



AGILITY: THE NEED OF AN HOUR FOR SOFTWARE INDUSTRY

Tushar Sambare
Research Scholar,
Department of Computer Science and Engineering,
Shri JJT University, Jhunjhunu, Rajasthan, India

Gulabchand K. Gupta
Department of Computer Science and Engineering,
Shri JJT University,
Jhunjhunu, Rajasthan, India

Abstract: The need of the current Software industry is changing rapidly, and the demand is ever increasing. The industry needs to provide the solution to this demand. As the software development proceeds, factors such as requirements, needs, priorities, underlying technology may change. Thus development process must be highly dynamic and a good software development methodology must adapt to these evolving and changing requirements. Traditional software development models are unable to handle such dynamic requirements. However, there are many new development models introduced in context to provide satisfactory solutions to increasing needs of the industry. Comparison between different new software development methods will help in the selection of appropriate development model in a particular scenario.

Keywords: CMMI, Agile Development, Scrum, XP, FDD, Crystal

I. INTRODUCTION

As a solution to the present developing difficulties in the programming industry, a wide range of new methodologies of SW development are introduced. Business exercises are quickly changing these days and there are progressively critical prerequisites set on programming paradigms. This puts traditional programming advancement strategies behind and prompts the requirement for various methodologies. Most present-day advancement procedures can be eccentrically portrayed as agile.

Agile SW development refers to a bunch of programming improvement procedures in based on iterative advancement, where prerequisites and arrangements develop through a joint effort among self-organizing cross-functional groups. Agile procedures are disciplined that team work and adjustment, an authority theory that energizes collaboration, self-association, and responsibility, an arrangement of designing accepted procedures expected to take into consideration fast delivery of top quality SW, and a business approach that adjusts improvement to client needs and organizational objectives [7].

In English, Agile signifies 'capacity to move rapidly and effectively' and reacting quickly to change – this is a key part of agile programming improvement also. These techniques give distinctive methods for creating SW [7]. In loads of cases they turned out to be more fruitful than conventional ones. In the subsequent sections some theses methodologies are discussed [11].

The research paper presented here compares the usefulness and applicability of the current software development methodology in context of the current industry needs.

II. CAPABILITY MATURITY MODEL INTEGRATION

CMMs holds the fundamental components of powerful procedures. These components depend on the ideas created by Crosby, Deming, Juran, and Humphrey.

The primary model to be created was the CMMI for Development (at that point basically called "CMMI"). [6].

To evolve and enhance the models for organizations three distinctive CMM models were joined to the coordinated one, CMMI.

At first, CMMI was alone model that joined three source models:

- The CMM for Software (SW-CMM) v2.0 draft C
- The Systems Engineering Capability Model (SECM) [EIA 2002a]
- The Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98.

These three main models were chosen in light of their effective appropriation or promising way to deal with enhancing forms in an association.

A. The Concept of CMMI

Continuous Representation and Staged Representation: Organizations need to look over one of two ways to take after when beginning utilizing CMMI, the continuous portrayal or the staged portrayal. The two ways offer a marginally unique way to deal with go up against CMMI [6]. On picking the ceaseless portrayal there is a great deal of ease and getting things done in various rates, and is appropriate if it is clear which forms that are risky. With the nonstop portrayal, the association is experiencing ability levels.

The Model Components: The model parts of the CMMI are gathered into classes that reflect how they should be translated; there are required, expected and enlightening segments. The required segments in the CMMI are the particular and non-exclusive objectives that speak to what an association must do to fulfil a procedure zone. Expected segments incorporate the nonexclusive and particular practices and are a manual for what an association need to execute to accomplish the particular and non-specific objectives. Enlightening parts are point by point data on, for instance, work items, sub hones, intensifications, non-specific practice titles, objective and practice notes, and references.

B. Advantages of CMMI

There are various advantages of executing CMMI in an IT/Software Development Organization, some of these advantages are recorded beneath:

- Culture for keeping up Quality in ventures begins in the brain of the normal developers to the senior software engineers and undertaking administrators.

