Computers II Lesson 3

3.0 Agile software development

Businesses now operate in rapidly changing environment.

They have to respond to new opportunities and markets, changing economic conditions, and the emergence of competing products and services.

We have said that Software is part of almost all business operations, new software must be developed quickly to take advantage of new opportunities and to respond to competitive pressure.

Rapid development and delivery is therefore now often the most critical requirement for software systems. In fact, many businesses are willing to trade off software quality and compromise on requirements to achieve faster deployment of the software that they need.

- Software development processes that plan on completely specifying the requirements and then designing, building, and testing the system are not geared to rapid software development.

- As the requirements change or as requirements problems are discovered, the system design or implementation has to be reworked and retested. As a consequence, a conventional waterfall or specification-based process is usually prolonged and the final software is delivered to the customer long after it was originally specified.

Fundamental characteristics to rapid software development:

1. The processes of specification, design, and implementation are interleaved. There is no detailed system specification, and design documentation is minimized or generated automatically by the programming environment used to implement the system. The user requirements document only defines the most important characteristics of the system.

2. The system is developed in a series of versions. End-users and other system stakeholders are involved in specifying and evaluating each version. They may

propose changes to the software and new requirements that should be implemented in a later version of the system.

3. System user interfaces are often developed using an interactive development system that allows the interface design to be quickly created by drawing and placing icons on the interface. The system may then generate a web-based interface for a browser or an interface for a specific platform such as Microsoft Windows.

Agile methods:

- Incremental development methods
- Increments are small
- New releases of the system are created and made available to customers every two or three weeks.
- They involve customers in the development process to get rapid feedback on changing requirements.
- They minimize documentation by using informal communications rather than formal meetings with written documents.

Some successful types of system development using Agile methods:

1. Product development where a software company is developing a small or medium-sized product for sale.

2. Custom system development within an organization, where there is a clear commitment from the customer to become involved in the development process and where there are not a lot of external rules and regulations that affect the software.

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

3.1 Agile Principals and Project Management

1. The success of customer involvement in the development process depends on having a customer who is willing and able to spend time with the development team. Frequently, the customer representatives are subject to other pressures and cannot take full part in the software development.

2. Individual team members may not have suitable personalities for the intense involvement that is typical of agile methods, and therefore not interact well with other team members.

3. Prioritizing changes can be extremely difficult, especially in systems for which there are many stakeholders. Typically, each stakeholder gives different priorities to different changes.

4. Maintaining simplicity requires extra work. Under pressure from delivery schedules, the team members may not have time to carry out desirable system simplifications.

5. Many organizations, especially large companies, have spent years changing their culture so that processes are defined and followed. It is difficult for them to move to a working model in which processes are informal and defined by development teams.

Agile project management:

- The principal responsibility of software project managers is to manage the project so that the software is delivered on time and within the planned budget for the project.

- They supervise the work of software engineers and monitor how well the software development is progressing.

The standard approach to project management is plan-driven.

Plan-driven approach to software engineering identifies separate stages in the software process with outputs associated with each stage. The outputs from one stage are used as a basis for planning the following process activity.

Managers draw up a plan for the project showing:

- What should be delivered
- When it should be delivered
- Who will work on the development of the project deliverables

A plan-based approach really requires a manager to have a stable view of everything that has to be developed and the development processes.

Like every other professional software development process, agile development has to be managed so that the best use is made of the time and resources available to the team. This requires a different approach to project management, which is adapted to incremental development and the particular strengths of agile methods.

The Scrum method is an agile method that provides a project management framework.

It is centered on a set of sprints, which are fixed time periods when a system increment is developed. Planning is based on prioritizing a backlog of work and selecting the highest- priority tasks for a sprint.

Scrum, when designed was intended for use with co-located teams where all team members could get together every day in stand-up meetings.

- Software development now involves distributed teams with team members located in different places around the world.

- Currently, there are various experiments going on to develop Scrum for distributed development environments

There are three phases in Scrum:

1. Outline-planning phase where you establish the general objectives for the project and design the software architecture.

2. Series of sprint cycles, where each cycle develops an increment of the system.

3. The project closure phase wraps up the project, completes required documentation such as system help frames and user manuals, and assesses the lessons learned from the project.

The innovative feature of Scrum is its central phase, the sprint cycles.

A Scrum sprint is a planning unit in which the work to be done is assessed, features are selected for development, and the software is implemented. At the end of a sprint, the completed functionality is delivered to stakeholders.

1. Sprints are fixed length, normally 2–4 weeks.

2. The starting point for planning is the product backlog, which is the list of work to be done on the project. During the assessment phase of the sprint, this is reviewed, and priorities and risks are assigned. The customer is closely involved in this process and can introduce new requirements or tasks at the beginning of each sprint.

3. The selection phase involves all of the project team who work with the customer to select the features and functionality to be developed during the sprint.

4. Once these are agreed, the team organizes themselves to develop the software. Short daily meetings involving all team members are held to review progress and if necessary, reprioritize work. During this stage the team is isolated from the customer and the organization.

5. At the end of the sprint, the work done is reviewed and presented to stakeholders. The next sprint cycle then begins.