Why we need agile?

[1]Arsheen Neda Siddiqui [2]Aamir Hamid Lone [3]Imad Shafi Kar [4] Mohd Aamir

[1]M.tech Student,Alfalah University Harayana India,[2] M.tech Student,Alfalah University Harayana India,[3] M.tech Student,Alfalah University Harayana India.4 M.tech Student,Alfalah University Harayana India.

Abstract

With the growing market complex software application is being used due to competition between various organizations. private sector want good quality software product in minimal cost and time. This paper aims to discuss the agile philosophy and methodologies that leads to answer major concern of private organization such as minimal testing speedy development good quality software project.

Index Terms– Agile philosphy, Agile Methodology, Crystal Method, Extreme Programming, Feature Driven Development, Scrum, Test Driven Development.

1. Introduction

As we know market is growing rapidly and agility is magnet that attract various organization worked as private sector .due to increase in agility ultimately decrease the development time for new processes but keeping in mind the quality of software. Software flexibility should not be underestimated all this leads to decrease time for solving client demands. Agility is no longer a necessity but termed as the condition to remain in the market to cope with competitive aspect of the market .agility minimize the time of development hence client demands fulfilled without extra expenses. Agility can be achieved when the stakeholder clearly understand the working ways of organization .software application now a days growing with convolution because of increasing demands pressure had reached to new height for quality assurance of developed software. So keeping in mind the growing and competitive nature of the environment efficient methodology and techniques should used. Agile Methodologies are a group of software development methods that are based on iterative and incremental development. The four major characteristics that are fundamental to all agile methodologies are: adaptive planning, iterative & evolutionary development, rapid and flexible response to change and promote communication [1,2].

1.1Software development life cycle

Sdlc is a process these process are followed during development of any software project with in organization . Sdlc has detailed information how planning development implementation of software project has been done simply this is known as software development process. Sdlc involves list of stages performed to obtain the desired target. Here are the list of stages typically involves in traditional approaches or we can say stages that are in typical sdlc. Traditional approaches consists of models like waterfall model, v model,increanmental model etc.