- Centralized QMS for usage in activities to guarantee consistency in the documentation which implies less learning cycle for new assets, better administration of task status and wellbeing.
- Incorporation of SW Engineering. Best Practices in the Organizations as portrayed in CMMI Model.
- Cost sparing as far as lesser effort because of less deformities and minimum rework [8].
- This likewise brings about expanded Productivity.
- On-Time Deliveries
- Making customer happy by providing proper product.
- Overall expanded ROI.
- Decreased Costs
- Improved Productivity

C. Disadvantages of CMMI

- CMMI-DEV is may not be reasonable for each association.
- It might include overhead regarding documentation.
- May require extra assets and information required in littler associations to start CMMI-based process change.
- May require a lot of time and effort for execution [8].
- Require a noteworthy move in hierarchical culture and state of mind.

III. AGILE MODEL

The term Agile signifies 'moving rapidly'[4]. The Agile process itself is a product improvement process by smaller groups, in a brief timeframe design, and including framework clients and additionally engineers [4]. This coordinated procedure is an iterative approach in which consumer loyalty is at most elevated need as the client has direct association in assessing the product.

The AGDM emphasizes on four important values:

1. Individual and group associations over procedures and tools.
2. Working SW over far reaching documentation.
3. Customer joint effort over contract specifications.
4. According to plan respond to changes.

A. Principles

What the originators of the agile practices held in like manner was an arrangement of qualities they mutually distributed as the Manifesto for Agile Software Development. The twelve key purposes of characterized in the Agile Manifesto are:

1. Satisfy client through right on time and persistent cycles [11].
2. Deploy first cycle inside couple of weeks and the entire programming inside couple of months.
3. Customer and Agile groups must work together day by day all through the task.
4. Agile group and client must have direct gatherings [18].
5. Accept requirements even in late periods of the framework development.
6. Trust and regard must be kept up among coordinated colleagues.
7. Velocity of the task must be measured after conveyance of every increment.
8. Emphasis ought to be on great plan to build agility.
9. Best engineering and configuration dependably turn out from self-association.

10. Adjust and tune as indicated by the circumstance.
11. Whole advancement process must take-off after making it simple (KIS) rule.
12. Agile undertaking needs predictable work until finalization [18].

The most vital of these standards is: "The most productive and viable strategy for passing on data to and inside a development group is direct discussion.

B. Advantages of Agile Model

- Adaptive to the evolving conditions [15].
- Agile accelerates the SDLC stages and sidesteps process steps that enhance the value to product development.
- Involves the partners ceaselessly with the goal that the new necessities are assembled quicker and there is no extension for guess work by the groups [13].
- Saves cost, time and endeavours by following iterative incremental work completion and in this way recognizing deviations early.
- Very less documentation is needed [8].
- Provides the final product of higher nature of the product delivered and highly happy client.

C. Disadvantages of Agile Model

- Time taking and wastage of assets due to steady difference in necessities.
- More supportive for administration than engineer.
- Only senior engineers are in a superior position to take the choices important for the deft sort of advancement.
- Once groups become large, this technique start to fail, as they don't scale to substantial groups, or groups spread crosswise over geologies.
- If the ventures are big, then it is hard to judge the endeavours and the time required for the undertaking in the SDLC.

IV. EXTREME PROGRAMMING (XP)

XP is a lightweight strategy for small to medium groups creating SW despite obscure or quickly evolving requirements. XP is easy and trained way to deal with programming improvement. XP emphasis on customer satisfaction [1]. An extraordinary endeavour to significantly improve the way toward creating SW frameworks is made concentrating on what delivers value: the prerequisites for the framework or the code that actualizes the framework [2]. Prerequisite determination as User Stories, code improvement by sets of engineers (Pair Programming), rearrangements of the code through Refactoring and watchful testing are the extraordinary highlights of XP method. XP enhances a product venture in four basic ways; correspondence, straightforwardness, criticism, and courage. XP has revived the idea of evolutionary outline with rehearses that enable evolution to end up noticeably a practical plan methodology [4].

