

Comparison of usage of crowdsourcing in traditional and agile software development methodologies on the basis of effectiveness

Himanshu Pandey



Research Scholar
MUIT, LUCKNOW, INDIA
hpandey010@gmail.com

Abstract

The authors here try to forecast the Effort in Person Months for developing Agile and the traditional way of software development including Prototyping. The comparison is made on the basis of considering both Agile and traditional software development methodologies in addition to the crowdsourcing paradigm applied to both approaches. DNA Matcher (DNAM) has been developed using both prototyping and Agile software development with crowd sourcing. For Agile development, first of all COCOMO II Model is applied on it utilizing the crowdsourcing technique. The authors have determined that Agile development proves to be considerably economical when both techniques use crowdsourcing. The case study used here is DNAM. First DNAM was developed using traditional prototyping methods. During its analysis, costing is done. This is done in accordance with the crowdsourcing used in parallel to the Prototyping method. The time and effort in Person Months (PM) was known. Then AGILE development methodology is used in the development of DNAM. Agile is used along with the crowdsourcing paradigm. As soon as the analysis phase is completed, Simple Build-up Approach forecasts the time and effort in terms of Number of Iterations and Person Months and we compare the results of Effort and Cost of both the techniques. The Agile method is found to be both, less in cost and effort, thereby increasing the Effectiveness and Efficiency of the progression of Software development.

Keywords: Crowdsourcing, Genomic Information Retrieval, Agile software development, COCOMO, Simple Build-up Method

1. Introduction

The DNAM System [1] comprises of 4 Modules that are refined though roles that came into existence though 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 [3]. Here, the authors inspect the pertinence of Crowdsourcing and a couple of procedures from Open Innovation to the intelligent technique and major science in a non-advantage condition [4], discover a U-shaped association between the convenience time and winning in the two sorts of difficulties. Social capital lifts the probability of winning a gathering assessed challenge just if the social capital is enough high [5], 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 occurs are encouraging and show to be favorable to both the understudies and industry in QCT wanders. [6], Revolves just around a legitimate perspective and on the frameworks available to these affiliations. 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 process creates. A resulting group examination exhibits general cases among these sorts and demonstrates an association with particular usages of crowdsourcing. [7], will demonstrate 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. [8], attempts to get an unrivaled appreciation of what crowdsourcing systems are and what general arrangement points are considered in the change of such structures. In this paper, the maker drove a ponder composing review in the space of crowdsourcing structures.

[10], portrayed the qualities of some conventional and agile techniques that are generally utilized as a part of software development. I have likewise talked about the qualities and shortcoming 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) [3]. 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.

OBJECTIVE

DNA SEQUENCING

A genome contains the complete set of DNA including all the genes. And this includes all the information required to build and uphold that organism[23]. A great amount of Genetic material is similar between organisms [13]. Understanding the relationship of an unknown gene or DNA sequence to known sequences is the key to assigning its function.

The main theme of interest for a Computer Science Researcher in the field of Genetics can be given as under:-

The Nucleotide Bases are used to identify the characteristics of all the species. The Nucleotide Bases [16] comprise of the following elements:

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

It is agreed that we have a big, monolithic anthology of heterogeneous and distributed databases containing the Nucleotide Base Sequences of many species. But, the size and divergence of this raw

information makes it very difficult for the Biologists to search for a “Pattern of some sequence” in these databases. The Biologists or Life Science Experts need to search for a specific sequence in the existing sequences stored in the databases. This pattern matching is done in order to find homologous patterns. This is helpful because if suppose there is a mutation in DNA of an organism and this mutated DNA is the reason for some problem eg. a type of cancer in that organism. Now if this mutated pattern matches with some other organism, and that organism got treated with some enzyme, the research can be directed to that hormone and can be used to remedy the cancer in the original organism. The DNAM project is taken as a case study for comparison of Agile with crowdsourcing and traditional prototyping also assisted by crowdsourcing.

CROWDSOURCING WITH TRADITIONAL PROTOTYPING

The work of Prototyping with crowd sourcing begins with identification of goals that need to be accomplished. The goal diagram is the best pictorial representation for what needs to be done at different levels of abstraction. 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 goals to the identified Roles as shown in the figure 2.

Now the process of assigning Roles to various identified crowd sourcing worker needs to be executed.

