IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

Inside Agile Family: Software Development Methodologies

Rakesh Kumar[1*], Priti Maheshwary[2], Timothy Malche[3]

[1,2,3]Department of Computer Science and Engineering, Rabindranath Tagore University, Bhopal, MP, India

Corresponding Author: rakeshmittan@gmail.com, Tel.: +91-85708-77600

DOI: https://doi.org/10.26438/ijcse/v7i6.650660 | Available online at: www.ijcseonline.org

Accepted: 13/Jun/2019, Published: 30/Jun/2019

Abstract— Software requirements are adapting by the customer to adjust in new environment because business environment is very dynamic in current era. Struggling for appropriate agile processes for development environments of Software developers and project managers is going on till the appropriate process is not matched. Need to adapt in a complex business environment is being faced by organization for helping them in continuous change and transformation. Organization agility is being gaining strategic advantages and market success in these conditions, for maintaining and achieving requirement of agility are agile techniques, architectures, tools, methods and able to react to change requirements in real time. In this research paper various agile family methodologies like AM, XP, Scrum Development, Feature FDD, DSDM, ASD, Kanban, LSD, Scrumban, RAD, Crystal, AUP, DAD has been studied and compared on the basis of various parameters along with their relationship. The research will help future developers to get new ideas about the methods for development along with selection of the right methodology for the product development.

Keywords—Agile, XP, FDD, DSDM, Scrumban, Crystal, AUP and DAD.

I. INTRODUCTION

In this fast life and business competition the many organizations make deployment of Internet based services timely to gain competitive benefits. Now pressure to produce enhanced or new implementations quickly has been increased on developers. Break down of larger projects into small, manageable chunks called iterations are in agile cycle. The product produced during iteration should be able to gain feedback from users. Iterative and incremental development approaches are the base of agile development model where in a highly collaborative manner to produce high quality software in a timely and cost effective manner which allows quickly adaptation of changes by the project. Mitigating and minimizing the overall risk is being helped by the agile and also allows the project to adapting the changes quickly. Like Waterfall Model Agile does not require a requirements freeze upfront. The lightweight methodology agile means moving quickly .According to agile there is a need that every project to be handled in a different manner and methodologies of the current environment need to be tailored to best suitability of the project requirements.

Number of concrete practices supports agile software development along with covering areas like requirements, design, modelling, coding, testing, planning, risk management, process, quality etc. In agile development family, AM (Agile Modeling), XP (Extreme Programming), SD (Scrum Development), (FDD) Feature Driven Development, DSDM (Dynamic Systems Development Method), ASD (Adaptive Software Development), Kanban, LSD (Lean Software Development), Scrumban, RAD (Rapid Application Development), Crystal Methodology and AUP (Agile Unified Process) are the popular agile software development frameworks which are very common in practise. Some more practices are ATDD (Acceptance Test Driven Development), AT (Agile Testing), Backlogs (Product and Sprint), BDD (Behaviour Driven Development), BADM (Business analyst designer method), CI (Continuous Integration), Cross Functional Team, DDD(Domain Driven Design) , Information Radiators (task board, scrum board, visual management board, burn-down chart),IID (Iterative and Incremental Development), PP (Pair Programming), Planning Poker, Refactoring, Retrospective, Scrum events (sprint planning, daily scrum, sprint review and retrospective), Story Driven Modeling, TDD (Test Driven Development), Time-boxing, User story, User story mapping and Velocity tracking.

Agile methodology is based on technique called iterative enhancement where iteration represents a self contained and small scale Software Development Life Cycle (SDLC) as agile methodology is iteration based methodology along with assuming simplicity in all practices like the Spiral model [1]. Agile is an incremental and iterative based Software development approach. In Agile methodologies at the initial stage planning is done and throughout the project the changes are accepted along with a constant feedback is provided by the users for improvement of the project [2, 9]. The Agile Manifesto, properties of agile, study of various agile methodologies along with comparison and results has been discussed in this research paper.

A. The Agile Manifesto

Seventeen independent minded software practitioners in February 2001 write the Agile Manifesto which was based on interactions and individuals over tools and processes, working software over documentation which is comprehensive in nature, customer collaboration over contract negotiation , responding towards change over following a plan having twelve principles which are [4,5, 10, 28] :-

i. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
ii. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
iii. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
iv. Business people and developers must work together daily throughout the project.
v. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
vi. The most efficient and effective method of conveying information to and within a development team is face to face conversation.
vii. Working software is the primary measure of progress.
viii. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
ix. Continuous attention to technical excellence and good design enhances agility.
x. Simplicity the art of maximizing the amount of work not done is essential.
xi. The best architectures, requirements, and designs emerge from self-organizing teams.
xii. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

B. Properties of Agile Software Development Methodologies

Properties of Agile Software Development Methodologies or Agile Methodologies are given below:-

