

# Agile Software Development Methodology – Framework For Efficient & Quick Delivery Of Software Product

Sharma Nimisha H.<sup>1</sup>, Pinal Patel<sup>2</sup>

<sup>1,2</sup> Government Engineering College Gandhinagar

**Abstract-** Agile process management alludes to an iterative, incremental strategy for dealing with the plan and constructs exercises of building, data innovation and different business regions that intend to give new item or administration improvement in an exceptionally adaptable and intelligent way. The paper will endeavor to unite ideas and practices of 'lithe advancement', 'lean item improvement', 'choric frameworks', 'administration studies' and ideas of the 'learning association', together with 'the Model of Concurrent Perception' to recommend another approach of Software Development. This paper likewise concentrates on basics of Agile Process, its attributes, extent of deft process with its points of interest and different procedures for coordinated programming advancement XP and Scrum.

## I. INTRODUCTION

Light-footed SDLC model is a blend of iterative and incremental process models with concentrate on process versatility and consumer loyalty by fast conveyance of working programming item. Lithe Methods break the item into little incremental forms. These fabricates are given in emphases. Every emphasis ordinarily keeps going from around 7 to 21 days. Light-footed strategy is a product advancement technique that is individuals centered compensations situated, adaptable, fast, lean, responsive, and learning [1].

In spry programming advancement approach the errands are isolated to little time periods to convey particular elements for a discharge with in unbending and brief time traverse. For every particular assignment cycle is taken and working programming manufacture is conveyed after every emphasis. Each form is incremental as far as elements; usefulness and same stride will be proceeding till the required last item with all usefulness; highlights we get. What's more, the last form holds every one of the components required by the client. Here graphical portrayal for deft programming advancement approach is drawn for better comprehension the proposed approach.

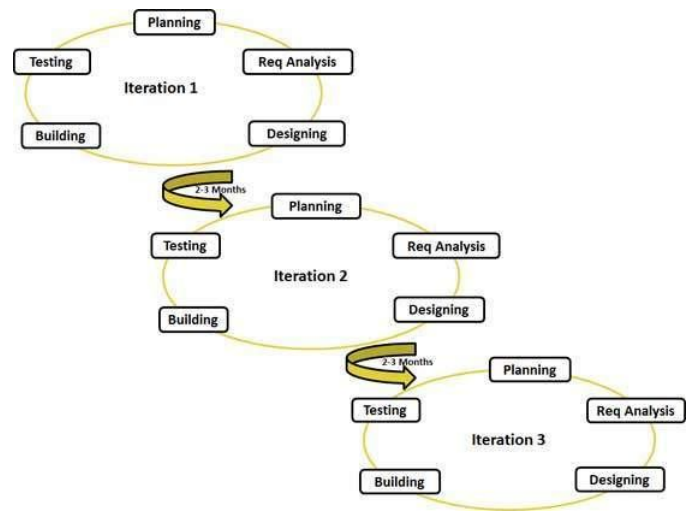


Figure 1. Graphical Model For Agile Methodology.

They had focused and suggested following features of Agile Software Development [3]:

- Individuals and communications - in dexterous advancement, self-association and motivation are critical, as are connections like co-area and combine programming [3].
- Working product module - Demo working programming can be seen as good practice for understanding our client requirements easily, rather than totally depending upon documentation & reports. In light of little working programming module correspondence turns out to be too simple and it will give clear thought to client too at each augmentation about the required their elements are splendidly coordinated or not [3].
- Customer coordinated efforts [3] - As the necessities can't be accumulated totally in the start of the venture because of different variables, as in beginning clients precisely now and then can't clarify about their completely coveted elements and usefulness in view of different reasons like poor specialized information. So constant communication & meeting with clients is very much vital to get legitimate item prerequisites.
- Responding to change - - spry advancement considers iterative approach for item improvement so at any phase

of client prerequisite changes then we can give him/her speedy reaction to fulfill his/her necessity change [3].

<b>Primary objectives</b>	<b>High safety</b>	<b>Quick value</b>
---------------------------	--------------------	--------------------

**II. COMPARISON OF AGILE SOFTWARE DEVELOPMENT & TRADITIONAL SOFTWARE DEVELOPMENT**

Table 1. Comparison of Agile SDLC & Traditional SDLC [3,9,10]

	<b>Traditional development</b>	<b>Agile development</b>
<b>Fundamental hypothesis</b>	Frameworks are completely specifiable, unsurprising and are created through amplified and point by point Arranging [9]	Excellent versatile programming is created by little groups that utilization the rule of ceaseless change of outline and testing in light of quick criticism and change [9]
<b>Management style</b>	Command and control	Leadership and collaboration
<b>Communication</b>	Formal	Informal
<b>User requirements</b>	Detailed and defined before coding/implementation	Interactive input
<b>Cost of restart</b>	High	Low
<b>Testing</b>	After coding is completed	After Every iteration
<b>Client involvement</b>	Low	High
<b>Appropriate scale of the project</b>	Large scale	Low and medium scale
<b>Requirements</b>	Very stable, known in advance	Emergent, with rapid changes
<b>Size</b>	Large teams and projects	Small teams and projects

**III. EXAMPLE OF AGILE DEVELOPMENT METHODOLOGY**

This section gives a brief idea about two agile methodologies.

