1. Introduction

The economy in recent years has had a major impact on the software industry. Because of the increased importance of bringing software products to the market quickly, revision of traditional methods has been apparent. Agile methods are techniques of software development designed to deliver these products quickly, with more functionality necessary to users.

In this topic, we will look at how agile methods are used to deliver software systems at a faster rate and with improved functionality.

Objectives: Agile Methods

Upon completion of this topic, you should be able to
- describe how agile methods are used to develop software systems faster
- explain the principles behind agile methods

2. Agile Programming

There are many different techniques that have been used in agile programming. These include collaborative techniques (e.g., pair programming, having customers work on site as team members). What this amounts to is a much closer working with customers to ensure that the software products developed are what customers need. For instance, a common agile practice is to have customers write acceptance tests.

The agile manifesto site provides a manifesto to uncover better ways of developing software. This approach emphasises
- focusing on individuals and their interactions rather than processes and tools
- delivering a working software rather than a comprehensive documentation
- customer collaboration rather than contract negotiation
- responding to change rather than following plans

This manifesto clearly diverges from the principles of the Capability Maturity Model (CMM) that we will discuss later. The agile approach maximises co-operation with users. This is very critical for the success of information systems (IS) projects. It also parallels the Spiral method of systems development, which will be discussed later as a way to control project risk.

Click each tab to learn about the principles and practices of agile programming.

Principles and Practices of Agile Programming

The following principles lie at the foundation of the agile movement:
1. Highest priority is given to satisfy customers through early and continuous delivery of valuable software
2. Requirement changes are incorporated, even late in the development process. Agile processes harness change with the aim of improving customer competitive advantage
3. Working software is delivered frequently, from a couple of weeks to a couple of months, preferably in shorter timescales
4. Co-operative work among business people and developers is needed on a daily basis
5. Projects are built around motivated individuals, who are trusted to get the job done and who operate in a positive environment with all the support they need
6. One-to-one communication is relied upon as the most efficient and effective method
7. The primary measure of progress is the working software
8. Agile processes promote sustainable development
9. Technical excellence and good design are sought continuously, enhancing agility
10. Simplicity is essential, do not unnecessarily over-complicate the project
11. Self-organising teams develop the best architectures, requirements and designs
12. Teams reflect on ways to become more effective at regular intervals, and tune and adjust behaviour accordingly

**Practices in agile programming**

Riefer (2002) reported results of a survey of 14 firms on the use of agile programming. These organisations cited the following agile practices:

1. Collective ownership
2. Concurrent development
3. Continuous integration
4. Customer collaboration
5. Daily stand-up meetings
6. Product demonstrations rather than documentation
7. Full stakeholder participation
8. Just-in-time requirements
9. Nightly product builds

The survey by Riefer (2002) covered 31 projects where agile methods were used. Agile methods were applied to these projects because of poor past performance in delivering acceptable products to the market on time and within budget. All projects were developed within the firm and lasted less than one year.

Projects were characterised as having stable requirements and high degrees of development flexibility. Most organisations were Level 2 or greater under the Capability Maturity Model (CMM). CMM assesses an organisation’s capability to deliver high-quality software using a scale that goes from 1 (initial) to 5 (optimising). CMM is explained in greater detail later on. Eight of the projects were by Level 5 organisations. Level 5 organisations were willing to try agile methods because they were having problems meeting delivery expectations. These organisations tended to modify agile methods to fit their style of software development.

The actual processes used varied, but included the Spiral method. The results were quite positive. Productivity gains averaged 15% to 23% based on published industry
benchmarks. Costs were reduced by 5% to 7% against published industry benchmarks. Time to market was reduced by 25% to 50% compared to previous projects these firms had worked on. The quality of the software products was similar to products developed using more traditional techniques. All the participants were convinced that agile methods were an improvement.