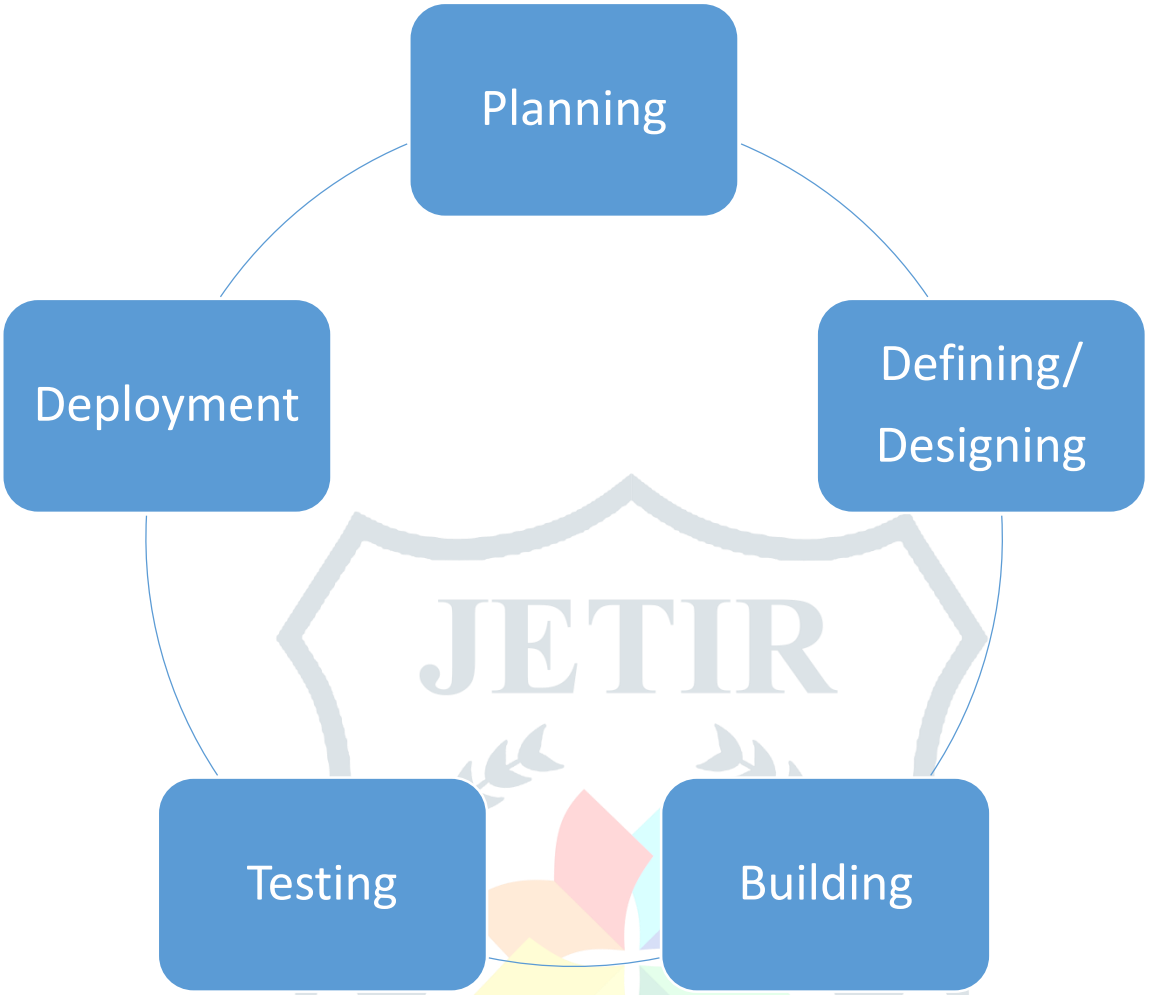


Fig: software development life cycle

1.2 Agile philosophy

Agile is simply a way of thinking how a development of any software project carried out to get good quality of product to meet the need of customers. With the use of 12 principles this philosophy justified and by [3]applying 4 core values and 12 principles. Scrum, XP etc. can be considered to frameworks under the Philosophy called "Agile"[4]

2. principles

P1: client is everything will increase his satisfaction by speedy delivery of computer code.

P2: Welcome and inclusion these days ever-changing necessities.

P3: Continuous delivery of operating computer code.

P4: improvement of technical excellence and smart style by keeping continuous attention.

P5: Simplicity is important.

P6: Progress measuring through operating computer code.

P7: Face to face communication is that the best communication.

P8: Develop comes in healthy setting with trustworthy.

P9: Have self organizing groups.

P10: Self judgment at regular intervals to become simpler.

P11: property development, ability to take care of a continuing.

P12: Co-operation between developers and business persons.

2.1 Implementation approches

Software quality has forever been a hot topic for software system developers and customers still. This can be achieved providing our software system project management framework is up to the mark so for improving the recent frameworks, new concepts came up day by day. scrummage is one among these new concepts.

From 2001, with the discharge of agile declaration, scrummage additionally created its position within the literature of the software system trade. scrummage focuses on project management in things where it's tough to arrange ahead. software system is developed by a self-organizing team in increments(called "sprints"), beginning with designing and ending with a review. options to be enforced within the system square measure registered during a backlog. Then, the merchandise owner decides that backlog things ought to be developed within the following sprint. Team members coordinatetheir add a daily stand-up meeting. One team member, the scrummage master, is accountable of finding issues that stop the team from working effectively [2].

