*A Seminar Report on*

# Agile Methodology

*submitted in partial fulfillment of requirements for the award of degree of*

## Bachelor of Technology in Computer Science and Engineering

*submitted* by

` **K. Ritesh Yadav (16211A0583)**

*Under the Guidance of*

## Mrs.K. Anusha

Assistant professor



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**B V Raju Institute of Technology**

(UGC Autonomous ,NBA and NAAC Accredited) Vishnupur, Narsapur, Medak District - 502313.

### 2016 – 2020

**CERTIFICATE**

This is to certify that the Seminar entitled **“AGILE METHODOLOGY”** has been submitted by **K.Ritesh Yadav (16211A0583)** under my guidance in partial fulfillment of the requirements of Under Graduate Degree **Bachelor of Technology** in **Computer Science and Engineering** at **B.V.Raju Institute of Technology**, Narsapur.

**Date:**

**Place:** Narsapur

**Signature of Guide** **Signature of HOD**

**Mrs. K. Anusha,**   **Dr. Ch. Madhu Babu**

Assistant professor

## CONTENTS

Acknowledgement i

Abstract ii

Table of Contents

1. INTRODUCTION 1

1. HISTORY AND DETAILS 4

1. FEATURES 6

1. FRAMEWORK AND ARCHITECTURE 9

1. ADVANTAGES AND DISADVANTAGES 13
2. CONCLUSION 15

1. REFERENCES 17

## ACKNOWLEDGMENT

The success and final outcome of this presentation required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completion. Whatever I have done is only due to such guidance and assistance and I would not forget to thank them.

I respect and thank **Mrs.K. Anusha** for giving me an opportunity to give a presentation and providing all support and guidance which made me complete the work on time. I am extremely grateful to **Mr. Karthik Kovuri**, Projects & Seminars Coordinator for providing such a nice support and guidance. Another person who has my gratitude is **Dr. Ch. Madhu Babu,** Professor and Head, Department of CSE.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of Department of Computer Science which helped me in successfully completing the work. Also, I would like to extend our sincere regards to all the non-teaching staff of department of Computer Science for their timely support.

Signature of the student

**K. Ritesh Yadav**

**16211A0583**

## ABSTRACT

The agile software development methodology has recently become one of the most commonly used software development techniques. Rather than the long drawn out release cycles in the previously popular waterfall methodology, the agile technique suggests regular short sprint release cycles. This allows the customers and stakeholders to have more involvement within the software development process. This helps promote a higher quality final product because it combats the difficult task of a customer fully understanding and identifying all requirements in the software project planning phase. This also allows for the stakeholders to adjust the priorities of remaining tasks easily throughout the entire software development process.

Guide Signature Student Signature

Name Name

Designation Roll No

**CHAPTER 1**

**INTRODUCTION**

## 1. INTRODUCTION

The iterative approach has become vastly effective in helping software developers improve their skills in estimating schedule for remaining tasks. Schedule estimation is one of the most difficult responsibilities for developers because software issues are common and are unpredictable by nature. By breaking the large requirements down into more manageable sub requirements, the agile process naturally promotes better estimation.

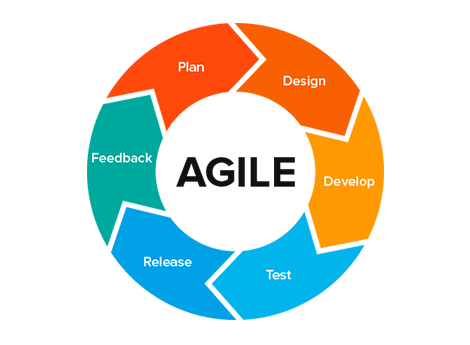
In today’s software development world, it is becoming more important than ever to keep the software stakeholders as involved as possible. With any software project there can be countless stakeholders who all can affect or can be affected by the outcome of the software. It is important for the software development team to identify the important stakeholders and find ways to connect with them individually to help promote stakeholder interest and involvement in the product. As agile promotes constant short release cycles, the stakeholders can see continual progress and make suggestions and develop new improved ideas for the software.