1) Deadline Emphasis: Deadline Emphasis is high in the Agile based methodologies.
2) Approach: The approach of software development is adaptive in nature in agile development.
3) Perspective to change: Here in Agile development the change adaptability is acceptable which gives the better result.
4) User requirement: The user requirements are very emergent with rapid change and interactive input which keeps the user busy in Agile development.
5) Primary objective: In Agile Software development methodologies the rapid value are the main objective.
6) Development life cycle: In Agile development the life cycle is based on Iterative along with the Evolutionary believing model delivery very fast.
7) Style of development: The style of development is adaptive in nature in Agile methodology.
8) Architecture: The feature of Agile is that it is designed for the current requirement based on the current environment.
9) Documentation: There is very low documentation in this development that has been replaced by the face to face communication, the tacit knowledge.
10) Software development process: The development process adopts the flexible approach after understanding the contextual needs results the faster development.
11) Fundamental Assumptions: Small teams are ready to develop the adaptive software by using the principles of continuous design along with improvement and testing which is based on feedback and change. These efforts results the high quality software.
12) Developers: Developers are with sound knowledge believes in collaboration. They are generally senior technical staff.
13) Team Size: Team size in Agile development is small along with the creative option depending upon project.
14) Management: Management in Agile development is people centric with leadership quality.
15) Goal: The goal in this development is very much clear for better result which may be explorative of adaptive in nature.
16) Change: This approach in lightweight development methodology is used in change at any stage.
17) Team organization: This is the beauty of Agile development that the self organizing team concept has been used.
18) Development direction: The main concept of any development methodology which is easily changeable in Agile development.
19) Additional abilities required from developers: Additional ability flowers the quality of a developer. Agile methodology requires interpersonal activity and basic knowledge of business from developer as extra ability for better performance.
20) Role Assignment: The self organizing perfection of Agile methodology encourages role interchange ability.
21) Project Cycle: In Agile development the project cycle depends upon product feature.
22) Desired Organizational Form or Structure: This feature is of Organic type in nature which means that structure would be flexible, participative, encouraging, cooperative and social active.

23) Technology: In technology agile development methodology favours Object Oriented technology.
24) Emphasis: The emphasis in agile methodology is people oriented for better performance.
25) Customer: This feature is like perfuming that the customer is dedicated, believes in collaboration with good knowledge in agile development methodology.
26) Clients: With hairline difference from customer feature the client is good knowledge, cooperative, representative and empowered.
27) Success Measurement: Business Value is directly proportional to the success measurement means that is the business value will be high than the success measurement will be high.
28) Project Size: In agile development the project size is small in nature which is beneficial for such methodologies.
29) Cycle: In project which is based on agile development methodology faces the numerous cycle in completing.
30) Domain: Here the domain is unpredictable along with if possible exploratory in nature.
31) Upfront Planning: In Agile development the Upfront planning is minimal which gives the better performance.
32) Return of Investment: The major issue, return of investment is early in the project which follows the agile development methodologies.
33) Quality Control: In Agile development the quality control is an art which is completed by understandable requirement, good design and solutions along with permanent testing.
34) Cost of restart: In such type of Agile methodology the cost of restart is low.
35) Planning and Control: In this methodology internalized plans and quality control are the factors for good results.
36) Refactoring: In Agile methodology the refactoring in inexpensive in nature giving a great advantage to this methodology.
37) Risk Impact: The risk impact is major in Agile development because of the unknown risk.
38) Testing: The major reason of quality is testing, because this is done in every iteration in project.
39) Suitable Project Scale: In Agile software development methodology the suitable scale of project in from low to medium size.
40) Organizational culture: In such type of methodology organization culture is leadership quality and collaborative culture in nature.
41) Short term schedule: In Agile development short term schedule are excellent and gives the good result.
42) Functionality: Functionality in Agile development is dynamic in nature.
43) Resource control: There in no control on resource in Agile software development methodologies.

II. AGILE SOFTWARE DEVELOPMENT METHODOLOGIES [12, 13, 14, 15, 16, 17]

A. Agile Modeling

AM (Agile Modeling) is practices based software development approach used for articulating a well organized project management procedure allowing for recurrent alterations. Agile is the theoretical outline for initiation many projects related to software engineering projects. Software development in short time boxes minimize the risks these short time boxes are called iterations which are incorporated with the time duration. Agile development describes various methodologies so it is not a methodology but an umbrella term. In Agile Manifesto the XP, Crystal, Scrum, DSDM and FDD methodologies has been included in the year 2001. This is shown in Fig-1.

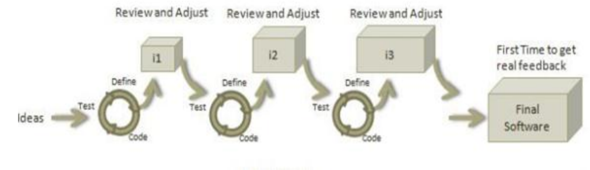

Fig 1: Agile Modeling [22]

*B. Extreme Programming**

Teamwork of Managers, customers and developers has been emphasizes by XP (Extreme Programming). Communication, simplicity, feedback, respect and courage parameters has been used to improve the software project.. The XP is responsiveness towards dynamic customer requirements and software quality improvement. Customer requirements adoption improves productivity and checkpoint introduction regarding new customer are advocated by the XP by frequent "release" in short development cycles. XP is a collection of concrete practices, simple and good general purpose method for software development. Difference between Agile and XP is discrete method. XP2 is the revised version of XP practices which includes work space. The work space is informative, whole team, peer programming, under pressure work stories, cycle which is trimester in nature, weekly cycle, incremental design, 10 minute development nature, continuous integration etc . Collaboration of total team as a unit in the presence of simple practices with feedback that is perfect enabling the team for visualization the progress and tuning the practices to their unique situation is in the working of XP [8, 11, 18]. This is shown in Fig-2.

