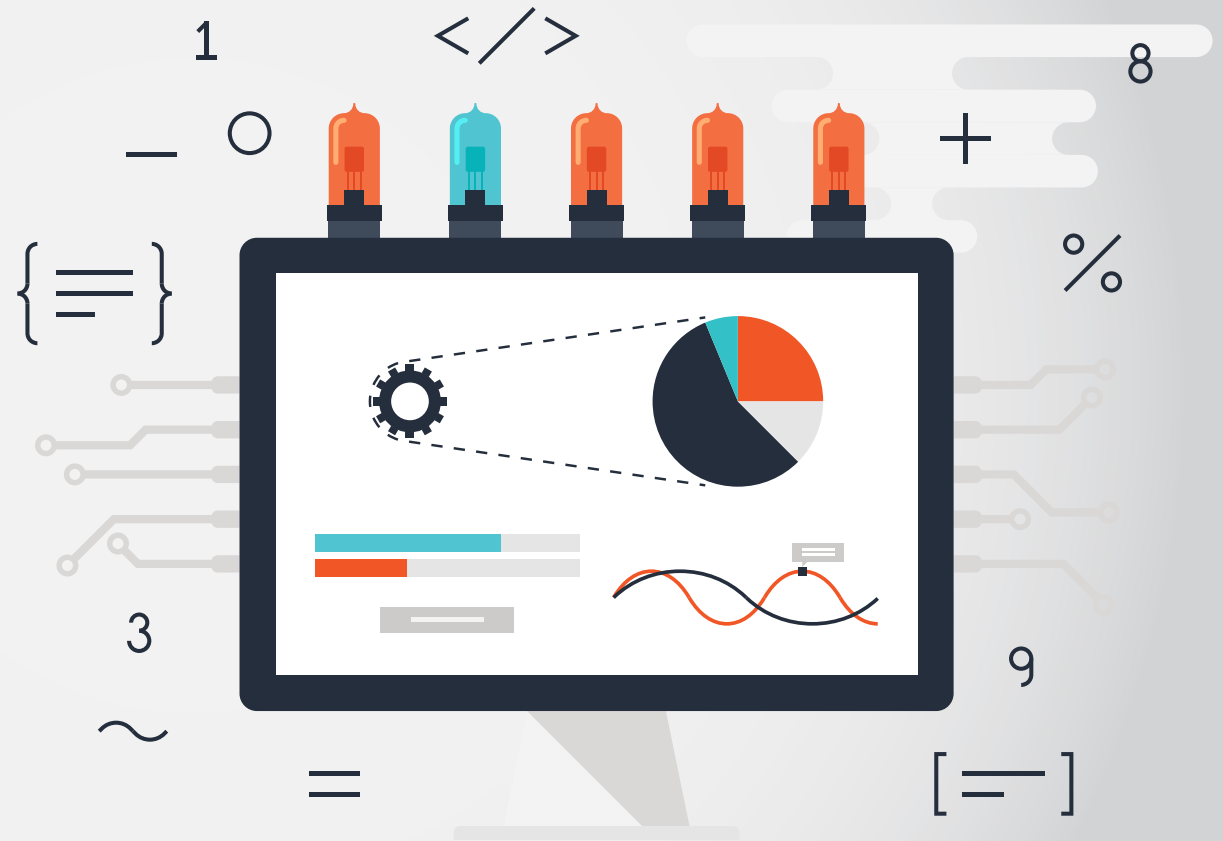


# Agile & Methodologies & Key Principles Series-I

# Introduction

Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change.

