CITY UNIVERSITY

AssignmentNo -01

Department Name: Computer Science & Engineering

sCourse Code :CSE-325

Course Name:System Analysis & Design

Assignment Name: Agile Development Methology Process

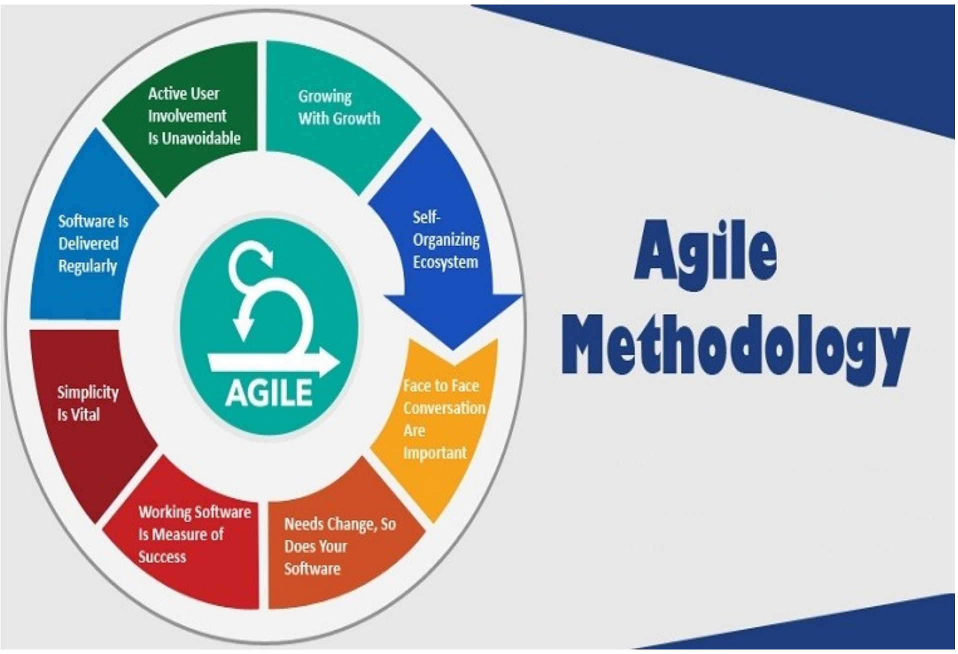
Submitted BY

Name:Sharmin
#ID: 171442621
Program: CSE(Evn)
Batch: 44th

Submitted To

Supta Richard Philip
Sr.Lecturer, Department
City University,

Date Of Submission:


Active User Involvement Is Unavoidable
Growing With Growth
Self-Organizing Ecosystem
Face to Face Conversation Are Important
Needs Change, So Does Your Software
Working Software Is Measure of Success
Simplicity Is Vital
Software Is Delivered Regularly
AGILE
Agile Methodology

Contents

What is the Methodology?

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.[1]

A methodology does not set out to provide solutions—it is therefore, not the same as a method. Instead, a methodology offers the theoretical underpinning for understanding which method, set of methods, or best practices can be applied to a specific case, for example, to calculate a specific result.

It has been defined also as follows:

1. "the analysis of the principles of methods, rules, and postulates employed by a discipline";[2]
2. "the systematic study of methods that are, can be, or have been applied within a discipline";[2]
3. "the study or description of methods".[3]

What is the waterfall development methodology?

The **waterfall model** is a relatively linear sequential design approach for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, deployment and maintenance.

The waterfall development model originated in the manufacturing and construction industries; where the highly structured physical environments meant that design changes became prohibitively expensive much sooner in the development process. When first adopted for software development, there were no recognized alternatives for knowledge-based creative work.[1]

Rapid application development methodology

Rapid-application development (**RAD**), also called **Rapid-application building** (**RAB**), is both a general term, used to refer to adaptive software development approaches, as well as the name for James Martin's approach to rapid development. In general, RAD approaches to software development put less emphasis on planning and more emphasis on an adaptive process. Prototypes are often used in addition to or sometimes even in place of design specifications.

RAD is especially well suited for (although not limited to) developing software that is driven by user interface requirements. Graphical user interface builders are often called rapid application development tools. Other approaches to rapid development include the adaptive, agile, spiral, and unified models.