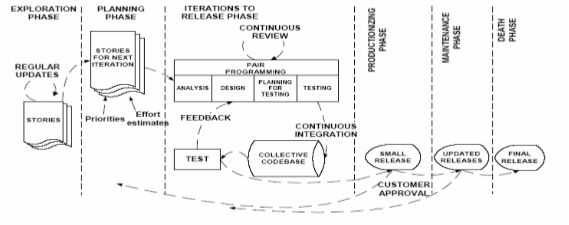

Fig 2: Life Cycle of the XP process [19]

C. Scrum Development

Scrum Development Methodology is combination of incremental and iterative software development framework of agile. Software projects which focus on a holistic flexible strategy of product development where to achieve common goal whole team efforts like a unit which is opposite from traditional and sequential approach are managed by Scrum. There are few questions from Scrum first is why efforts are so lengthy and tedious to do stuff? Second why the calculation of accuracy of efforts is not measured in project along with embracing creativity and uncertainty because of working nature of people? A structure around the learning process and enabling teams has been created to assess both what they have created and how they created it using the Scrum? Development Team, Scrum Master, Product Owner, Stakeholders and Managers are the core roles for producing the product [11]. This is shown in Fig-3.

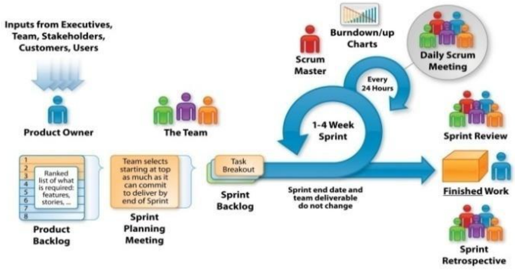
Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective

Fig 3: Scrum Process [3]

D. Feature Driven Development

FDD (Feature Driven Development) focuses on building and design phase of the development rather than the whole development in software development life cycle. In Build there is a phase known as feature list in which a comprehensive list of features has been identified by team and after grouping the feature into sets of feature and after that main feature sets systematically. The collected feature list is prioritized in plan by feature phase that means according to priority nature and development plan is established where order in which feature sets realization has been added. In design based and fourth build based on feature phase the team launches into design series and build both by feature iterations where breaking of them into feature teams and design, test and features build in time boxes of two week. Repetition of this step has been repeated till there is no more existence of features [6, 8, 18]. This is shown in Fig-4.

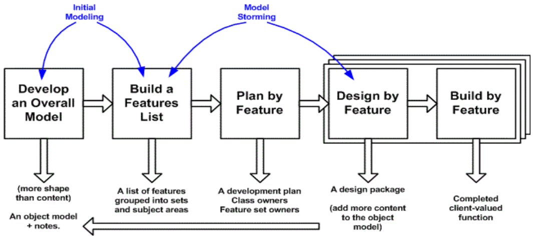
Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
An object model + notes.
A list of features grouped into sets and subject areas
A development plan Class owners Feature set owners
A design package
(add more content to the object model)
Completed client-valued function

Fig 4: Feature Driven Development Process [3]

E. Dynamic Software Development Method

DSDM (Dynamic Software Development Method) in agile project development is forward as well as backward looking framework. DSDM focuses on quick delivery of product along with a guiding methodology for control the process simultaneously. Technique which are used for requirements prioritizing are assigned on Should have, Must have, Could have, Want to have bases but will not have this time round which is known as Moscow rule. When a project is developed along with DSDM the feasibility and Business both study must be done in sequentially manner. Decision of the ground for the rest of the project is done by these two phases. The ending three phases are incremental and iterative in nature. Actually the development is done in the project during these phases [6,8, 18]. This is shown in Fig-5.

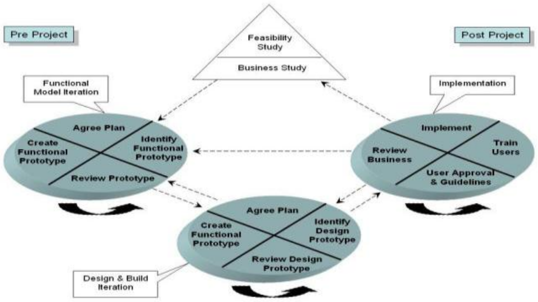

Fig 5: DSDM Process Diagram [3]

F. Adaptive Software Development

ASD (Adaptive Software Development) states that may be fuzzy requirements in e-business project beginning. The setting project mission handling and identifying requirements, objectives, project time box which is rooted on the feature set requirements, estimates, scope and resources availability is done by speculation. Overall project size and degree of uncertainty decides iteration length. After that iteration is being assigned time box. A statement which is objective in nature is written by team members for the each iteration. Features are assigned by the users and developers to the each iteration in the last. Teamwork needs of trustful and respectful collaboration. Team must collaborate on rapid decision making, requirements and problems. Level of real understanding can be improved by learning. The focus of this phase on groups provides feedback, formal technical reviews and post mortems [6]. This is shown in Fig-6.