A. XP Core Practices

The Planning Game: Business and improvement cooperate to create the most extreme business esteem as quickly as could reasonably be expected. The planning game occurs at different scales, yet the essential principles are the same [1] [3].

Small Releases: XP groups rehearse small outputs in two critical ways: First, the group releases running, tested programming, conveying business esteem picked by the Customer, each

cycle. The Customer can utilize this product for any reason, regardless of whether assessment or even release to end clients. **Straightforward Design:** XP utilizes the most straightforward conceivable plan that takes care of business. The necessities will change tomorrow, so just do what's expected to meet the present prerequisites. Plan in XP is not a one-time thing but rather an all-the-time thing [21].

Metaphor: XP groups build up a typical vision of how the program functions, which we call the "illustration" or "Metaphor". Taking care of business, the illustration is a basic suggestive depiction of how the program functions.

Continuous Testing: XP groups concentrate on approval of the product consistently. Developers create SW code by composing tests initially, and afterward code that satisfies the prerequisites reflected in the tests. Clients give acknowledgment tests that empower them to be sure that the highlights they require are given.

Refactoring: XP group Refactor out any copy code created in a coding session. Refactoring is rearranged because of broad utilization of computerized test cases [3].

Pair Programming: All creation code is composed by two software engineers sitting at one machine. This training guarantees that all code is looked into as it is composed and brings about better Design, testing and better code.

Collective Code Ownership: No single individual "claims" a module. Any engineer is relied upon to have the capacity to take a shot at any piece of the code-base any time.

Continuous Integration: All changes are coordinated into the codebase, in any event, every day. The unit tests need to run 100% both previously, then after the fact reconciliation. Occasional combination prompts major issues on a product venture. Most importantly, despite the fact that integration is basic to deliver great working code, the group is not rehearsed at it, and frequently it is appointed to individuals who are not familiar with the entire framework [21].

40-Hour Work Week: Software engineers go home on time. In crunch mode, up to one week of extra minutes is permitted. In any case, various back to back a long time of additional time are dealt with as a sign that something is off with the procedure as well as calendar.

On-site Customer: Programming group has constant access to the client who will really be utilizing the framework. For activities with loads of clients, a client agent (i.e. Product Manager) will be assigned for Development group get to.

Coding Standards: Everybody codes to similar benchmarks. The specifics of the standard are not critical: what is vital is that entire code looks natural, in support of aggregate possession [1]

B. Advantages of Extreme Programming Methodology

- XP techniques emphasise on client association.
- XP builds up sound designs and plans and to get the engineers by and by focused on their timetables which are unquestionably a major favourable position in the XP model [9].
- XP is reliable with most current advanced development strategies along these lines, engineers can deliver quality programming [14].
- It concentrates on client inclusion.
- Establishes reasonable plans and schedules [15].
- Developers are particularly dedicated to the task.
- Equipped with advanced techniques for quality programming.

C. Disadvantages of Extreme Programming Methodology

- This philosophy is just as viable as the general population included, Agile does not eliminate this issue.
- This sort of programming advancement model requires gatherings at visit interims at enormous cost to clients [21].
- It requires excessively improvement changes which are truly exceptionally hard to adapt each time for the product designer.
- In this technique, it watches out for difficult to be known correct estimates of work exertion expected to give a quote, on the grounds that at the beginning of the venture no one aware about the whole degree and necessities of the task.
- Effectiveness relies upon the general people included [9].
- Requires visit meeting for improvement raising aggregate expenses.
- Necessitates for excessive advancement changes.
- Exact conceivable outcomes and future results are truly unknown.

V. SCRUM

Jeff Sutherland made the scrum procedure in 1993; he utilized the expression "scrum" from an investigation established forth in a recent report by Takeuchi and Nonaka, distributed at Harvard Business Analysis [4]. It is the main deft advancement approaches. The Scrum Association changes the way we handle complex SW projects.