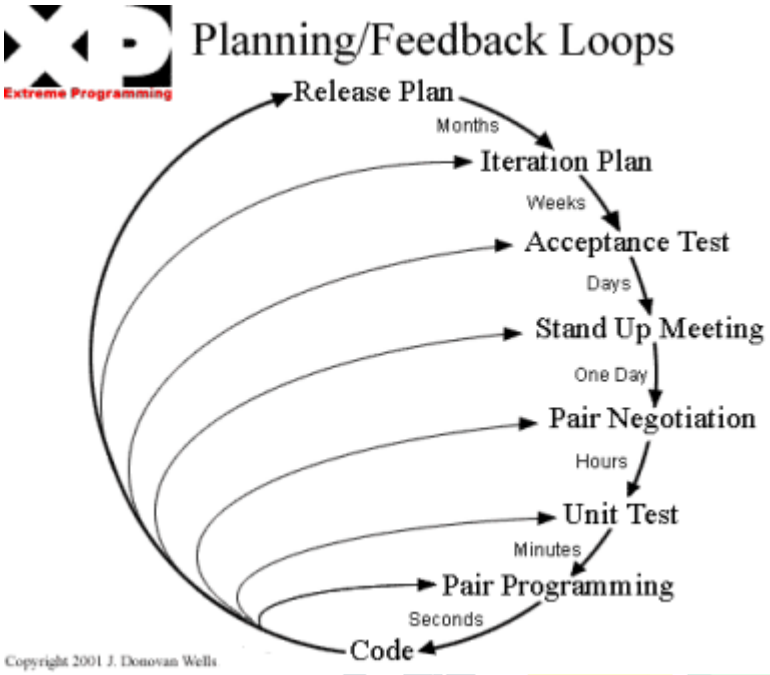
For example, supported expertise, one report claims that scrummage practices akin to daily scrummage, scrumof scrums, sprint designing and retrospective conferences have interaction distributed team members in collaboration, facilitate mental image of hidden issues, develop trust and increase shared aims [1]. Daily scrum conferences bring transparency and encourage informal communication among distributed stakeholders; sprints give frequent offsite work observation opportunities; sprint designing meetings give shared understanding of common goals and improve task awareness, and; sprint"demos" bring transparency to stakeholders and stop issues early[4]. associate open sprint designing among team members give a full summary and information to each team individual. This review tries to guage and gift the analysis findings on scrummage up to gift day. This systematic review provides an outline of fields researched, their results and future implications. we tend to hope this

review shall be terribly useful for the researchers WHO square measure enthusiast for locating one thing new inthe field of scrummage and software system engineering.

Extraordinary Programming (XP)

Extraordinary Programming (XP) was presented in 1996 by Kent Beck when he was taking a shot at a Chrysler finance venture. Despite the fact that the term was new around then, the strategies utilized, for example, match programming, unit testing, and client acknowledgment tests have existed previously. What XP did any other way was to take these accepted procedures to "outrageous levels". It was made to consider changes that are going on around then in programming advancement and innovation, to be specific, question situated programming and the Internet. XP places significance in co-area of clients and engineers, and match programming, which additionally agrees with the estimations of collaboration in

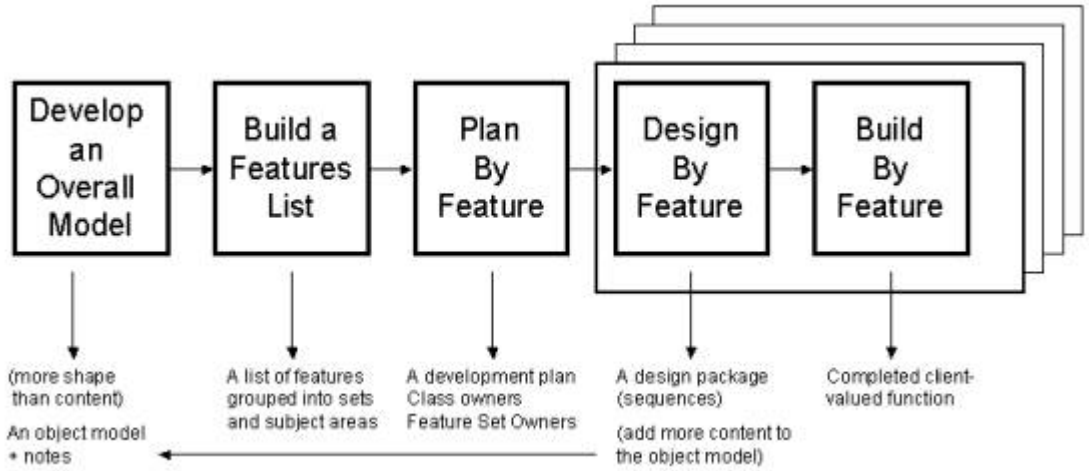
Agile. It likewise has short improvement cycles, concentrating on the requirements of today instead of those of things to come, in some cases called the "You aren't going to require it!" or YAGNI approach. Kent Beck is one of the signatories of the Agile Manifesto Extreme programming's goal is to deliver software that is needed when it is needed. XP is best used by a small team of programmers, between 2 and 12, working on projects with dynamic requirements or high risk. The XP development team works together with managers and customers. An example project that used XP was Encyclopedia Britannica available on CD and web subscription.[4]