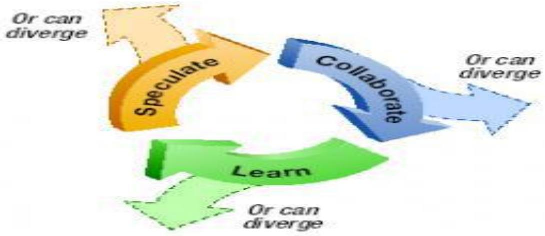

Fig 6: ASD Life Cycle [3]

G. Kanban

Kanban is a system to control the chain which is logistic not an inventory control system from a production point of view. Kanban was developed to improve the system and keeping up a production level high. Kanban became popular and effective tool in support of running a production system as a whole and it proved to be a way for promoting improvement which is excellent [7]. This is shown in Fig-7.

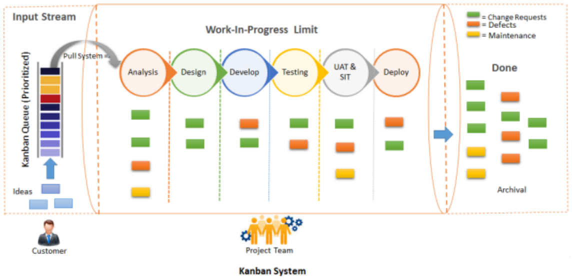

Fig 7: Kanban System [23]

H. Lean Software Development

LSD (Lean Software Development) considers the resources expenditure for any goal as it is production practice .Value is defined so that customer willing to pay. LSD is centred in nature for preserving value along with less work. The seven principles of LSD are elimination of waste, building of quality product, creating knowledge, fast deliver the software, people respect and optimizing the whole and dissimilar commitment [7, 18]. This is shown in Fig-8.

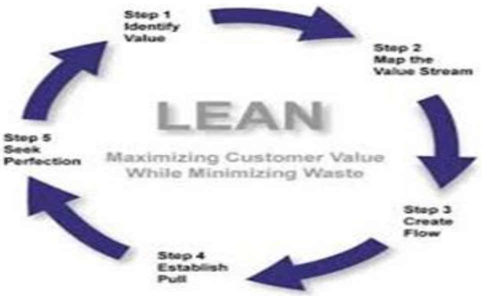

Fig 8: Lean Software Development [24]

I. Scrumban

Scrumban is the mixture of set of elements from Scrum and Kanban, a hybrid methodology of agile designed to play with customer requirements of dynamically changing in nature and frequent coding problems. Scrumban does not contain sprints, possessing the practices which are best in nature of the Scrum development like daily stand-up meetings, user-stories and self-organized team aspects. Like a Scrum task board is not enough to the changes reflection where sprints were replaced with a style of Kanban pull driven coordination mechanism with work in progress limitations. Improvement in workflow as the software teams improve their processes is guaranteed by the pull mechanism. [9].This is shown in Fig-9.

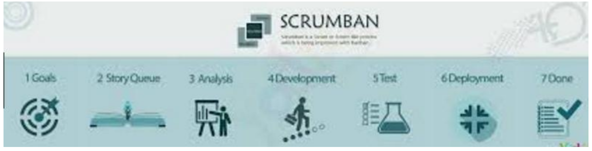

Fig 9: Lean Software Development [27]

J. Rapid Application Development

The quick result to give excellent development processes with the assistance of other development approaches to take the maximum advantage from the development software is the aim of RAD (Rapid Application Development). It is specially designed to augment the workability of the software development procedure as a whole along with the active user participation. This is shown in Fig-10.

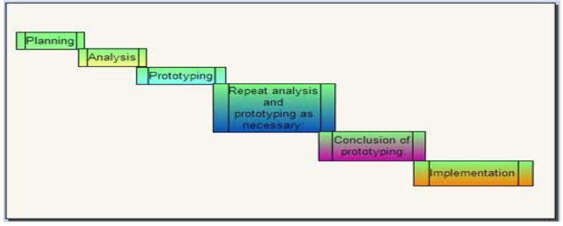
Planning
Analysis
Prototyping
Repeat analysis and prototyping as necessary
Conclusion of prototyping
Implementation

Fig 10:–RAD Model [20]

K. Crystal Methodology

Crystal Methods was developed by Alistair Cockburn. Methods of Crystal are colour coded for signification the risk to life of human like risky projects will use Crystal Sapphire while without risky use Crystal Clear. There are six aspects which are primary in nature include interaction, people, communication, community, talents and skills on Crystal focus. Process is secondary consideration in Crystal method. The reflective improvement, frequent delivery, easy access to expert users and osmotic communication are the properties in Crystal which indicates the higher possibility of success. Due to human powered or people centric focus the methods are flexible in nature means non rigid nature. The different colours are used in the Crystal family of methodologies for denoting the "weight" of which methodology to use. Methodology such as Crystal Clear, Crystal Orange or Crystal Yellow can be used for small project or for critical project may be dangerous for human life the methods Crystal Sapphire or Crystal Diamond may be used are Crystal Clear, Crystal Yellow, Crystal Orange, Crystal Orange Web, Crystal Red, Crystal Maroon, Crystal Diamond and Crystal Sapphire are the division of colours in Crystal family. This is shown in Fig-11.