Agile development methodology

Agile software development is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s).[1] It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.[2][*further explanation needed*]

The term *agile* (sometimes written *Agile*)[3] was popularized, in this context, by the *Manifesto for Agile Software Development*.[4] The values and principles espoused in this manifesto were derived from and underpin a broad range of software development frameworks, including Scrum and Kanban.[5][6]

There is significant anecdotal evidence that adopting agile practices and values improves the agility of software professionals, teams and organizations; however, some empirical studies have found no scientific evidence.[7][8]

Devops development methodology

DevOps (**development** and operations) is an enterprise software **development**phrase used to mean a type of **agile** relationship between **development** and IT operations. The goal of **DevOps** is to change and improve the relationship by advocating better communication and collaboration between these two business units.

Introduction and Definition of agile methodology

Agile is one form of software development methodology. Its main focus is on client satisfaction through continuous delivery. The focus of Agile is more on limiting the project scope. An agile project sets a minimum number of requirements and turns them into a deliverable product.

Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental”. In waterfall, development teams only have one chance to get each aspect of a project right.

(Haag & Cummings, 2009) says an Agile project sets a minimum number of requirements and turns them into a deliverable product. Agile means what it sounds like: fast and efficient; small; lower cost; fewer features; shorter projects.

In February 2001, the Manifesto for Agile Software Development (The Agile Manifesto, 2001) was created by seventeen people with desires to find alternative approaches to software development. Each of them played a prominent part in the opposition of the prevailing software development processes, which they considered rigid, heavyweight and too focused on documentation. Their response, summarized in the manifesto, clarifies their focus by valuing:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

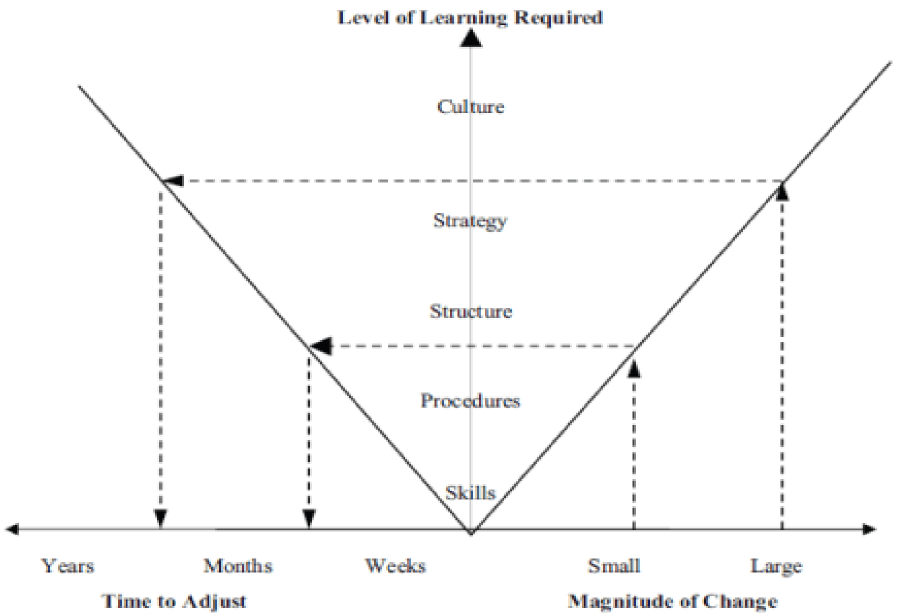
In Agile literature, Agile methods generally denote a family of methods under the umbrella of the Agile Alliance, including: extreme Programming, Scrum, Dynamic Systems Development Method, Crystal Methods, Feature-Driven Development, Lean Development and Adaptive Software Development. Although differing in specific techniques, these methods have much in common, including short iterative life cycles, quick and frequent feedback from customers, and constant learning. Among them, Scrum and XP/Scrum hybrid are by far the most widely adopted in the past decade. Agile processes bring about a dramatic increase in productivity and quality. This is achieved through a high degree of communication and interaction, short iterative development and a strong sense of team responsibility.