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 take care of platform transparency such that information from heterogeneous data sources can be extracted ad the DNA Pattern be searched.

This is shown in figure 3.

Goal Diagram

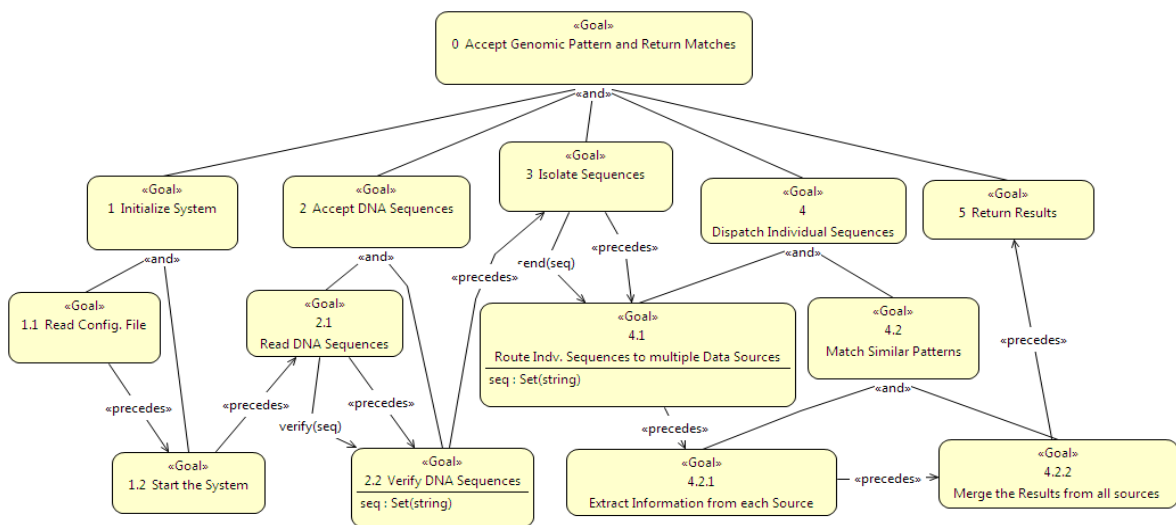


Fig. 1 DNAM Goal Diagram

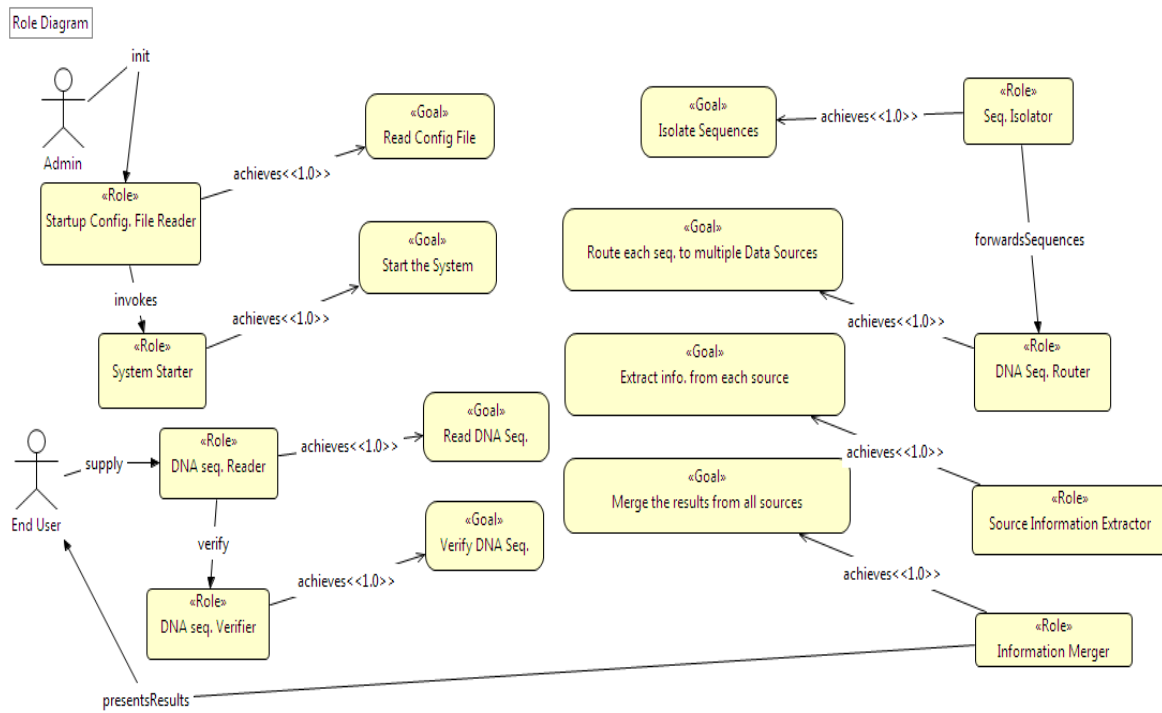


Fig. 2. DNAM Role Diagram