Scrum is known as lightweight procedure system for flexible SW development, and the most broadly utilized one. Scrum is a one of the part of agile strategy. A "product framework" is a specific arrangement of strategies that must be followed all together for a procedure to be reliable with the structure [4]. "Lightweight" implies that the overhead of the procedure is kept as little as could reasonably be expected, to boost the measure of work done.

An agile form of Scrum process benefits the organization by helping it to...[16].

- Better finished result
- Easy to change and apply
- Takes less time and create better estimates
- Has better control on project plan
- The primary practical unit of scrum is a working group. The group does not lead by a specific appointed team leader. Nobody chooses who will do what. The issues are handled by the group as a whole [10].

Things with respect to status of the project task, issue identified with the tasks, work that is done since last meeting, and work that needs be done before next meeting (i.e. past, current and future) are examined each day [16]. The Team may comprise of 5 to 9 individuals. These meetings are time boxed and conducted not more than 15 minutes. Scrum does not characterize exactly what kind of requirements are to take, yet essentially says that they are assembled in the Product Backlog, and referred to nonexclusively as "Product Backlog Items," or "PBIs" for short [5].

A. Advantages of Scrum

- The real favourable position of User Story lies in the client driven definition itself. This is on the grounds that, eventually, the client will be utilizing the item in the important client situations. It associates the end clients to the colleagues [19].

- The sentence structure of the User Story itself guarantees to catch the objective or advantage or esteem that the client needs to accomplish [20].
- Since the acknowledgment criteria frames some portion of client story itself, it will be an additional favourable position to the Scrum Team [10].
- It is conceivable to make improvements to a client story in course of the execution of the undertaking. On the off chance that the extent of the client story turns out to be vast, it should be part into littler client stories.

The conditions in the acknowledgment model can likewise be changed. As working item increments are conveyed to the clients toward the finish of each sprint, the scrum group can get criticism from the clients in run survey meeting [19]. This empowers consolidation of criticism into the item ceaselessly.

B. *Diadvantages of Scrum*

- For extensive undertaking, once in a while it winds up noticeably hard to assess the exertion required [20].
- The plan is less absolutely said.
- The venture may go toward another path if client agent does not have clear thought regarding the necessity [21].
- Just senior software engineers can take choices regarding the improvement procedure.

VI. FEATURE DRIVEN DEVELOPMENT (FDD)

For making quick functionality from SW, FDD can serve the purpose could be the key. FDD revolves around fast improvement cycles and provides organizations with include rich frameworks since they are always creating. The FDD was suggested by Jeff Luca in 1997 to meet the SW product development needs of a Singapore bank. His idea was a gathering of five procedures intended to cover the model's improvement and furthermore it's listing, outline, arranging and the working of its features [12].

Since its unique implication, FDD, and its five essential exercises have ceaselessly been utilized to create enterprise SW since it is viewed as both agile and practical oriented [20]. When it is delivered well, FDD can offer auspicious status reports and precise advance following in light of all levels of authority in the undertaking.

A. *Five processes of FDD*

1. **Build up a general model:** The FDD strategy demands that groups apply the sufficient measure of input toward the beginning of the task so as to build an object procedure featuring the domain issue [21]. Demonstrating with FDD is time-boxed and community oriented. Domain models ought to be made in detail by small gatherings and afterward introduced for associates to survey. It is trusted that a proposed model – or conceivably a blend of them – will then be utilized for every zone of the domain. They will then be compiled after some time to create a general model.
2. **Prepare a feature list:** From the experience during the product building procedure, a rundown of highlights is set up by separating domains into branches of knowledge that contain data on business exercises. The means that are utilized for every business action speak to an arranged list of features. Highlights are commu-

nicated as: "activity, result, and question". The desire is that they won't take over two weeks to finish, on the off chance that they do, they ought to be broken into smaller tasks [20].