However, there has been some criticism of Agile as well. (Kruchten, 2011) compared Agile methodology to a teenager: very self-conscious, checking constantly its appearance in a mirror, accepting few criticisms, only interested in being with its peers, rejecting en bloc all wisdom from the past, just because it is from the past, adopting fads and new jargon, at times cocky and arrogant.

(Boehm & Turner, 2004) convincingly argue that there is a pragmatic need to balance stability and agility. They analyze the home grounds of Agile and traditional approaches based on application characteristics, management characteristics, technical characteristics, and personnel characteristics. Further, they assert that the choice of traditional or agile methods for a given project is largely contingent on five factors:

1. The size of the systems development project and team
2. The consequences of failure (i.e., criticality)
3. The degree of dynamism or volatility of the environment
4. The competence of personnel
5. Compatibility with the prevailing culture

(Krill, 2013) also pointed out that barriers to Agile adoption include an inability to change an organization’s culture, followed by general resistance to change and trying to fit agile into a non-agile framework. The framework for organizational change articulated by Adler and Shenhar (1990) is useful for assessing the effort required to meet these challenges. The biggest concerns about Agile include lack of upfront planning, loss of management control, and management opposition. Other reasons include communication problems between development teams and other areas of the business and problems with the Scrum master.



Framework for Organizational Change (Adler & Shenhar, 1990)

Scrum

The founder of Scrum (Schwaber & Sutherland, 1995) described Scrum as a process framework that has been used to manage complex product development since the early 1990s. Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques.

Scrum is about organizing people and work into short "sprints" of activity, to develop code in short, small chunks, rather than building one big monolithic blob of code that takes forever to build, test and "drop" into the system.

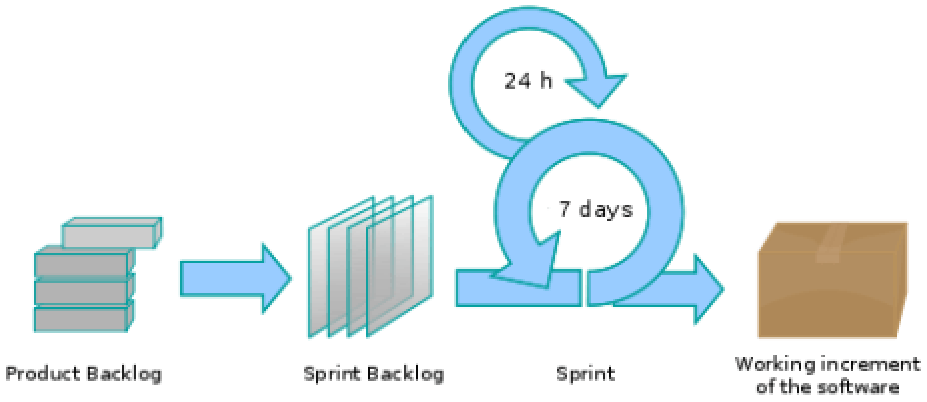


Illustration of Scrum framework

Three distinct roles are identified within the Scrum methodology:

1. The Scrum master, who ensures the process is followed, removes impediments, and protects the Development Team from disruption
2. The Product Owner, who represents the stakeholders and the business
3. The Development Team, a cross-functional, self-organizing team who do the actual analysis, design, implementation, testing, etc.

Scrum is a suitable method to support development process. It is very efficient in creating solutions with the highest business value in the shortest possible time. Due to the daily scrums, any impediments are known to everyone as they occur making it possible to resolve them as quickly as possible. Furthermore, it adds support to prioritize work and closely monitor the progress of a project with little to no overhead.

Conclusion

Conclusion. If you want to be **agile**, the first tenet is "Every project needs a slightly different set of policies and conventions, or **methodology**." People's attitude toward communication, user involvement, and frequent releases is more important than the specific **process** you use.

Reference

Web Sites http://www.agilelogic.com/ Check into our resources section

http://www.agilealliance.org/ Agile Alliance organization

http://www.scrumalliance.org/ Scrum Alliance organization

http://www.xprogramming.com Ron Jeffries' site, XP-oriented but great for agile ideas

http://www.refactoring.com/ Home for refactoring and tools

http://www.junit.org/ Home for JUnit and testing tools

http://www.eclipse.org/ Eclipse and plug-ins for agile support