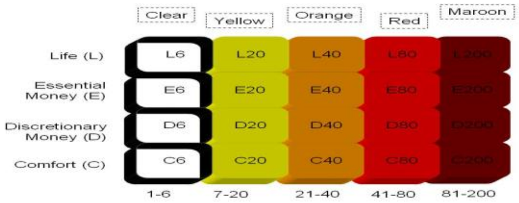

Fig 11: Crystal Methodology [21]

L. Agile Unified Process

AUP (Agile Unified Process) is a version of the Rational Unified Process (RUP) which is simplified in nature focusing on real time and web based development. AUP is a combination of RUP (Rational Unified Process) and AM (Agile Method). DAD (Disciplined Agile Delivery) pressed AUP in 2012. TDD (Test Driven Development), AM, agile change management and database refactoring agile techniques are being used by AUP to refine the productivity. In AUP iteration consists of seven disciplines or work areas that are to be performed during iteration includes model, implementation, test, deployment, configuration management, project management and environment. Sets of artefacts means work product, roles which includes responsibilities which are taken by members of development team and activities like units of work on the artefacts are defined by Agile Unified Process in each discipline.AUP consists of four phases which includes Inception, Elaboration, Construction and Transition. A refinement to the Rational Unified Process is that iterations which are of two types firstly iteration which belongs to development release resulting in a the quality assurance deployment along with demo area and secondly iteration belongs to production release resulting in the production area deployment are characterised by the Agile Unified Process [15, 29]. This is shown in Fig-12.

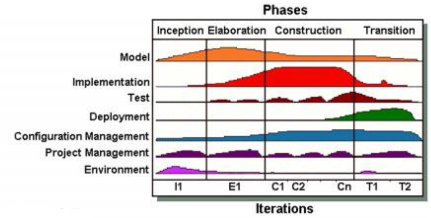

Fig 12: Agile Unified Process Phases [30]

III. COMPARISON BETWEEN AGILE METHODOLOGIES

There are twelve agile methodologies which are represented in table no 1 shows that as per requirement analysis the AM (Agile Modeling), XP (Extreme programming), Scrum Development, FDD (Feature Driven Development), DSDM (Dynamic Systems Development Method), ASD (Adaptive Software Development), Kanban, LCD (Lean Software Development), Scrumban, RAD (Rapid Application Development), Crystal Methodology and AUP (Agile Unified Process) gives positive response on change of requirement. In the defining of the requirement at the stating level the response is poor except FDD, DSDM and LCD agile methodologies. The response of the requirement based complexity detection of the system is good in the AM, XP, FDD, DSDM, ASD and AUP. In simplicity the agile methodologies which have been defined are very much along with the simplicity. In methodologies like AM, Scrum, DSDM, Kanban and Scrumban the overlapping phase are incorporated. The methodologies are very much familiar in the changes which are incorporated. As the status based on development team the less experience on similar project, less knowledge of domain, less knowledge of technology, less experience to the tools to be used in project of the development team are not entertained in the agile methodologies which are to describe in the table no 1. Training availability to the development team on requirement is kept in hands in agile methodologies. Variation has negative response in these methodologies. Understanding ability is not simple in the projects entertaining agile methodologies. Regarding user's participation, fully involvement and feedback from user in the project which are handled with agile methodologies the full participation gives best result. Customer satisfaction and customer priority is at very high degree in most of the agile methodologies. On the basis of the project type and associated risk the agile methodologies which are associated in table no 1 the funding, requirement reliability, project tightness has positive response in the projects incorporated. The methodologies of agile which are describe changing speed according to requirement, project predictability, risk identification of the project, practically implementation of the completed project, usability of the project and industry approach towards the methodologies have the good response in agile methodologies which are described. Elasticity of the methodologies towards the project is high and moderate which gives the good response. Failure of the project using the agile methodologies is about to be nil which are describe in the table no 1. Involvement of the risk is low in AM, XP, Scrum, ASD, Kanban, Scrumban, Crystal and AUP methodologies. Regarding integrity and security nature the methodologies like AM, XP, Crystal and AUP are robust rest are not robust. ASD and LSD methodologies are project dependent nature and DSDM is team dependent in nature regarding time frame parameter. AUP methodology weight is heavier than that of other methodologies which have been described. Described agile methodologies have not very much positive towards the complex system.
The explanation of the comparison is shown in table no. 1.

IV. RESULT AND DISCUSSION

Researchers studies and compare various methodologies of agile family on the basis of the fifty parameters and results that no one methodology is perfect for all software product development. Every methodology has merits and demerits. The project nature, developer behavior and understanding are main factors which responsible for selection of the methodology for development.

V. CONCLUSION AND FUTURE SCOPE

Agile methodologies are giving their best by focusing on customer satisfaction, people relations, producing best product and cost benefit analysis including increase in the flexibility. Agile approach is to break down the large tasks into smaller ones which help them to get completed within the given time frame. Agile software development methodologies are very critical and depend on the host of factors including culture of organization, flexibility offered by the customer, customer awareness and knowledgeable demand of the customer. This paper is intended to deal and come out with a fair comparison of the agile software development models and make the users aware of the characteristics of each, in order to enabling them to match the same with their experience. One thing is clear is that there is no one size fits to all solution. Understanding of the differences between various software development methodologies improves the decision power for selecting of the most suitable methodology in a suitable manner to the developer.