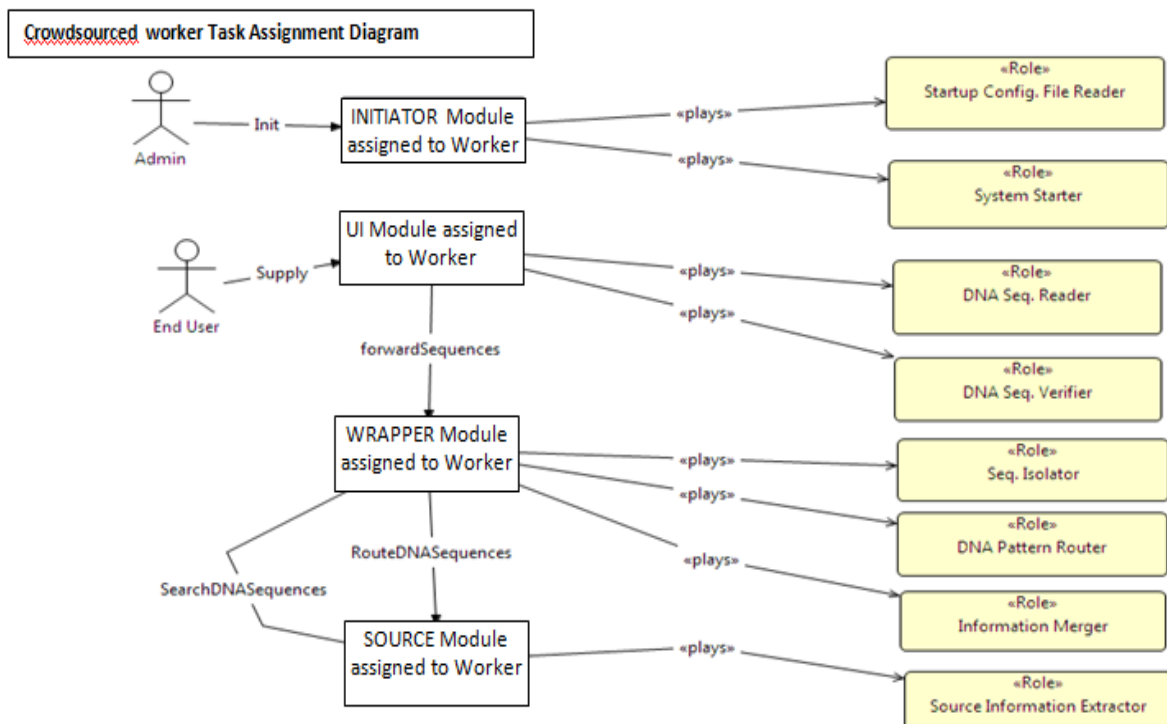


Fig. 3. DNAM CrowdSourced Worker Task Assignment Diagram

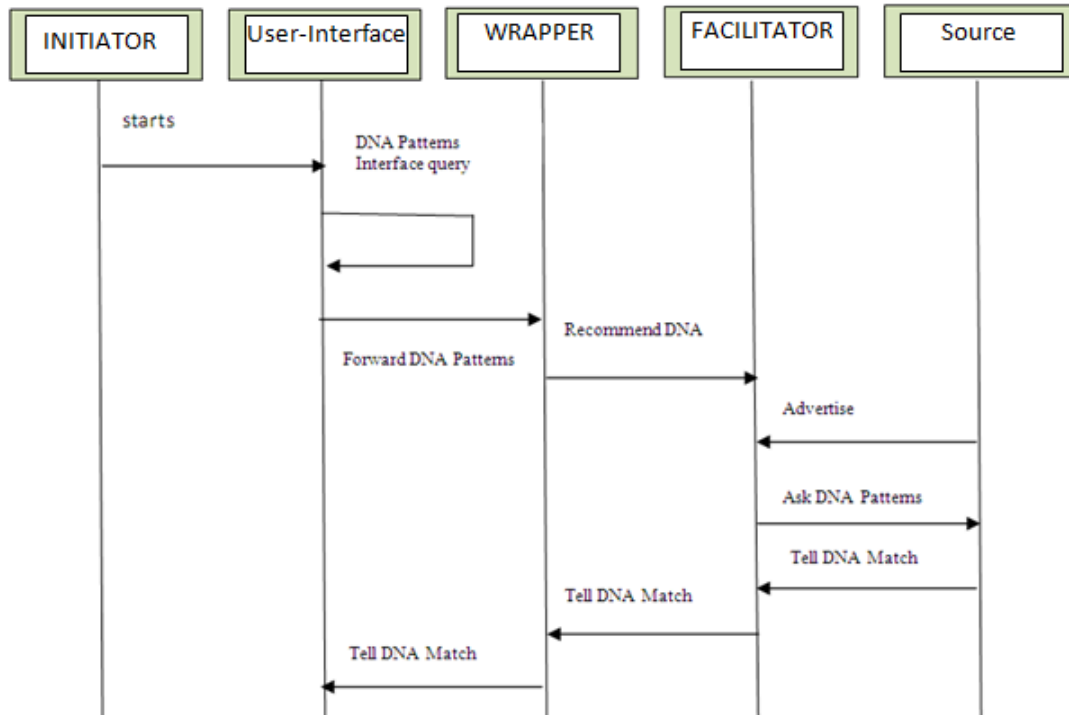


Fig. 4. Interactions and Messages Exchanged Between Modules

Plan Diagram: Add Goals Here

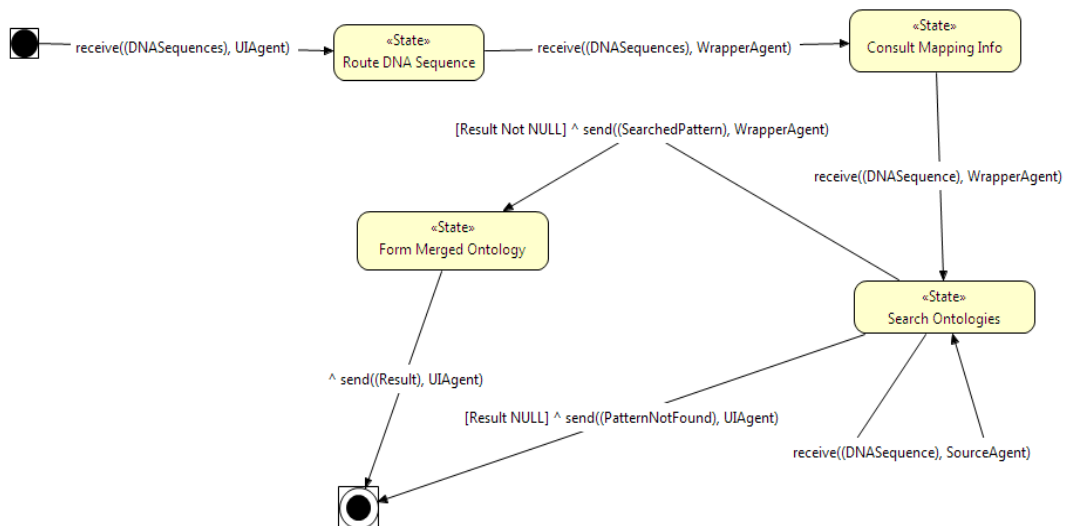


Fig. 5. DNAM State Diagram

2. COST ESTIMATION MODEL COCOMO II in Traditional (Prototyping) Software Development approaches combined with CrowdSourcing

There are many mathematical formulae used to forecast Effort required in developing a Software System in different methodologies like Waterfall, Spiral or Incremental and Agile Approaches. I have used COCOMO II Early Design Model [15] 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.