3. **Design by features:** Once the component list has been built up, the accompanying procedure includes appointing the different feature to the SW engineers.
4. **Outline by feature:** Coders at that point deliver a plan bundle for each component with a central SW engineer choosing a bunch of features that ought to be produced inside a two-week time span. The main developer will likewise build up point by point outlines for each element while refining the model. At the point when this is finished, prologues are delivered, and an outline inspection is done.
5. **Make by feature:** Once outline investigations are finished, designers design an action for each element and build up the code for their individual classes. At the point when the investigation is finished and a unit test completed, the component is then pushed to the main product build.

B. *Advantages of FDD Methodology*

- Lads to move to huge ventures and acquire repeatable achievement.
- Practicing the five procedures gets new staff with a shorter increase time [19].
- Feature-Driven Development is worked around a centre of industry-perceived prescribed procedures.
- Regular Builds: Regular Builds guarantee there is dependably an updated framework that can be shown to the customer and helps featuring incorporation mistakes of source code for the functions early.
- Visibility of progress and results: By visit, fitting, and exact progress revealing at all levels inside and outside the venture, in view of finished work, administrators are helped at directing a task effectively [21].
- Risk Reduction by means of emphasis of outline and work in little pieces. FDD helps in minimizing risks by utilizing shorter cycles of planning, comprehension of the necessities and the framework in an unmistakable and particular way, in this manner prompting a state where there are no ambiguities, as the requirements and desires are as of now saw exceptionally well [19].
- Clarity of necessities and better understanding of framework to be built is increased through the Develop Overall Model process. This procedure incorporates abnormal state walk-through of the extent of the framework and its unique situation. Next, in depth domain walkthroughs are held for each modelling region.
- Costing the task by include prompts more noteworthy accuracy.

C. *Disadvantages of FDD Methodology*

- Not a perfect technique for smaller tasks along these lines, it is bad for an individual SW developer [19].
- High reliance on the main engineer implies the individual ought to be completely prepared for a go about as organizer, lead planner, and mentor.
- No drafted documentation given to customers in this philosophy in this way, they are not ready to get a proof for their own particular SW [21]

VII. CRYSTAL METHODOLOGY

The Crystal system is a standout amongst the most light-weight, versatile ways to deal with SW development. Crystal is really included a group of coordinated approach, for example, Crystal Clear, Crystal Yellow, Crystal Orange and others, whose exceptional attributes are driven by a few factors, for example, group measure, framework criticality, and task needs[12]. This Crystal family tends to the acknowledgment that each project may require a marginally custom fitted arrangement of approaches, practices, and procedures keeping in mind the end goal to meet the undertaking's extraordinary attributes.

A. Process Categories of Crystal Methodologies

- Four levels of criticality have been characterized, in view of what may be lost as a result of a disappointment in the delivered framework
 - Comfort (C)
 - Discretionary Money (D)
 - Essential Money (E)
 - Life (L)
- The most number of individuals that may need to get engaged with a task is viewed as the measure of the venture's size.
- A class L40 venture is a task including up to 40 individuals building up an existence basic framework.

B. Advantages of Crystals Methodology

- Iterative-incremental process
- Continuous reconciliation [12]
- Iterative improvement engine administered by arranging and considering
- Flexible and configurable process
- Methodologies utilized for a low – criticality undertaking can normally be tuned to fit a higher-criticality venture, gave that the task measure is not expanded significantly
- Active client association

C. Disadvantages of Crystals Methodology

- Only restricted adaptability
- Lack of an unambiguous regular process
- Limited applicability, not reasonable for growing exceedingly basic frameworks
- Over-dependence on human correspondence

VIII. COMPARATIVE ANALYSIS OF CURRENT DEVELOPMENT METHODOLOGIES

The modern methodologies for development of SW follows some good principles from the traditional methodologies. All the modern SW development approaches are either iterative or incremental. In some cases, like SCRUM and XP, they follow both Incremental as well as Iterative strategy.

These model follows different time scale for their iteration life cycle as XP takes 1-6 weeks, SCRUM takes 2-4 weeks, and the FDD takes from 2 days to 2 weeks.

