Computers II Lesson 10

10.0 Project Planning

Project planning is one of the most important jobs of a software project manager.

As a manager, you have to:

- Break down work into parts
- Assign these parts to project team members
- Anticipate problems that might arise
- Prepare tentative solutions to those problems

The project plan, which is created at the start of a project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.

Project planning takes place at three stages in a project life cycle:

1. At the proposal stage, when you are bidding for a contract to develop or provide a software system. You need a plan at this stage to help you decide if you have the resources to complete the work and to work out the price that you should quote to a customer.

2. During the project startup phase, when you have to plan who will work on the project, how the project will be broken down into increments, how resources will be allocated across your company, etc. Here, you have more information than at the proposal stage, and can therefore refine the initial effort estimates that you have prepared.

3. Periodically throughout the project, when you modify your plan in light of experience gained and information from monitoring the progress of the work. You learn more about the system being implemented and capabilities of your development team. This information allows you to make more accurate estimates of how long the work will take. Furthermore, the software requirements are likely to change and this usually means that the work

breakdown has to be altered and the schedule extended. For traditional development projects, this means that the plan created during the startup phase has to be modified. However, when an agile approach is used, plans are shorter term and continually change as the software evolves.

Plan-driven or plan-based development is an approach to software engineering where the development process is planned in detail.

A project plan is created that records:

- Work to be done
- Who will do it
- The development schedule
- The work products

Managers use the plan to support project decision-making and as a way of measuring progress.

Plan-driven development is based on engineering project management techniques and can be thought of as the 'traditional' way of managing large software development projects.

This contrasts with agile development, where many decisions affecting the development are delayed and made later, as required, during the development process.

The principal argument against plan-driven development is that many early decisions have to be revised because of changes to the environment in which the software is to be developed and used.

Plans normally include the following sections:

1. Introduction - This briefly describes the objectives of the project and sets out the constraints (e.g., budget, time, etc.) that affect the management of the project.

2. Project organization - This describes the way in which the development team is organized, the people involved, and their roles in the team.

3. Risk analysis - This describes possible project risks, the likelihood of these risks arising, and the risk reduction strategies that are proposed.

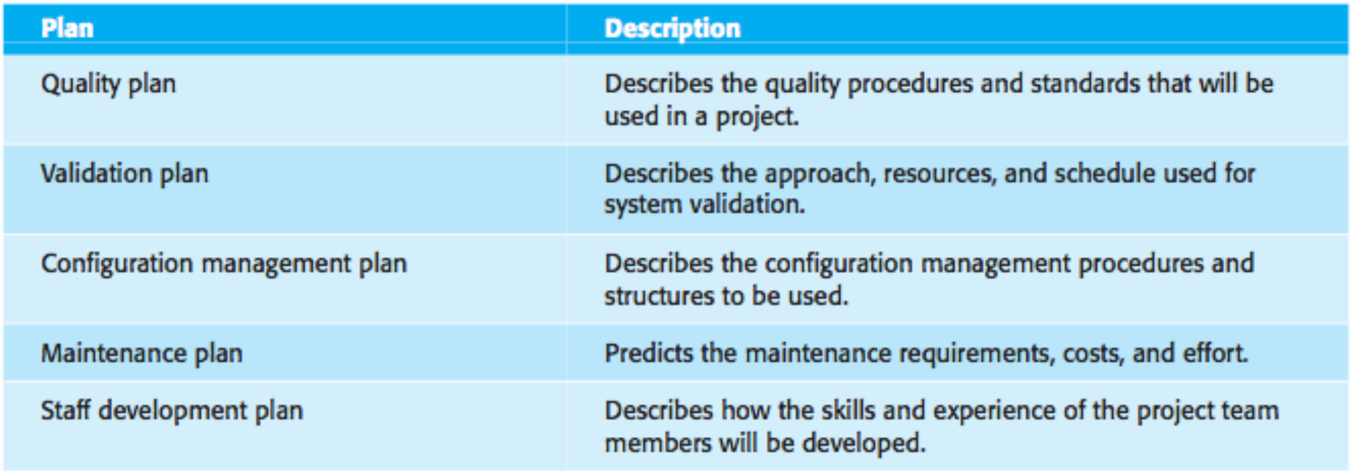
4. Hardware and software resource requirements - This specifies the hardware and support software required to carry out the development. If hardware has to be bought, estimates of the prices and the delivery schedule may be included.

5. Work breakdown - This sets out the breakdown of the project into activities and identifies the milestones and deliverables associated with each activity. Milestones are key stages in the project where progress can be assessed; deliverables are work products that are delivered to the customer.