In the future a common methodology is the demand of development era with minimum efforts and maximum output. As the new emerging technologies era is growing and customer is being aware about technological era so that the pressure on the developer is being increasing day by day. The one factor is the involvement of the customer during the development process. So the new development methodology is solution of this problem.

REFERENCES

[1] M. Al-Zewairi, M. Biltawi, W. Etaiwi and A. Shaout, “Agile Software Development Methodologies: Survey of Surveys”, Journal of Computer and Communications,Vol. 5, pp. 74-97, 2017.

[2] A. Sharma and M. Bali, “Comparative Study on Software Development Methods: Agile vs Scrum”, International Journal of Emerging Research in Management &Technology (IJERMT), Vol.6, Issue 6, pp. 165-168, 2017.

[3] Priyanka and P. Kantha, “A Comprehensive Study of Traditional and AGILE Software Development Methodologies”, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Vol. 6, Issue 11, pp. 128-138, 2016.

[4] http://agilemanifesto.org/ [Accessed Dec. 14, 2018].

[5] https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/ [Accessed Dec. 14, 2018].

[6] Shelly, “Comparative Analysis of Different Agile Methodologies” , International Journal of Computer Science and Information Technology Research, Vol. 3, Issue 1, pp. 199-203, 2015.

[7] R. P. Pawar, “A Comparative study of Agile Software Development Methodology and traditional waterfall model”, Innovation in engineering science and technology, pp. 1-8, 2015.

[8] Manvender Singh Rathore and Deepa V. Jose, “Oriental Journal of Computer Science & Technology”, (2017), Vol. 10 (2), pp. 352-358.

[9] H. Saeeda, M. Ahmed, H. Khalid, A. Sameer and F. Arif, “ Systematic Literature Review of Agile Scalability for Large Scale Projects”, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 6, Issue 9, pp. 63-75, 2015..

[10] T. Sharma, M. Mann and R. Thakur, “ Comparison Between Agile Methodology and Heavyweight Methodology: A Survey”, International Journal of Technical Research and Applications, Vol. 3, Issue 5, pp. 275-284, 2015.

[11] M. Almseidin, K. Alrfou, N. Alnidami and A. Tarawneh, “A Comparative Study of Agile Methods: XP versus SCRUM”, International Journal of Computer Science and Software Engineering (IJCSSE), Vol. 4, Issue 5, pp. 126-129, 2015.

[12] M. L. DESPA, “Comparative study on software development methodologies”, Database Systems Journal, Vol. 5, pp. 37-56, 2014.

[13] N. Rashid, “ Applying Agile Methodologies on Large Software Projects”, International Journal of Recent Research in Mathematics Computer Science and Information Technology, Vol. 2, Issue 1, pp. 273-278, 2015.

[14] A. Kaushik, “ A Literature Review on Agile Software Development”, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 5, Issue 9, pp. 337-339, 2016.

[15] H. K. Flora and S. V. Chande, “A Systematic Study on Agile Software Development Methodologies and Practices”, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5, Issue 3 , pp. 3626-3637, 2014.

[16] K. Jammalamadaka and V R. Krishna, “AGILE SOFTWARE DEVELOPMENT AND CHALLENGES”, International Journal of Research in Engineering and Technology (IJRET), Vol. 02, Issue 8, pp. 125-129, 2013.

[17] A B M Moniruzzaman & S. A. Hossain, “Comparative Study on Agile Software Development Methodologies”, Global Journal of Computer Science and Technology Software & Data Engineering”, Vol.13 issue 7, Ver. 1.0, 2013 .

[18] U. Kumari and A. Upadhyaya, “Comparative Study of Agile Methods and Their Comparison with Heavyweight Methods in Indian Organizations”, International Journal of Recent Research and Review,Vol.6 , 2013.

[19] M.A. Awad, “A Comparison between Agile and Traditional Software Development Methodologies,” 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.464.6090&rep=rep1&type=pdf [Accessed Nov. 14, 2018]

[20] D. Thakur, “Rapid Application Development (RAD) Model and its Advantages and Disadvantages of RAD Model”
Available: http://ecomputernotes.com/software-engineering/rapidapplication- development [Accessed Nov. 14, 2018].

[21] A. I. Khan, R. J. Qurashi and U. A Khan, “A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies”, International Journal of Computer Science Issues (IJCSI), Vol. 8, Issue 4, pp 441- 450, 2011.

[22] https://www.tutorialspoint.com/agile/agile_tutorial.pdf

[23] Tutorialpoint, “Agile Software Development methods,” 2014. [Online]. Available: https://www.tutorialspoint.com/agile/agile_tutorial.pdf [Accessed Jan. 14, 2018].

[24] L.K. Shinde, Y.S. Tangde and R.P. Kulkarni, “Traditional Vs. Modern Software Engineering – An Overview of Similarities and Differences”, Advances in Computational Research, Vol. 7, Issue 1, pp 187-190, 2015.