**1) Extreme Programming (XP):**

(XP) is a product advancement strategy which is planned to improve programming quality. It also focus and gives response to change in customers’ needs after starting development. As a sort of spry programming improvement, it advocates visit "discharges" in short advancement cycles, which is planned to improve efficiency and it introduces checkpoints so at those points new requirements of the customers can be added easily. Extraordinary Programming (XP) [6] originators went for building up an approach reasonable for "question arranged ventures utilizing groups of a 12 or less developers in one area." [3]. The technique is based upon five essential standards: correspondence, effortlessness, input, boldness, and regard.

1. Communication [1]. XP has a culture of oral correspondence and its practices are intended to empower collaboration. The correspondence esteem depends on the perception that most venture troubles happen in light of the fact that somebody ought to have talked with another person to clear up a question, work together, or get offer assistance. "Issues with undertakings can constantly be followed back to some person not conversing with another person about something critical." [1,3]
2. Simplicity. Outline the most straightforward item that addresses the client's issues. An imperative part of the esteem is to just outline and code what is in the present prerequisites instead of to envision and plan for implicit necessities [1].
3. Feedback. The improvement group acquires input from the clients toward the finish of every emphasis and outside discharge. This input drives the following emphasis. Moreover, there are short outline and usage criticism circles incorporated with the strategy by means of match programming and test-driven improvement [1,3].
4. Courage. The other three qualities permit the group to have mettle in its activities and basic leadership. For

instance, the advancement group may have the bravery to oppose weight to make unlikely duties [1].

5. Respect. Colleagues need to think about each other and about the venture [1].

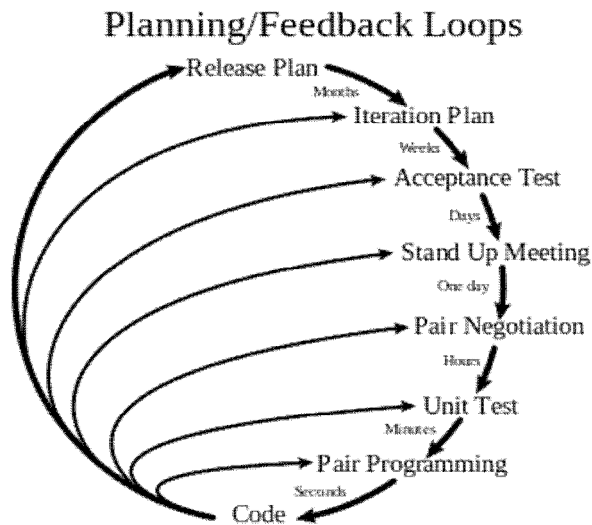


Figure 2. Graphical Model For Extreme Programming

## 2) Scrum:

It is an example of iterative, incremental development framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach" to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members. It also includes every days' face-to-face communication among all team members & branches involved in product development [2].

## IV. CONCLUSION

Agile development gives ideas to develop new software product according to customers' requirements as early as possible and after starting development at certain stage if customer requirement changes it can be handled efficiently & effectively. In Section I we have explained in detail with graphical diagram about working of Agile paradigm. Section II gives idea about comparisons of Agile Vs. Traditional Software development Paradigm. Where as in Section III we have presented overview about various examples (frameworks) for Agile Methodology.

## REFERENCES

- [1] Agile.csc.ncsu.edu
- [2] Manish Kumar, Santosh Kumar Singh, Dr. R. K. Dwivedi, "A Detail Study of Agile Software Development with Extreme Programming", IJARCSSE Vol. 5 Issue 10, Pg. Num 719-725.
- [3] www.greenstechnology.com
- [4] Lauri Williams, "A survey of Agile Development Methodologies", 2007.
- [5] [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)
- [6] BODJE N'Kauh Nathan-Regis and Dr G.M Nasira , "AGILE METHODS AND QUALITY \_ A SURVEY" CS & IT-CSCP 2013 , PG. Num 167-182.
- [7] Roy Morein, "An Agile Software Project Management Manifesto-A Reference Discipline framework for Agile Development" Int. J. Advance Soft Compu. Appl, Vol. 6, No. 1, March 2014.
- [8] Kuda Nageswara Rao, G. Kavita Naidu, Praneeth Chakka , "A Study of the Agile Software Development Methods, Applicability and Implications in Industry" International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011
- [9] Marin Stocia , "Sofyware Development : Agile Vs. Traditional" Informatica Economica Vol. 17 , n0. 4/2013.
- [10] [http://www.iacis.org/iis/2008\\_iis/pdf/s2008\\_95.pdf](http://www.iacis.org/iis/2008_iis/pdf/s2008_95.pdf)