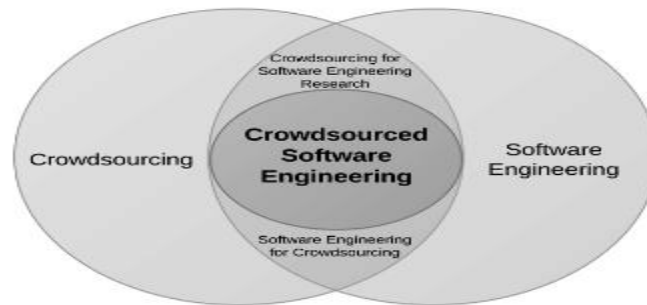


Fig. 6. Coagulation of Traditional Software Engineering and CrowdSourcing

The diagrams are refined from Highest Level of Abstraction to Lowest Level of Abstraction. High Abstraction deals with Goals and Plans and Lowest deal with the State Machines and coding and implementation.

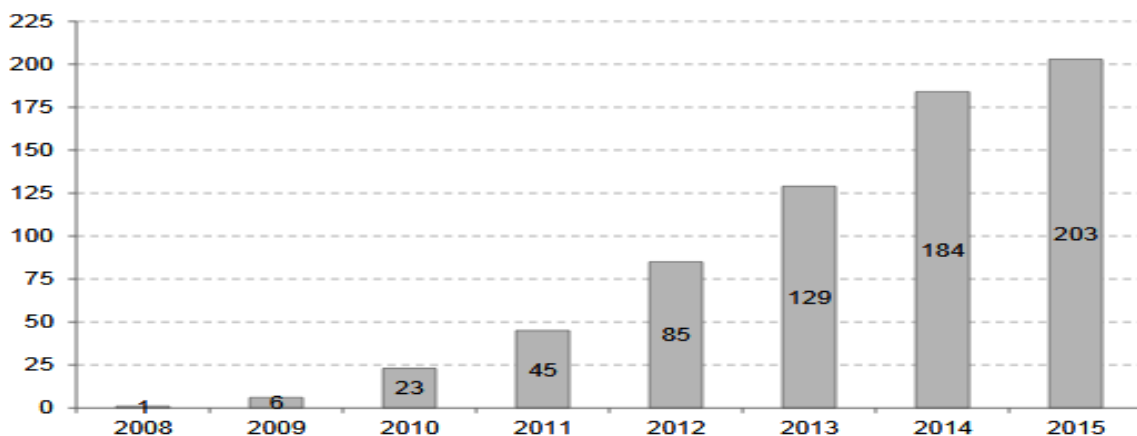


Fig. 7. Cumulative growth of Crowdsourced Software Engineering studies published before April 2015.

2. 1 Effort Estimation

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

2.1.1 DNAM through Prototyping (Traditional SDLC)

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

A = 2.94 (for COCOMO II.2000)

(1)

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

B = 0.91 (for COCOMO II.2000)

(2)

The values and calculations provided in the charts shown under represent the analysis done on Agile Development techniques involving crowdsourcing.

Table 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 interchanged	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 2

Statistical Data About Functional Behaviour

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 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 Resolution	Low	6.24
Team Cohesion	Low	2.19
Process Maturity	Low	6.24
		18.110

2. 1.2 Scale Factors

The exponential part of the application is governed by five scale factors (SF) that describe relative economies or diseconomies of scale. A project has economies of scale if the exponent is less than 1. if Exponent=1 then Economies and diseconomies of scale are in balance. If the exponent is more than one the project has diseconomies (Boehm 2000:30).

Table 4
 Cost Drivers

Cost Drivers	Range(Prototype)	Value(Prototype)
Product Reliability and Complexity	High	1.10
Reusability	Nominal	1.00
Platform difficulty	Low	1.10
Personnel Capability	Nominal	1.00
Personnel Experience	Nominal	1.00
Facilities	Nominal	1.00
Required Development Schedule	Low	1.14 IIVal=1.38

In addition to the scale factors there are other relevant factors that affect the efforts done by the developer, called cost drivers [15]. These are:

- 1.Product Reliability and Complexity (RCPX) is the combination of Software Reliability (RELY), Database Size (DATA), Software Complexity (CPLX) and Documentation (DOCU).
- 2.Reusability (RUSE)
- 3.Platform Difficulty (PDIF) is the combination 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) combines Analyst Experience (AEXP), Programmer Experience (PEXP) and Language and Tools Experience (LTEX).
- 6.Facilities (FCIL) combines 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 5
 Equivalence Of Each Elements 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:

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

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

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

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

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

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

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.