[25] Murat Yilma and Rory V. O’Connor, “A SCRUMBAN Integrated Gamification Approach to Guide Software Process Improvement: A Turkish Case Study”, Technical Gazette, Vol. 23 Issue 1, pp. 237-245, 2016.

[26] Łukasz D. Sienkiewicz, “SCRUMBAN – The KANBAN as an Addition to Scrum Software Development Method in A Network Organization”, BUSINESS INFORMATICS, Vol. 2, Issue 24, 2012.

[27] M. Stoica, M. Mircea and B. G. Micu, "Software Development: Agile vs. Traditional", Informatica Economica, Vol. 17, Issue 4, pp. 64-76, 2013.
[28] http://agilemanifesto.org/principles.html
[29] C. Edeki, "Agile Unified Process", International Journal of Computer Science and Mobile Applications (IJCSMA), Vol.1, Issue 3, pp. 13-17, 2013.
[30] Lisana, "Review on The Effectiveness of Agile Unified Process In Software Development with Vague System Requirements", ARPN Journal of Engineering and Applied Sciences, Vol. 9 Issue 10, pp.1763-1768, 2014.

Authors Profile

Rakesh Kumar is a Ph.D. research scholar in Computer Science and Engineering department at Rabindranath Tagore (Formerly known as AISECT) University, Bhopal. His area of research is Software Methods and Development Techniques for the Internet of Things (IoT). He has worked on a research project titled 'A Novel Disaster Detection and Environmental Monitoring System Developed for Simhastha 2016 at Ujjain' on IoT based sponsored by AISECT University. He is an author of two journal papers. He has completed his Master of Philosophy Computer Science from Chaudhary Devi Lal University, Sirsa, Master of Technology from Karnataka State Open University, Mysore, Master of Computer Application from Indira Gandhi National Open University, New Delhi and Diploma in Advance Computing from Advance Computing Training School, Pune. His area interests include IoT, IoE, Software Engineering, Operating System, Database Technologies, Data Structures, Data Mining, Data Warehousing, Artificial Intelligence and Computer Architecture. He has knowledge of programming languages and technologies like C, C++, C#, Java, .NET, MFC, SDK and Python. His experience is about 10 years of experience of teaching in Computer Science at college and University level. He has also guided final year computer science students of Bachelor and Master Degree in their final projects. He is crazy about new technologies.

Priti Maheshwary is an Associate Professor in Computer Science and Engineering department and Director, Centre of Excellence for IoT & Advanced Computing (CIoTAC) at Rabindranath Tagore University (Formerly known as AISECT University), Bhopal. She had her Doctorate from MANIT, Bhopal in Remote Sensing Image Retrieval. She has completed research project on Climate Change detection and monitoring funded by SAC and Environment Monitoring using Sensor devices funded by AISECT University. Ongoing Project is on Crop Monitoring sponsored by SAC. She is the author of more than 20 publications in different journals of repute out of which two journal papers and two conference papers on IoT. She is the author of book chapter on Software Copyright. Her interests include Internet of Things, Cyber Physical Systems, Mobile Networks, WSN, Adhoc Networks, Data mining, and Image Processing. Her work experience includes 20 years in teaching computer science and engineering. She is guiding PhD in the field of image processing, IoT and Networks.

Timothy Malche is a research scholar in Computer Science and Engineering department at Rabindranath Tagore (Formerly known as AISECT) University, Bhopal. His area of research is Internet of Things (IoT) for Smart City. He has worked on a research project titled 'A Novel Disaster Detection and Environmental Monitoring System Developed for Simhastha 2016 at Ujjain' on IoT sponsored by AISECT University. He is the author of two journal papers and two conference papers on IoT. He has completed Master of Computer Application from JRN Rajasthan Vidyapeeth (Deemed) University, Udaipur. His interests include application programming, website development and embedded system programming. He has mastered programming languages and technologies like C/C++, C#, Java, Python, HTML/CSS, Javascript, ASP.NET, PHP. His work experience include 10 years of experience in teaching computer science and website and software development. He has also guided final year computer science students in their final software projects.

