Method Study from a Customer Perspective”. IEEE Transactions on Software Engineering.
- [146]. Stol.K.J, Fitzgerald. B, 2014. “Researching CrowdsourcingSoftware development: Perspectives and Concerns”. CSI-SE’14, Hyderabad, India. ACM 978-1-4503-2857-9/14/06.
- [147]. Stol, K.J., LaToza, T.D. and Bird, C., 2017. “Crowdsourcing for software engineering”. IEEE software, 34(2), pp.30-36.
- [148]. Stolee. K. T, Elbaum. S, 2012. “Exploring the Use of Crowdsourcing to Support Empirical Studies in Software Engineering”. University of Nebraska – Lincoln Lincoln, NE, U.S.A..
- [149]. Storey, M.A., Treude, C., van Deursen, A. and Cheng, L.T., 2010. “The impact of social media on software engineering practices and tools”. In Proceedings of the FSE/SDP workshop on Future of software engineering research (pp. 359-364). ACM.

- [150]. Störl, U., Müller, D., Klettke, M. and Scherzinger, S., 2017. "Enabling Efficient Agile software development of NoSQL-backed Applications". *Datenbanksystemefür Business, Technologie und Web*.
- [151]. Suganthy, A. and Chithralekha, T., 2016. "Application of crowdsourcing in software development". In *Recent Trends in Information Technology (ICRTIT), 2016 International Conference on* (pp. 1-6). IEEE.
- [152]. Tarwani, S. and Chug, A., 2016. "Agile Methodologies in Software Maintenance: A Systematic Review". *Informatica*, 40(4).
- [153]. Thomas D. LaToza. T. D and Hoek. A.V.D, 2015. "A Vision of Crowd Development". *IEEE/ACM 37th IEEE International Conference on Software Engineering*.
- [154]. Tripp, J.F., 2012. "The impacts of agile development methodology use on project success: A contingency view" (p. 217). Michigan State University. *Business Information Systems*.
- [155]. Tsai, W.T., Wu, W. and Huhns, M.N., 2014. "Cloud-based software crowdsourcing". *IEEE Internet Computing*, (3), pp.78-83.
- [156]. Ullah, M., 2014. "Comparison and problems between Traditional and Agile software development methods".
- [157]. Vaish. R and Davis. J, 2015. "Crowdsourcing the Research Process". *Collective Intelligence 2015*.
- [158]. Vijayarathy, L.R. and Butler, C.W., 2016. "Choice of software development methodologies: Do organizational, project, and team characteristics matter?". *IEEE software*, 33(5), pp.86-94.

- [159]. Vukovic, M., 2009. "Crowdsourcing for enterprises". In *Services-I, World Conference on* (pp. 686-692). IEEE.
- [160]. Wang, J., Wang, S., Cui, Q. and Wang, Q., 2016. "Local-based active classification of test report to assist crowdsourced testing". In *Automated Software Engineering (ASE), 31st IEEE/ACM International Conference on* (pp. 190-201). IEEE.
- [161]. Wasserman, A.I., 2010. "Software engineering issues for mobile application development". In *Proceedings of the FSE/SDP workshop on Future of software engineering research* (pp. 397-400). ACM.
- [162]. Wexler M. N. 2011. "Reconfiguring the sociology of the crowd: exploring crowdsourcing". *International Journal of Sociology and Social Policy* 31(1): 6-20.
- [163]. Wongthongtham, P., Chang, E., Dillon, T. and Sommerville, I., 2009. "Development of software engineering ontology for multisite software development". *IEEE Transactions on Knowledge and Data Engineering*, 21(8), pp.1205-1217.
- [164]. Wongthongtham, P., Chang, E., Dillon, T.S. and Sommerville, I., 2005. "Software engineering ontologies and their implementation". In *Proceedings of the IASTED International Conference on Software Engineering (SE)* (pp. 208-213). ACTA Publishing.
- [165]. Wu, W., Tsai, W.T. and Li, W., 2013." An evaluation framework for software crowdsourcing". *Frontiers of Computer Science*, 7(5), pp.694-709.
- [166]. Wu, W., Tsai, W.T. and Li, W., 2013. "Creative software crowdsourcing: from components and algorithm development to project concept formations". *International Journal of Creative Computing*, 1(1), pp.57-91.
- [167]. Ying, X. and Fu-yuan, X., 2006. "Research on context modeling based on ontology. In *Computational Intelligence for Modelling, Control and Automation*". International

Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on (pp. 188-188). IEEE.

[168]. Yuan, H., Han, Y. and Hu, J., 2008. "Research on Agile Development Methodology of Service-Oriented Personalized Software". In Computer Science and Software Engineering. International Conference on (Vol. 4, pp. 1075-1078). IEEE.

[169]. Yuen, M.C., King, I. and Leung, K.S., 2011. "A survey of crowdsourcing systems. In Privacy, Security, Risk and Trust (PASSAT)". IEEE Third International Conference on Social Computing (SocialCom).

[170]. Zakariah, Z., 2015. "Business Model of Crowdsourcing: Review Paper". 6th Control and System Graduate Research Colloquium, UiTM, Shah Alam, Malaysia. IEEE.

[171]. Zhao, H., Zhang, S. and Zhao, J., 2012. "Research of using protege to build ontology". In Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on (pp. 697-700). IEEE.

[172]. Zogaj, S., Bretschneider, U., and Leimeister, J.M. 2014. "Managing Crowdsourced Software Testing: A Case Study Based Insight on the Challenges of a Crowdsourcing Intermediary," *Journal of Business Economics* (84:3), pp. 375-405.