3 Analysis and Estimation of DNAM project using Agile with Crowdsourcing

3.1 Terminology

3.1.1 User Story

A "User Story" is a simple statement about what a user wants to do with a feature and the value the user will gain from that feature.

1. The User
2. What the user wants to do with a feature
3. Value gained by the user
4. Acceptance Criteria
5. Front and back of a card

3.1.2 Story Points

A story Point represents a value given to a user story that is used to measure the effort required to

implement the story.

It is a number that represents story size based on how hard a story is to implement.

3.1.3 Story Point Velocity

Velocity measures how much of something can be achieved over a fixed period of time; e.g. how many Story Points are completed during a Sprint.

3.1.4 Simple Buildup Approach for Agile Project Cost Estimation

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

Iterations needed = Total SP / Velocity2

In most cases a story point uses one of the following scales for sizing:

- 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

3.1.5 Crowdsourcing the Agile Software Development

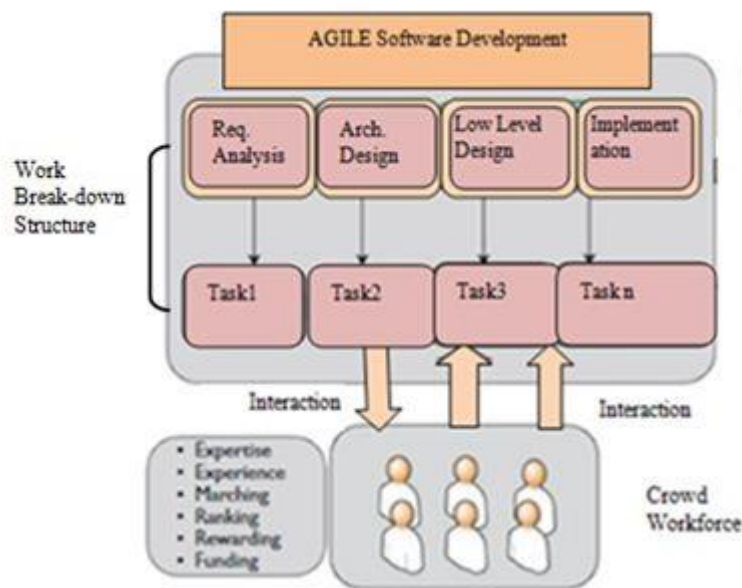


Fig. Agile with Crowdsourcing

The process of crowdsourcing Agile Software development begins with the construction of Work Break-down Structure, also called Task Reduction. Figure 2 shows this process. 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.

3.2 Identifying Stories

<p>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>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>Forward Pattern to Wrapper module SP=2</p> <ul style="list-style-type: none"> > Route patterns to source modules. > Refer ontology > Merge results > Return matched results back to UI module 	<p>Receival of Pattern by Source Module SP=2</p> <ul style="list-style-type: none"> > Convert local database to XML > search local repository > Find matches > Pass matched DNA content to Wrapper Module
<p>User Interface SP=1</p> <ul style="list-style-type: none"> > Accept matched DNA sources > Return matched DNA sources list to the end user > Ready to accept new DNA patterns as query 	

The authors have developed first two User Stories.
 By equation 1 and 2,

$$\text{Velocity}(V)=2/2=1;$$

$$\text{Total Iterations needed}=\text{Total SP}/V=7/1=7;$$

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,

$$\text{Time consumed by Prototype with Crowdsourcing}=\frac{21}{4}=\mathbf{5.25\ months}$$

This represents a significant increase in efficiency of the Agile developers in crowdsourced 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. Let us say, costs through scheduling to be CS.

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.

4. Conclusion and Future Scope

This paper tries to apply Effort Forecasting in Genomic Information Retrieval (DNAM) taken as case study for comparing Traditional and Agile System currently developing under crowdsourcing. The authors predicted 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 amalgamating crowdsourcing. The estimation of Agile with crowdsourcing is done using Simple Buildup Approach. In this paper the authors have tried to calculate the Effort required for a developer in manifesting the DNAM project thereby projecting the overall time taken in completion of the project by both software development methodologies.