Feature-Driven Development (FDD)

Feature-Driven Development (FDD) used to be brought in 1997 by Jeff De Luca when he was working in a software development challenge for a massive Singapore bank. FDD used to be also built round software engineering satisfactory practices such as domain object modeling, creating by using feature and code ownership. The blending of these practices that resulted in a cohesive entire is the nice

characteristic of FDD. It consists of five basic activities, namely, the improvement of an normal model, the constructing of a characteristic list, the planning via feature, the designing by feature, and the constructing by using feature. Every task will have its personal special model, which will result in a characteristic list. The remaining three things to do are quick iterative processes, with a function now not taking longer than two weeks to build. If it will take more than two weeks, then it will have to be damaged down into smaller features.

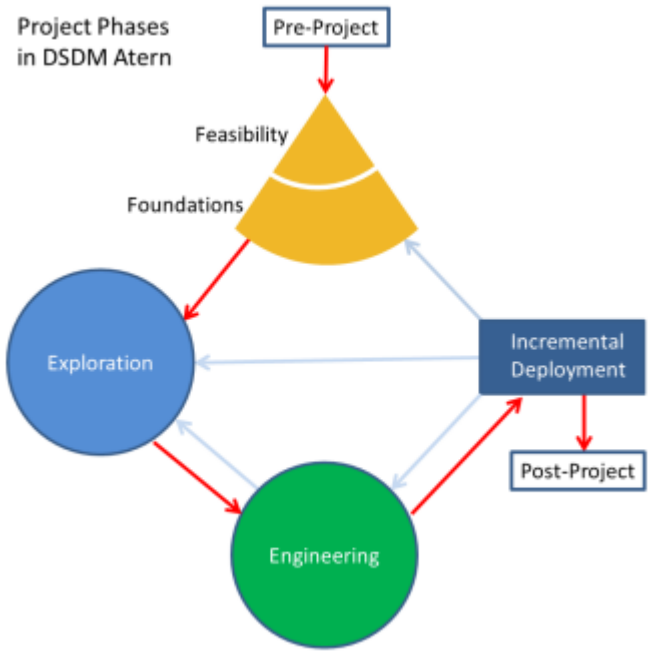


FDD's main purpose is to deliver tangible, working software over and over in a timely manner. The advantage of using FDD is that it is scalable even to giant groups due to the concept of 'just ample diagram initially' (JEDI). It is a extremely good solution to hold control over agile, incremental and inherently complex projects due to the fact of its feature-centric process. It is used in organization projects as properly as net tasks such as the Mousebraker.com on line gaming site.

FDD is a client-centric, architecture-centric, and pragmatic software process [5].