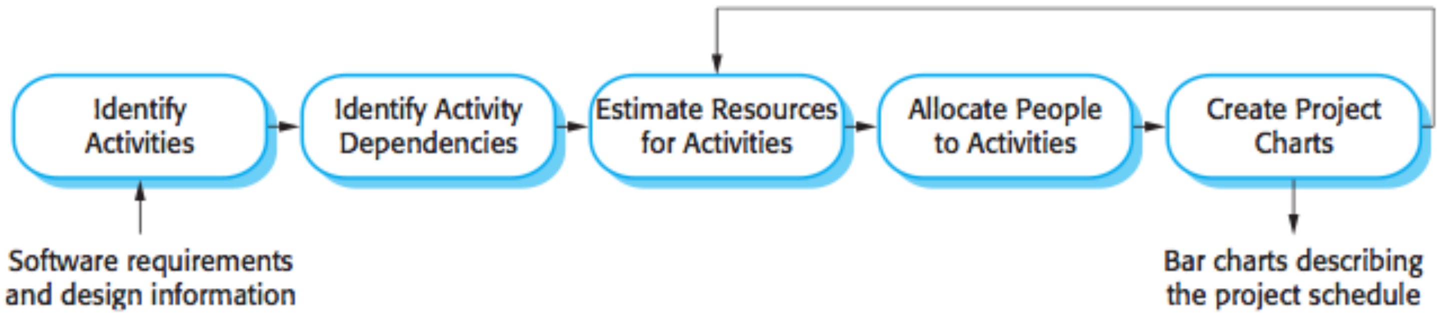
6. Project schedule - This shows the dependencies between activities, the estimated time required to reach each milestone, and the allocation of people to activities. The ways in which the schedule may be presented are discussed in the next section of the chapter.

7. Monitoring and reporting mechanisms - This defines the management reports that should be produced, when these should be produced, and the project monitoring mechanisms to be used.

Supplementary plans to support other process activities such as testing and configuration management:



Plan	Description
Quality plan	Describes the quality procedures and standards that will be used in a project.
Validation plan	Describes the approach, resources, and schedule used for system validation.
Configuration management plan	Describes the configuration management procedures and structures to be used.
Maintenance plan	Predicts the maintenance requirements, costs, and effort.
Staff development plan	Describes how the skills and experience of the project team members will be developed.


Identify Activities
Identify Activity Dependencies
Estimate Resources for Activities
Allocate People to Activities
Create Project Charts
Software requirements and design information
Bar charts describing the project schedule

10.1 Programming Language

Now that you have learned about the basis of developing software and project management I will cover some programming language definitions to refresh your memory on some important programming terms.

When a programmer sits down to write a program, which is a set of instructions for your computer he or she is using what is called a programming language.

Programming languages consist of symbols and statements that give the computer directions to perform a task.

What he writes in this language is referred to as “source code or code”.

Many of these languages exist and they all have pros and cons.

Examples are ruby on rails, golang, PHP, visual basic, java, C++

These languages however, are not what the computer understands.

Because computers are electrical devices, they only understand machine language or binary code.

Binary code is composed of only 0s and 1s.

- Each different type of CPU has its own unique machine language.
- For example, 01100001 represent the lower case letter “a”.

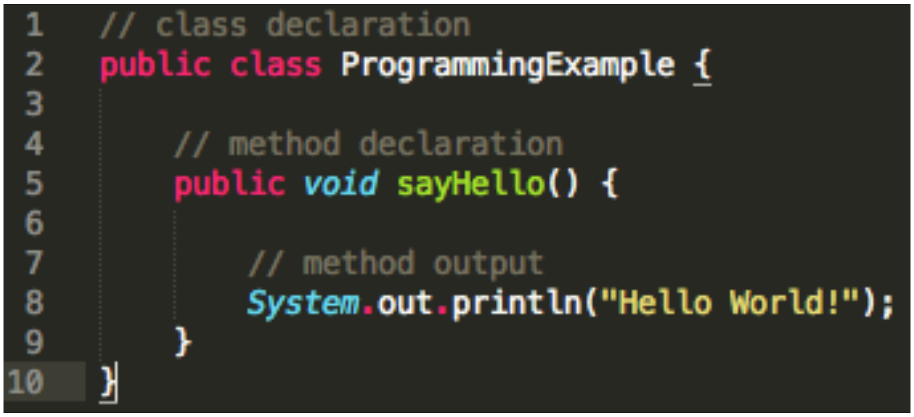
Each language has a unique set of keywords (words that it understands) and a special syntax for organizing program instructions.

Syntax - the syntax of a computer language is the set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in that language.

Programming languages, while simple compared to human languages, are more complex than the languages the computer actually understands

Object-oriented programming - A type of programming in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure.

- In this way, the data structure becomes an object that includes both data and functions.
- Programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects.



```
// class declaration
public class ProgrammingExample {

    // method declaration
    public void sayHello() {

        // method output
        System.out.println("Hello World!");
    }
}
```