All Agile methodologies share a set of core ideas which are prescribed from method to method in subtly different ways; iterative and incremental delivery of working code, frequent collaboration with stakeholders, closely working, self-organizing teams, and the ability to embrace change late in the project. Agile methods are shamelessly incestuous, borrowing from each other and using existing ideas in slightly different ways so it becomes very difficult to tell whether a project is following any one method as even the slightest adaptation of any aspect of a process can make it seem more like another.

**What is Agile Methodology?**

Agile methodology can be defined as the process to develop software such as V-Model, Iterative Model, Waterfall Model, etc. It is basically a set of methods and practices that are based on the principles and values expressed in the Agile Manifesto. The whole process depends upon the coordination between the teams for proper utilization of the appropriate practices.

All the agile models come with iterations or Sprints that are much shorter in duration varying from two weeks to two months. During this period, the pre-determined features are delivered. The Agile models generally have one or more iterations and deliver the entire project at the end of the last iteration. All the feature in the agile way of working are completed in terms of development, testing, design and rework prior to the end of the process. All the work is performed in the single phase



**CHAPTER 2**

**HISTORY AND DETAILS**

**2. HISTORY AND DETAILS**

Iterative and Incremental development methods can be traced back as early as 1957, with evolutionary project management and adaptive software development emerging in the early 1970s.

During the 1990s, a number of *lightweight* software development methods evolved in reaction to the prevailing *heavyweight* methods that critics described as overly regulated, planned, and [micro-managed](https://en.wikipedia.org/wiki/Micromanagement). These included: [rapid application development](https://en.wikipedia.org/wiki/Rapid_application_development) (RAD), from 1991, the [unified process](https://en.wikipedia.org/wiki/Unified_Process) (UP) and [dynamic systems development method](https://en.wikipedia.org/wiki/Dynamic_systems_development_method) (DSDM), both from 1994; [Scrum](https://en.wikipedia.org/wiki/Scrum_(software_development)), from 1995; Crystal Clear and [extreme programming](https://en.wikipedia.org/wiki/Extreme_programming) (XP), both from 1996; and [feature-driven development](https://en.wikipedia.org/wiki/Feature-driven_development), from 1997. Although these all originated before the publication of the *Agile Manifesto*, they are now collectively referred to as agile software development methods. At the same time, similar changes were underway in manufacturing and aerospace.

In 2001, these seventeen software developers met at a resort in [Snowbird](https://en.wikipedia.org/wiki/Snowbird,_Utah), [Utah](https://en.wikipedia.org/wiki/Utah) to discuss these lightweight development methods: [Kent Beck](https://en.wikipedia.org/wiki/Kent_Beck), [Ward Cunningham](https://en.wikipedia.org/wiki/Ward_Cunningham), [Dave Thomas](https://en.wikipedia.org/wiki/Dave_Thomas_(programmer)), [Jeff Sutherland](https://en.wikipedia.org/wiki/Jeff_Sutherland), [Ken Schwaber](https://en.wikipedia.org/wiki/Ken_Schwaber), [Jim Highsmith](https://en.wikipedia.org/wiki/Jim_Highsmith), [Alistair Cockburn](https://en.wikipedia.org/wiki/Alistair_Cockburn), [Robert C. Martin](https://en.wikipedia.org/wiki/Robert_C._Martin), [Mike Beedle](https://en.wikipedia.org/wiki/Mike_Beedle), [Arie van Bennekum](https://en.wikipedia.org/w/index.php?title=Arie_van_Bennekum&action=edit&redlink=1), [Martin Fowler](https://en.wikipedia.org/wiki/Martin_Fowler_(software_engineer)), James Grenning, [Andrew Hunt](https://en.wikipedia.org/wiki/Andy_Hunt_(author)), [Ron Jeffries](https://en.wikipedia.org/wiki/Ron_Jeffries), [Jon Kern](https://en.wikipedia.org/w/index.php?title=Jon_Kern&action=edit&redlink=1), [Brian Marick](https://en.wikipedia.org/w/index.php?title=Brian_Marick&action=edit&redlink=1), and [Steve Mellor](https://en.wikipedia.org/wiki/Stephen_J._Mellor). Together they published the *Manifesto for Agile Software Development*.