As per the need of current SW process requirements the product can large as well as complex. With respect to this

measure, the XP methodology can deal with small and simple projects. The Scrum strategy can be used with big and critical problems. The crystal approach is applicable to any size project as the scope for human correspondence is more. In CMMI strategy, it can handle almost any size project but the use of extra resources and due to extra documentation it is burdening process for large size problems.

There is very much important factor of involving the user in SW building process, which is must in current development scenario, it is addressed in most of the current development strategies. In the XP and in the Crystal approach customer is enthusiastically involved in almost every phase of development. Which is helpful to build a fully functional quality product. In SCRUM the customer or the end user is not directly involved. The feedback from the customer is obtained through product owner. In FDD approach customer interaction happens through the reports. In CMMI approach customer is involved at the time of requirement gathering only.

The documentation is again an essential part for follow up during the maintenance as well as for guiding next venture. The XP, the SCRUM and the Crystal follows only basic documentation. The CMMI, the FDD approach follows high level of documentation, which sometimes become time consuming process result in delay to release the end product.

In regards with models work process practices all the current approaches have different style of practices. The XP approach mainly achieved with simple steps, programming in pair and by following test driven approach. In SCRUM approach complete work is formed and done through regular meetings. In FDD approach object modelling is used. The functionality is accomplished through feature driven approach. For architecture design purpose the UML is used. The CMMI is having continuous and staged representation. There are many versions of CMMI, which are applicable on various types of SW associations. The CMMI majorly keeps the exact track of each activity with proper documentation. The crystal approach comes in variety options which is has more human correspondence of work. As the task size increases the Crystal team is also increases in size.

In all the modern SW development technologies one point is common that, these all models support concurrent functionality development. Almost in all the modern development approaches the requisition from the clients are acquired in iterative manner. As the approach is iterative and sometimes incremental the cost for the reworks is less. The flexibility in designing the architecture is easily achievable. There is no rigidity in all the approaches, which lets computer programmer to concentrate on features than the whole process. These all are very flexible processes in terms for directions for development. The modern approaches have better processes to address the issues of bugs, the processes are mature and capable to catch the faults in functionality in earlier phases.

The modern approaches believe on continuous testing. Almost in every approach testing is performed after every iteration. These approaches stresses on interpersonal skills and required the basic business knowledge among the working group, so that the particular domain requirement can implemented with full expected functionality exactly as per the requirement by the client. These model approaches support reusability up to larger extent. By using these features of modern approaches of SW development organisation can achieve high client satisfaction and high performance.

However, as there are many advantages of modern approaches for the project development, there is downside also. One the problem with modern approaches is these models are not suitable for big and critical level projects. As the criticalness of venture increases these models face difficulty in implementation. This issue can be conquered by avoiding some problematic practices and introducing some new practices in these model processes.

IX. CONCLUSION

The research paper discussed and compared six new development models. Agile SW procedures are picking up prominence and are presently favoured over traditional SW programming strategies which have a few deficiencies, for example, failure to adapt up to the always showing signs of change client necessities and surpassing the allocated time and spending plan budget. However as per the analysis of the comparison made in above section, the XP can be a good option for satisfying current industry needs. Still there are shortcomings in this model which need to be addressed. All the above model proved their usefulness in specific domain or specific project size. There is a need to have a comprehensive model for the development. According to the study performed in this research paper the XP approach can turn in to slandered model if some the process modified with some better option.