References

- [1] Building Custom, Adaptive and Heterogeneous Multi-Agent Systems for Semantic Information Retrieval Using Organizational-Multi-Agent Systems Engineering, O-MaSE, IEEE Explore, ISBN: 978-1-5090-3480-2, Gaurav Kant Shankhdhar, Manuj Darbari.2016.
- [2] O-MaSE: A customisable approach to designing and building complex, adaptive multi-agent systems, Scott DeLoach, <https://www.researchgate.net>, 2016.
- [3] Bücheler, T., and Sieg, J. H. "Understanding science 2.0: crowdsourcing and open innovation in the scientific method," *Procedia Computer Science* (7) 2011, pp 327-329.
- [4] Chen, L., and Liu, D. "Comparing Strategies for Winning Expert-rated and Crowd-rated Crowdsourcing Contests: First Findings," 2012.
- [5] Chen, Z., and Luo, B. 2014. "Quasi-Crowdsourcing Testing for Educational Projects," *Companion Proceedings of the 36th International Conference on Software Engineering: ACM*, pp. 272-275
- [6] Doan, A., Ramakrishnan, R., and Halevy, A.Y. 2011. "Crowdsourcing Systems on the WorldWide Web," *Communications of the ACM* (54:4), pp. 86-96.
- [7] Geiger, D., Seedorf, S., Schulze, T., Nickerson, R. C., and Schader, M. "Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes," *AMCIS*, 2011.

- [8] Graber, M. A., and Graber, A. "Internet-based crowdsourcing and research ethics: the case for IRB review," *Journal of medical ethics* (39:2) 2013, pp 115-118.
- [9] http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf
- [10] H. Pandey, V.K Singh, "A Fuzzy logic based Recommender System for E- learning system with multi agent framework", *International Journal of Computer Applications* (0975-8887), Volume-122 No. 17, July 2015.
- [11] H. Pandey, S. Kumar, V.K. Singh, "A Study of the pertinence of Crowdsourcing in Agile Software Development", *IJRECE* Vol. 5 Issue 4 Oct. - Dec.2017, ISSN: 2393-9028 (Print), ISSN: 2348-2281(Online).
- [12] H. Pandey, V.K. Singh, "LR Rotation rule for creating Minimal NFA", *International Journal of Applied Information Systems (IJ AIS – ISSN: 2249-0868 Foundation of Computer Science FCS, New York, USA.*
- [13] H. Pandey, V.K. Singh, "A new approach for NFA minimization", *International Journal of Applied Information Systems (IJ AIS – ISSN: 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 8– No.3, February 2015.*
- [14] H. Pandey, S. Kumar, V.K. Singh, "Enhancing efficiency of the agile software by using crowdsourcing", *IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org ISSN (e): 2250-3021, ISSN (p): 2278-8719 Vol. 08, Issue 4 (April. 2018), ||VII|| PP 71-77.*
- [15] H. Pandey, S. Kumar, W. Ahmad, V.K. Singh, "Enhancing efficiency of the agile software by using crowdsourcing", *International Journal of Advanced Research in Engineering and Technology (IJARET) Volume 9, Issue 2, March – April 2018, pp. 68–76, Article ID: IJARET_09_02_009 Available online at <http://www.iaeme.com/ijaret/issues.asp?JType=IJARET&VType=9&IType=2> ISSN Print: 0976-6480 and ISSN Online: 0976-6499.*
- [16] M. A. Awad "A Comparison between Agile and Traditional Software Development Methodologies", *School of Computer Science and software Engineering, The University of Western Australia, 2005.*
- [17] J. Howe, "Crowdsourcing: A definition," <http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing.a.html>, June 2006.
- [18] "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.
- [19] K. R. Lakhani, D. A. Garvin, and E. Lonstein, "TopCoder(A): Developing software through crowdsourcing," *Harvard Business School Case*, 610-032, January 2010.
- [20] T. D. LaToza, W. Ben Towne, A. van der Hoek, and J. D. Herbsleb, "Crowd development," in *Proceedings of the 6th International Workshop on Cooperative and Human Aspects of Software Engineering*, May 2013, pp. 85–88.