In 2005, a group headed by Cockburn and Highsmith wrote an addendum of [project management](https://en.wikipedia.org/wiki/Project_management) principles, the [PM Declaration of Interdependence](https://en.wikipedia.org/w/index.php?title=PM_Declaration_of_Interdependence&action=edit&redlink=1),[[18]](https://en.wikipedia.org/wiki/Agile_software_development%2523cite_note-18) to guide software project management according to agile software development methods.

In 2009, a group working with Martin wrote an extension of [software development](https://en.wikipedia.org/wiki/Software_development) principles, the [Software Craftsmanship Manifesto](https://en.wikipedia.org/wiki/Software_craftsmanship), to guide agile software development according to [professional](https://en.wikipedia.org/wiki/Professional) conduct and mastery.

In 2011, the Agile Alliance created the *Guide to Agile Practices* (renamed the *Agile Glossary* in 2016), an evolving open-source compendium of the working definitions of agile practices, terms, and elements, along with interpretations and experience guidelines from the worldwide community of agile practitioners.

**CHAPTER 3**

**FEATURES**

**3. FEATURES OF AGILE METHODOLOGY**

**1. Iterative:**

The main objective of agile software processes is satisfaction of customers, so it focuses on single requirement with multiple iterations.

**2. Modularity:**  
Agile process decomposes the complete system into manageable pieces called modules. Modularity plays a major role in software development processes.

**3. Time Boxing:**  
As agile process is iterative in nature, it requires the time limits on each module with respective cycle.

**4. Parsimony:**  
In agile processes parsimony is required to mitigate risks and achieve the goals by minimal number of modules.

**5. Incremental:**  
As the agile process is iterative in nature, it requires the system to be developed in increments, each increment is independent of others, and at last all increments are integrated into complete system.

**6. Adaptive:**  
Due to the iterative nature of agile process new risks may occurs. The adaptive characteristic of agile process allows adapting the processes to attack the new risks and allows changes in the real time requirements.

**7. Convergent:**

All the risks associated with each increment are convergent in agile process by using iterative and incremental approach.

**8. Collaborative:**

As agile process is modular in nature, it needs a good communication among software development team. Different modules need to be integrated at the end of the software development process.

**9. People Oriented:**  
In the agile processes customer satisfaction is the first priority over the technology and process. A good software development team increases the performance and productivity of the software.

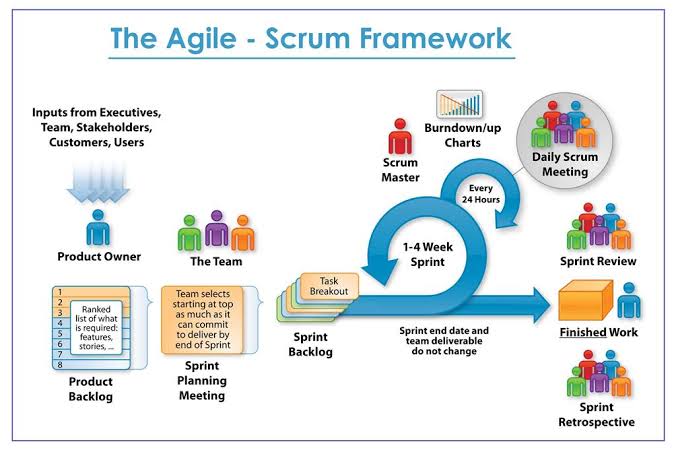
**10. The Right Final Product:**

Finally, one of the most crucial characteristics of Agile projects is that the focus is on delivering the right final product.

**CHAPTER 4**

**FRAMEWORK AND ARCHITECTURE**

**4. FRAMEWORK AND ARCHITECTURE**

**The Agile-Scrum Framework:**

* **Product Owner:**

The Product Owner (PO) is a member of the Agile Team responsible for defining Stories and prioritizing the Team Backlog to streamline the execution of program priorities while maintaining the conceptual and technical integrity of the Features or components for the team.