Agile methods have proven very effective when applied to small projects. They also work (with care) for larger projects, although effort is needed to get teams to work well together. The best case is when a small, self-organised and motivated team located in the same location works with a small number of on-site customers. There are even cases when teams have worked in a geographically separated environment, although synchronisation of efforts is required.

Click the link below for an example of an Agile method of software development.

Rational unified process

Rational unified process

Rational, a software developer now owned by IBM, developed an agile method known as the rational unified process (RUP). RUP is probably the most widely used agile method that is supported by a growing number of modelling tool vendors.

RUP breaks the software life cycle into a number of cycles, each of which consists of the following four phases:

1. Inception – The business need is identified and project scope delimited
2. Elaboration – The requirements are elaborated in detail, the architectural impact is assessed, project risks are identified and a project plan is drawn up
3. Construction – The relevant features are developed and integrated with the rest of the system and tested
4. Transition – The solution is released and rolled out to the user community

For more details about RUP, you can read the white papers found at IBM’s library.

3. Exercise

Click the link below for an exercise to examine your knowledge of agile practices.

Exercise: Agile Practices

Exercise: Agile Practices

Exercise Alternate Text

South Belle Hair Care, a cosmetics company is planning to automate its sales division. The project would involve maintaining and updating their sales regions, details of sales persons, sales for different months, target sales, etc. The project team has decided to use an agile approach for developing the new automated
system.

Answer the following questions regarding practices for agile development.

**Q1.** You need to monitor the progress of the project.
Which one of the following is the primary measure of progress?
1. The working system you incrementally develop
2. The percentage of work activities completed so far

The correct answer is
- option 1, The working system you incrementally develop

**Q2.** The users keep changing their mind about the functionality of the system. In fact, they add between 4–8 requirements per week to the scope of the project. Which one of the following represents the action you need to adopt?
1. Baseline the changes and discourage further requirement changes
2. Incorporate the changes and assess how urgent they are

The correct answer is
- option 2, Incorporate the changes and assess how urgent they are

**Q3.** You wish to ensure users are clear about the system being developed. Which one of the following should you do?
1. Provide product demonstrations as frequently as required
2. Ensure the requirements documents are as detailed and complete as possible

The correct answer is
- option 1, Provide product demonstrations as frequently as required

**Q4.** Which one of the following should you implement in order to ensure that the development team has total control of the schedule?
1. The team should have complete ownership of the system and pass this ownership onto the users once they are satisfied with the system
2. The team and users should take joint ownership of the system right from the start

The correct answer is
- option 2, The team and users should take joint ownership of the system right from the start

---

**4. Self-Assessment**

Now, try the self-assessment questions to test your understanding of the topic. Click the following link to open the Self-Assessment in a new window.

Self-Assessment

Q1. Which one of the following best describes agile methods?
1. It is an implementation of Level 5 of the Capability Maturity Model (CMM)
2. It involves thorough documentation
3. It expends a great deal of time with users to develop all the requirements before the project begins
4. It sacrifices formality for practicality

Q2. Which one of the following defines the objective of agile methods?
1. It is designed to get software projects completed quickly
2. It is designed to get software projects accomplished with maximum functionality
3. It isolates systems developers from users to allow them to apply concentrated effort
4. It carefully follows the initial plan to maintain project focus

Q3. Which one of the following is a principle of agile methods?
1. It refrains from obtaining user feedback until the system is completed
2. It measures productivity by lines of code completed
3. It measures productivity by working software
4. It focuses on the project leader as the owner of the software

5. Summary

This topic covered the following main points:
• As a response to the poor performance record of IS with respect to time, budget and functionality, some in the software industry have developed collaborative techniques known as agile methods to develop software much faster and closer to user requirements.
• The common base for agile methods include:
  o Working closely with users throughout the project.
  o Iteratively developing software versions in the spirit of prototyping.
  o Demonstrating the prototypes with users to identify required changes in direction.
  o Avoiding heavy documentation in favour of developing software faster.