X. REFERENCES

- [1] Radmila Juric, "Extreme Programming and its Development Practices", International Conference on Information Technology Interfaces, ITI 2000, 13-16 June 2000, ISBN : 9539676916.
- [2] Pekka Abrahamsson and Juha Koskela, "Extreme Programming: A Survey of Empirical Data from a Controlled Case Study", ACM-IEEE International Symposium on Empirical Software Engineering , ISESE, 2004.
- [3] Kai Stapel, Daniel Lübke, Eric Knauss, "Best Practices in eXtreme Programming Course Design", International Conference on Software Engineering, 2008, ISBN: 978-1-60558-079-1/08/05.
- [4] MALIK HNEIF, SIEW HOCK OW, "REVIEW OF AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT", International Journal of Research and Reviews in Applied Sciences, Volume 1, Issue 1, October- 2009, EISSN: 2076-7366.
- [5] MALIK HNEIF et. Al. , "Review Of Agile Methodologies In Software Development", International Journal of Research and Reviews in Applied Sciences, Volume 1, Issue 1(October 2009), ISSN: 2076-734X.
- [6] Therese Clara V, Alagarsamy Dr. K., "Extreme Programming versus CMMI – Conflicts and Compatibilities", IJCSET , June 2011, Vol 1, Issue 5,203-209, ISSN: 2231 -0711.
- [7] Ani Liza Asnawi, Andrew M. Gravell, and Gary B. Wills, "Empirical Investigation on Agile Methods Usage: Issues Identified from Early Adopters in Malaysia", Springer, XP, 2011, e-ISSN 1865-1356.
- [8] Sanjana Taya, et. al., "Comparative Analysis of Software Development Life Cycle Models" IJCST Vol. 2, Issue 4, Oct. - Dec. 2011, ISSN: 0976-8491 (Online) | ISSN: 2229-4333 (Print).
- [9] Seema, Sona Malhotra, "Analysis and tabular comparison of popular SDLC models", INTERNATIONAL JOURNAL OF ADVANCES IN COMPUTING AND INFORMATION TECHNOLOGY, July 2012, ISSN 2277-9140.
- [10] Igor Ribeiro Lima et. Al., "Adapting and Using Scrum in a Software Research and Development Laboratory", Salesian Journal on Information Systems, Vol. 9, pp. 16-23, 2012.
- [11] W.K.S.D Fernando, D.G.S.M Wijayarathne, J.S.D Fernando, M.P.L Mendis, C.D Manawadu, "Emergence Of Agile Software Development Methodologies: A Sri Lankan Software R & D Outlook", INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 2, ISSUE 11, NOVEMBER 2013, ISSN 2277-8616.
- [12] T Bhuvanewari et al., "A Survey on Software Development Life Cycle Models" International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 5, May 2013, pg.262 – 267, ISSN 2320-088X.
- [13] Pankaj Vohra et al., "A Contrast and Comparison of Modern Software Process Models", International Conference on Advances in Management and Technology, 2013, ISSN: 0975 – 8887.
- [14] Anu P Salim et al., "Survey on Different Process Models Used In Software Development", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2855-2860, 2014, ISSN: 0975-9646.
- [15] Mihai Liviu DESPA, "Comparative study on software development methodologies" Database Systems Journal, Vol V No. 3, pp. 37-56, 2014.
- [16] Dr. Mark C. Paulk et. Al., "On Empirical Research Into Scrum", Carnegie Mellon University, 5 February 2014.
- [17] Astha Singhal et. Al., "Scrum: An Agile Method", International Journal of Engineering Technology, Management and Applied Sciences, November 2014, Volume 2 Issue 6, ISSN 2349-4476.
- [18] Asmita et al., "REVIEW ON COMPARATIVE STUDY OF SOFTWARE PROCESS MODEL", International Journal of Science, Technology & Management, Volume No 04, Special Issue No. 01, March 2015, ISSN (online): 2394-1537.
- [19] Shelly, "Comparative Analysis of Different Agile Methodologies", International Journal of Computer Science and Information Technology Research, Vol. 3, Issue 1, pp: (199-203),Month: January - March 2015, ISSN 2348-1196.
- [20] Kiran Hiwarkar et al., "Comparative Analysis of Agile Software Development Methodologies-A Review" Int. Journal of Engineering Research and Applications, Vol. 6, Issue 3, March 2016, pp.80-85, ISSN: 2248-9622.
- [21] Malek Al-Zewairi et al., "Agile Software Development Methodologies: Survey of Surveys", Journal of Computer and Communications, 5, 74-97, 2017, ISSN Print: 2327-5219.