Table 1: Comparison of Different Methodologies

S. No.	**Models → Parameters ↓**	**Agile Modeling (AM)**	**eXtreme Programming (XP)**	**Scrum Development**	**Feature Driven Development (FDD)**	**Dynamic Software Development Method (DSDM)**	**Adaptive Software Development (ASD)**	**Kanban**	**Lean Software Development (LSD)**	**Scrumban**	**Rapid Application Development (RAD)**	**Crystal Methodology**	**AUP**
1	Do we change requirements quit often?	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes
2	Can we define requirements at the starting of iteration?	No	No	No	Yes	Yes	No	No	Yes	No	Yes	No	No
3	Requirements are indicating a complex system to be built.	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	No	No	Yes
4	Clear Requirement Specifications	Change incrementally	Initial Level /Defined immediately	At medium level / Frequently Changed	No	Yes	No	Frequently Changed	Yes	Frequently Changed	Initial Level /Timebox Released	At medium level / Frequently Changed	Change Frequently
5	Precondition	No	No	Clean idea of Reuse component	No	Yes	No	Reuse Component	No	Reuse Component	Clean idea of Reuse component	Clean idea of Reuse component	No
6	Simplicity	Simple	Simple	Simple	Intermediate	No	No	Simple	Simple	Simple	Very Simple		Simple
7	Overlapping Phase	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	No	No
8	Change Incorporated	Easy	Easy	Difficult	Easy	Easy	Easy	Yes	Yes	Easy	Easy	Easy	Easy
9	Development Team Less experience on similar projects	No	No	No	No	Yes	No	No	Yes	No	No	No	Yes
10	Development Team Less domain knowledge (new to the technology)	No	No	No	No	Yes	No	No	Yes	No	No	No	Yes
11	Development Team Less experience on tools to be used	No	No	No	No	No	No	No	Yes	No	No	No	Yes
12	Availability of training to Development Team if required	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
13	Any variation done	No	No	No	No	No	Yes	No	No	No	No	No	No
14	Understand ability	Much Complex	Intermediate	Intermediate	Complex	Intermediate	Complex	Intermediate	Intermediate	Intermediate	Intermediate	Intermediate	Simple

15	Expertise Required	High	High	High	High	Moderate	Moderate	High	No	High	Medium	High	Moderate
16	User participation in all phases	Yes	Yes	Yes	Intermediate	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
17	Limited user participation	No	No	No	No	No	No	Yes	Yes	No	No	No	Yes
18	User have no previous experience of participation in similar projects	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	Yes	No
19	Users are experts of problem domain	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	No
20	Feedback from user	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
21	Loom	Highly customer satisfaction and incremental development	Customer satisfaction and incremental development	Highly customer satisfaction and incremental development	Customer satisfaction and incremental development	Customer satisfaction and incremental development	Customer satisfaction and incremental development	Customer satisfaction and incremental development	Customer satisfaction and incremental development	Customer satisfaction and incremental development	Use readymade component	Highly customer satisfaction and incremental development	Customer satisfaction and incremental development
22	Customer priority	High	Intermediate	High	High	High	High	High	High	High	Nill	High	High
23	User Involvement	Client onsite and considered as a team member, Active/proactive	High	High	Moderate Involvement through Reports	High Involvement through frequent Release	High Involvement through frequent Release	High	Low	High	High	High	Moderate
24	Project is the enhancement of the existing system	Yes	Yes	No	No	No	No	No	No	No	Yes	No	Yes
25	Funding is suitable for the project	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
26	High reliability requirements	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes
27	Tight project schedule	Yes	No	Yes	Yes	No	Yes	No	No	No	Yes	Yes	Yes
28	Use of reusable components	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
29	Are resources (time, money, people etc) scare	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes
30	Speed to change	High	High	High	High	Low	Medium	High	Medium	High	No	Medium	High
31	Predictability	High	High	High	High	Medium	Medium	High	Low	High	Low	High	High
32	Risk identification	Yes	Yes	Yes	Yes	Yes	Moderate	High	Low	High	No	Yes	Yes

33	Practically implementation	High	High	Medium	High	Medium	High	Medium	Medium	Medium	No	Medium	High
34	Usability	High	High	High	High	Medium	High	High	Medium	High	Medium	Medium	High
35	Industry approach	High	Medium	High	High	Medium	High	High	Medium	High	Medium	Medium	High
36	Cost	Much Expensive	High	High	High	Medium	High	High	Medium	High	Very High	Low	Medium
37	Resource organization	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
38	Elasticity	Very High	Medium	Medium	Medium	Medium	High	High	High	High	Yes	Medium	Medium
39	Gurentee of Success	High	High	Medium	High	Medium	Medium	Medium	Medium	High	High	High	Medium
40	Risk Involvement	Low	Low	Low	High	Medium	Low	Low	High	Low	Very Low	Low	Low
41	Flexibility	Flexible	Flexible	Flexible	Flexible	Less Flexible	Flexible	Flexible	Flexible	Highly flexible	Highly Flexible	Flexible	Flexible
42	Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance	Easily Maintenance
43	Integrity and Security	Robust	Robust	Weak	Weak	Weak	Weak	Weak	Weak	Robust	Vital	Robust	Robust
44	Time Frame	Short	Short	Short	Short	All time independent Team	Depends on Project	Short	Depends on Project	Depending on project	Short	Short	Short
45	Weight of Model	Light Weight	Light Weight	Light Weight	Light Weight	Light Weight	Light Weight	Light Weight	Light Weight	Light weight	Light Weight	Light Weight	Medium Weight
46	Complex System	Bad	Bad	Bad	Good	Moderate	Good	Bad	Moderate	Good	Bad	Bad	Moderate
47	Reliable	Moderate	Good	Bad	Good	Moderate	Moderate	Bad	Moderate	Moderate	Good	Good	Good
48	Schedule Visibility	Good	Good	Good	Good	Good	Good	Good	Moderate	Good	Good	Good	Good
49	Cost Control	NO	No	No	Yes	Yes	Yes	No	Yes	No	No	No	Yes
50	Failure normally due to	Code	Architecture and design	Architecture and design	Architecture and design	Functional Model	Customer	Roles of Team Members	Priject Manager and Team	Role of team members	Architecture and design	Architecture and design	Code