* **Product Backlog:**

The agile product backlog in Scrum is a prioritized features list, containing short descriptions of all functionality desired in the product.

* **Sprint Planning:**

Sprint planning is an event in the Scrum framework where the team determines the product backlog items they will work on during that sprint and discusses their initial plan for completing those product backlog items.

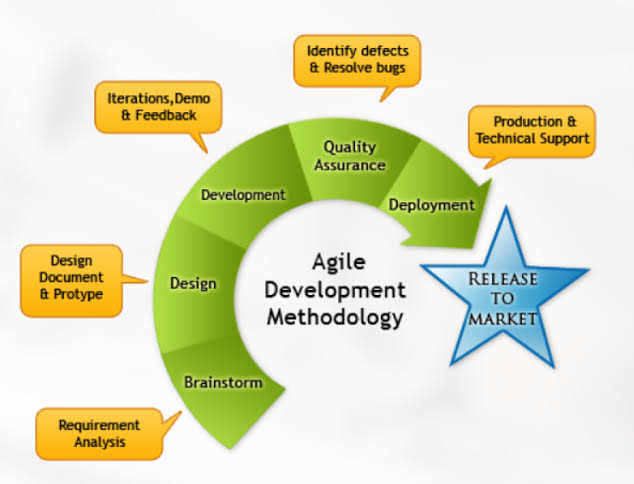
* **Daily Stand-up meeting:**

Daily stand-up, as the name suggests, is a daily status meeting among all the members of an agile team. It not only provides a forum for regular updates but also brings the problems of team members into focus so that it can be quickly addressed.

* **Sprint Retrospective meeting:**

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is at most a three-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose.

**ARCHITECTURE**



## CHAPTER 5

**ADVANTAGES AND DISADVANTAGES**

### 5. ADVANTAGES AND DISADVANTAGES

**ADVANTAGES OF AGILE MODEL:**

* Customer satisfaction by rapid, continuous delivery of useful software.
* People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
* Working software is delivered frequently (weeks rather than months).
* Face-to-face conversation is the best form of communication.
* Close, daily cooperation between business people and developers.
* Continuous attention to technical excellence and good design.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed.

**DISADVANTAGES OF AGILE MODEL:**

* In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
* There is lack of emphasis on necessary designing and documentation.
* The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
* Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

Agile Developments also fails at times due to Unrealistic Expectations – Agile actually is and what it can help you achieve. Agile is commonly believed to be a set a practices, processes and tools, when in fact, Agile is really more of a mind-set and culture.

**CHAPTER** **6**

**CONCLUSION**

### 6.CONCLUSION

The agile software development methodology is being widely accepted within the software development community. Agile provides multiple benefits over the previously used waterfall methodology. Agile attempts to simplify the software planning and estimation process by decomposing large requirements into small individual tasks. Analyzing small tasks allow the software development team to more accurately predict the level of effort required in order to implement the change. This allows the project manager to accurately depict the percentage complete of the software which allows them to continually track overall project progress against the originally planned progress. The agile process also is designed to help train developer in their schedule estimating skills throughout the lifecycle. For each task the developer should be required to make an estimation of how long they believe they will need to complete the task, after the task is completed they should enter in the actual time spent on the task. This will show the developer the delta between their estimation and their actual time spent.

**CHAPTER 7**

**REFERENCES**

**7. REFERENCES**

1. Ghule, S. (2014). Risk analysis and mitigation plan in software development. International Journal of Engineering, Sciences and Research Technology, 3(8), 546-548.

2. Wysocki, R. K. (2013). Effective project management : Traditional, agile, extreme. Hoboken: Wiley.

3.Heeager, Lise Tordrup . "INTRODUCING AGILE PRACTICES IN A DOCUMENTATION-DRIVEN SOFTWARE DEVELOPMENT PRACTICE: A CASE STUDY." Journal of Information Technology Case and Application Research 14.1 (2012): 3-24.