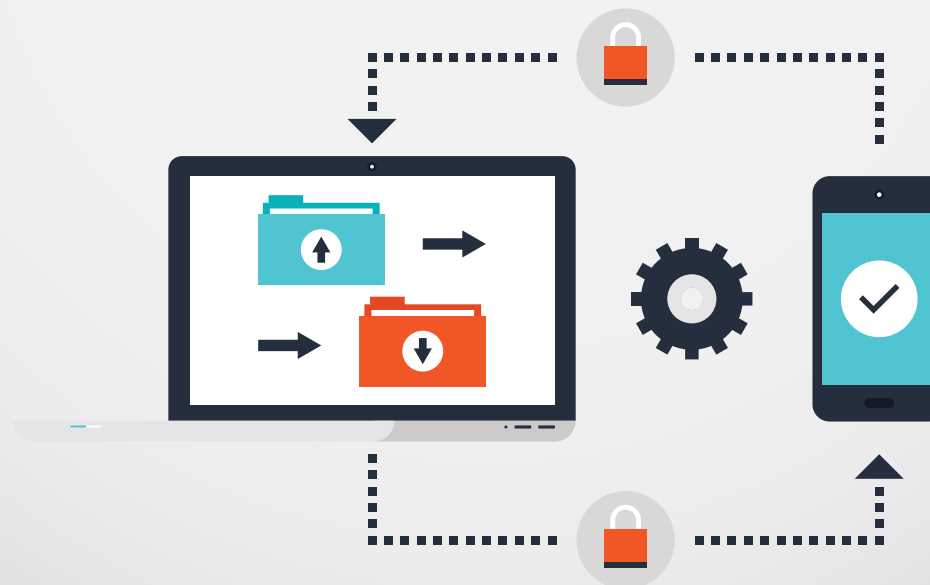
The Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications.



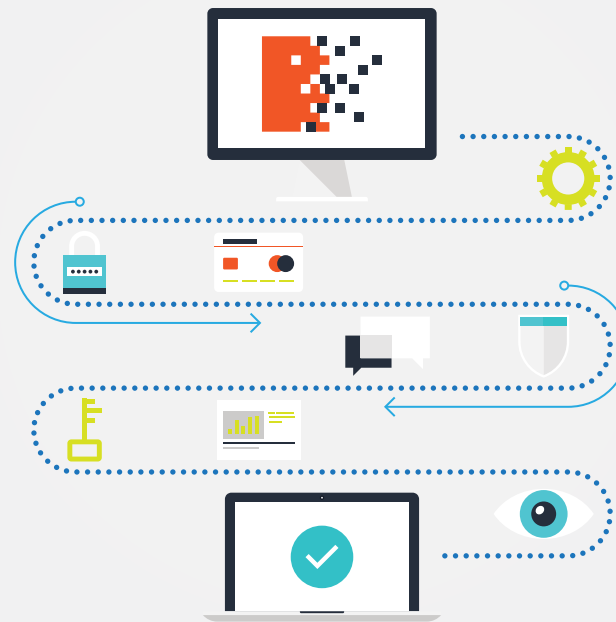
## Shortcomings of Traditional Approaches

The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance. This method has several shortcomings which can impede the progress of a project. The pro and anti waterfall model camps had their own take on the subject—until the all encompassing Agile model was introduced.

In his paper titled, “Managing the Development of Large Software Systems,” in 1970, noted American computer scientist and pioneer in the field of software development, Dr. Winston W. Royce picked apart sequential development. He explained that software should not be developed like a vehicle in a factory assembly line; wherein every component is combined into a whole, in a set of sequential phases, where every such phase depends on what has gone before.



Therefore, Dr. Royce opined that it was not appropriate to adhere to the phase-based approach; one where the developers begin by gathering project requirements before moving on to finish work on all of its architecture design, code and the like. By backing his beliefs with data, Royce particularly disapproved of the deficiency in communication among various specialized groups that are tasked with completing every work phase.

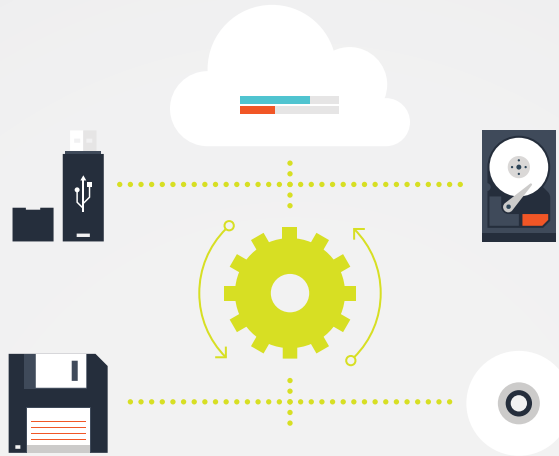


In 1974, E. A. Edmonds wrote a paper that introduced an adaptive software development process. Years later, in February 2001, a group of 17 software developers met at the Snowbird resort in Utah to discuss lightweight development methods. This grand association of knowledgeable minds later led to the publishing of the “Manifesto for Agile Software Development”.