Dynamic Systems Development Method (DSDM)

Dynamic Systems Development Method (DSDM) used to be posted in 1995 via the DSDM Consortium, an association fashioned via providers and specialists in software program engineering to furnish a shape for Rapid Application Development methods added on with the aid of object oriented programming. The Consortium at the same time developed and promoted a tool- and technique-independent development framework from excellent exercise experiences of human beings working in large agencies such as British Airways, American Express, Oracle and Logica. It has now advanced into a undertaking shipping framework that is entirely compatible with ISO 9000 and PRINCE2. In 2007, it used to be rebranded Atern after the chicken Arctic Tern. However, for the reason that 2014, it has reverted lower back to its authentic title as DSDM Agile Project Framework. Also, in 2016, the DSDM Consortium rebranded as the Agile Business Consortium.



DSDM consists of eight standards that will direct the group and create a mind-set to supply on time and within budget. Principles consist of focusing on the enterprise need, handing over on time through time boxing work, and emphasizing collaboration with cease users, crew members, business representatives and different stakeholders. As a framework and not simply a software program improvement method, it can also be used in non-IT projects. It is often used in government projects the place it is paired with task management standards, such as PRINCE2.

Crystal Methods

Crystal Methods are a family of software program improvement methodologies developed by using Alistair Cockburn from his find out about and interviews of teams. The strategies are color-coded to signify the hazard to human life. For example, initiatives that may involve threat to human existence will use Crystal Sapphire while projects that do no longer have such risks will use Crystal Clear.

Crystal focuses on six fundamental aspects: people, interaction, community, communication, skills, and talents. Process is considered secondary. There are also seven common homes in Crystal that indicate higher possibility of success and they include generic delivery, reflective improvement, osmotic communication, and effortless access to specialist users. The techniques are very flexible and keep away from rigid tactics because of its human-powered or people-centric focus. Alistair Cockburn is also one of the unique signatories of the Agile Manifesto.

Crystal Methods think about human beings as the most important, so tactics need to be modeled to meet the requirements of the team. It is adaptive, barring a set of prescribed tools and techniques. It is additionally lightweight, besides too lots documentation, management or reporting. The weight of the methodology is determined via the project environment and team size. For example, Crystal Clear is for momentary tasks by way of a team of 6 builders working out of a single workspace

3. We need agile because of following questions which can be answered by agile framework

1. How agile philosophy leads to better software development?
2. How cost can be maintained and get better quality of software?
3. How agile philosophy leads to minimal testing?

4. Conclusion

Sdlc is predefined mould where experts are bounded to go with the predefined software development methodologies. By using traditional approaches experts are bounded to close environment. Agile gives flexibility during the development phase we have analyzed that agile leads to minimal testing hence reduce the effort time and cost.

REFERENCES

[1] Andrew Begel, Nachiappan Nagappan, "Usage and Perceptions of

Agile Software Development in an Industrial Context:An Exploratory

Study", First International symposium on empirical software engineering and measurement, pp. 255-264, 2007.

[2] Peter Maher, "Weaving Agile Software Development Techniques into a Traditional Computer Science Curriculum", Proc. of 6th IEEE International Conference on Information Technology: New Generation, pp. 1687-1688, 2009.

[3]Agile – A Philosophy or A Framework? by knowledge hut editor.

[4] Agile Manifesto and Agile Principles, http://agilemanifesto.org/

[5] Jeffrey A. Livermore, "Factors that impact implementing an Agile Software Development Methodology", pp. 82-85, IEEE 2007.

[6] A. Ahmed, S. Ahmad, Dr. N. Ehsan, E. Mirza, S.Z. Sarwar, "Agile Software Development:Impact on Productivity and Quality" , pp. 287-290, IEEE 2010.

[7] Software Testing Techniques: A Literature Review by Muhammad Abid Jamil 1,2, Muhammad Arif 2, Normi Sham Awang Abubakar 1, Akhlaq Ahmad 1, 3 1 KICT, International Islamic University, Kuala Lumpur, Malaysia 2 College of Comp.

[8]www.extremeprogramming.org/introduction.html

www.educba.com/all-about extreme-programming/

[9] Kristin Fergis, "The Impact of an Agile Methodology on Software Development Costs", Project Report, EAS 499 Senior Capstone Project, April, 2012.