## The Agile Manifesto is based on 12 Principles

- » Customer satisfaction by rapid delivery of useful software.
- » Welcome changing requirements, even late in development.
- » Working software is delivered frequently (weeks rather than months).
- » Close, daily cooperation between business people and developers.
- » Projects are built around motivated individuals, who should be trusted.
- » Face-to-face conversation is the best form of communication (co-location).
- » Working software is the principal measure of progress.
- » Sustainable development, able to maintain a constant pace.
- » Continuous attention to technical excellence and good design.
- » Simplicity—the art of maximizing the amount of work not done—is essential.
- » Self-organizing teams.
- » Regular adaptation to changing circumstance.

## Old to New: The Software Journey



It is quite simple to see the areas of concern with the waterfall method, which is a sequential design process, used in software development processes. The basic premise and assumption of this method is that it presumes that every requirement can be identified well in advance—before any design or coding can be done.

Software developers everywhere have grasped such situations by enduring considerable hardships. At the end of a project, a team might have built the software it was asked to build, but, in the time it took to create, business realities have changed so drastically that—odd as it may sound—the end product is quite immaterial.

Little wonder then that nowadays many organizations maintain a studied silence regarding their use of the waterfall or traditional command and control methods; mainly because of the notion that they will be frowned upon.



## Why Agile?

The Agile software development method allows for many possibilities to evaluate the direction throughout the development lifecycle. This is achieved through regular cadences of work, known as Sprints or iterations, by the culmination of which teams must offer a shippable product increment. Owing to the repetition of abbreviated work cycles and functional product they yield, the agile methodology is well-known for being “iterative” and “incremental.”

In the near-obsolete waterfall method, development teams get only one opportunity to ensure every single aspect of the project has gone according to plan. However, the opposite holds true for the agile paradigm where all aspects of development, ranging from requirements, design and so on are revisited often. Consider this: A team gains the capacity to channelize their project in an entirely different direction if they have the benefit of re-evaluating their competencies bi-weekly.





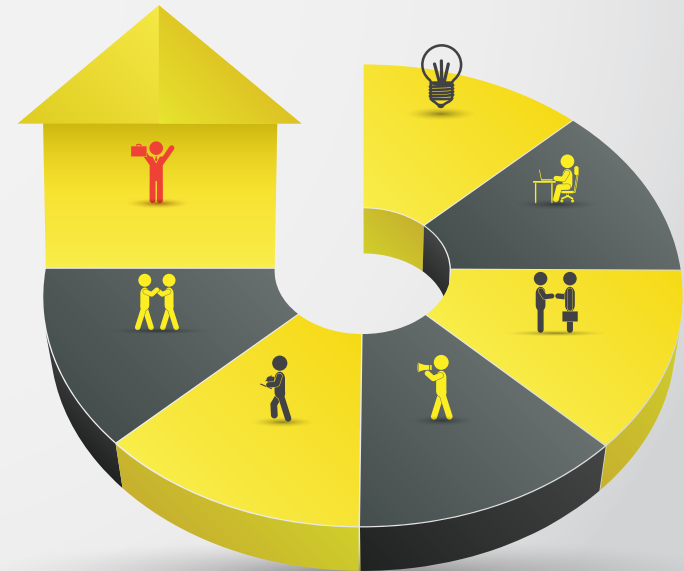
As a direct consequence of this possibility, the “inspect-and-adapt” approach to development helps to reduce cost of development and also, time to market. Because a team’s work cycle is limited to two weeks, stakeholders have recurring opportunities to calibrate releases that can succeed in the market.

Agile development helps companies build the right product. Instead of committing to market a piece of software that hasn’t been written yet, agile empowers teams to continuously re-plan their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. The agile development method enables the preservation of a product’s crucial relevance to the market; thus, negating non-release situations.



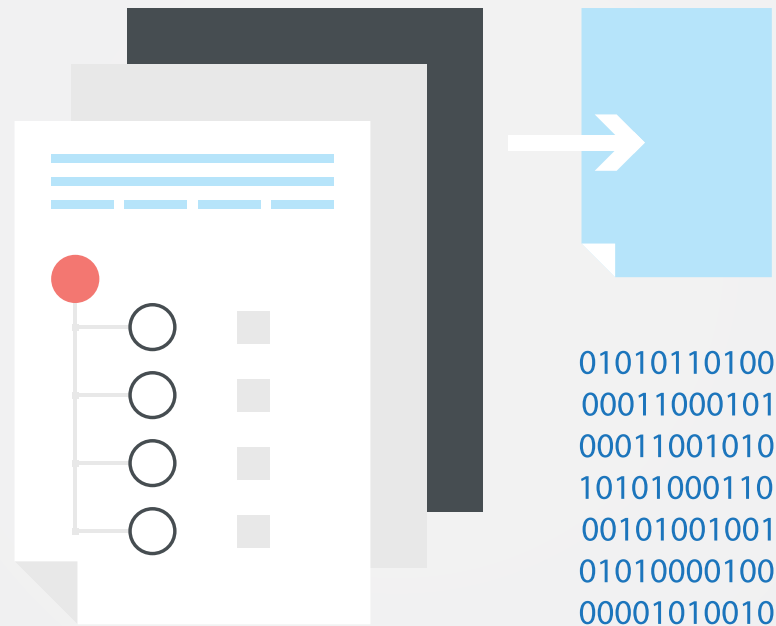
## Advantages of Agile model

- » Customer satisfaction by rapid 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 welcome.



## Pressing need to introduce Agile

Agile provides a certain freedom and opportunities for far-reaching changes, which make the model highly significant. New changes can be introduced rather easily, and at minimal spend, because of the frequency of new increments that are produced. If developers intend to implement some new features, they stand to forfeit only a few days' worth of work (at times no more than a few hours), to get back to the drawing board and originate it afresh.



Contrary to the waterfall model, hardly any planning is required to jump-start a project based on the agile model. Given that we inhabit an ever-changing IT and business ecosystem, Agile presumes that end users' needs are also dynamic. This affords possibilities for changes to be discussed which result in the introduction or removal of features depending on the feedback received. This caters efficiently to the unique needs of the customer.

## Relevance of Scrum



Scrum is an iterative and incremental agile software development methodology for managing product development. It is renowned for its simplicity and immense flexibility which is why it is considered as the best means to introduce Agility. Several enterprises state that they are “Doing Scrum” to let the market know they follow the latest methodologies.

Scrum accentuates empirical feedback, team self-management and aiming to build well-tested product increments, all in short iterations. The practice of Doing Scrum is at odds with existing processes in companies that can be classified as being Non-Agile. It is basically like trying to fit a square peg in a round hole!

There are several methodologies that come under the banner of agile, more so because they increase the values of the Agile Manifesto. Some of the most widely-approved ones are:

- **DSDM** is probably the original agile development method. DSDM was around before the term 'agile' was even invented, but is based on all the principles we've come to call as agile.
- **Scrum** is also an agile development method, which concentrates specifically on how to manage tasks within a team-based development environment.
- **XP (Extreme Programming)** is a more radical agile methodology, focusing more on the software engineering process and addressing the analysis, development and test phases with unique approaches that make a substantial difference to the quality of the end product.
- **DSDM** is probably the most complete agile methodology, whereas Scrum and XP are easier to implement and complementary; because they tackle different aspects of software development projects and are both founded on very similar concepts.



## Conclusion

In agile software projects, project management tend to often assume different proportions, depending on the aptitude of the project manager in fields such as communication, facilitation, coordination. There is minimal emphasis on actual planning and control.

Collaboration and enhanced visibility can help to contribute greater experiences on every level for teams to develop the best software products. Agile development is clearly the leader, as waterfall method doesn't make the cut as it needs a considerable amount of documentation and lacks adequate flexibility.



## About Orchestrate

Orchestrate is a US based business process management organization with Headquarters in Dallas, Texas. Orchestrate offers services to the diverse outsourcing requirements of clients in an extensive range of businesses including IT, finance, mortgage and contact center. We provide a comprehensive suite of technology and services to our clients that help accelerate sales and boost their profit. Our comprehensive solutions and services help SMEs and enterprises to implement technologies and processes that boost their profitability across the organization.



1330 Capital Parkway, Carrollton TX 75006

Toll Free: 800-232-5130

| success@orchestrate.com

[www.orchestrate.com](http://www.orchestrate.com)

