UNIVERSITY OF RAJSHAHI

Rajshahi, BANGLADESH.

Course Code:

ICE-3251

Course Title :

Information System Analysis and Software Engineering

Unit Details

Software Projects and Risk Management: Project Management Concepts: People, Product and Process. Product Metrics, Process and Project Metrics, Estimation for Software Project, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management.

What is Software Project Management (SPM)?

- Project management involves the planning, monitoring, and control of the people, process, and events that occur as software evolves from a preliminary concept to full operational deployment.

Who does it?

- Everyone "manages" to some extent, but the scope of management activities varies among people involved in a software project.
- A **software engineer** manages his/her day-to-day activities, planning, monitoring, and controlling technical tasks.
- **Project managers** plan, monitor, and control the work of a team of software engineers.
- **Senior managers** coordinate the interface between the business and software professionals.

Why it is important to have SPM?

- Building computer software is a complex undertaking, particularly if it involves many people working over a relatively long time. That's why software projects need to be managed.

What are the steps in SPM?

- Understand the four Ps— **people**, **product**, **process**, and **project**.
- **People** must be organized to perform software work effectively.
- Communication with the customer and other stakeholders must occur so that **product** scope and requirements are understood.
- A **process** that is appropriate for the people and the product should be selected.
- The **project** must be planned by estimating effort and calendar time to accomplish work tasks: defining work products, establishing quality check- points, and identifying mechanisms to monitor and control work defined by the plan.

The Software Project Management Spectrum

- Effective software project management focuses on the four Ps:
    - people,
    - product,
    - process, and
    - project.

The people as Project Players

Categorization of Stakeholders (Cont…)

- The software process (and every software project) is populated by stakeholders under people who can be categorized into one of **5(FIVE)** constituencies:
  - ***Senior managers*** who define the business issues that often have a significant influence on the project.
  - ***Project (technical) managers*** who must plan, motivate, organize, and control the practitioners who do software work.

The people as Project Players

Categorization of Stakholders (Cont…)

- ***Practitioners*** who deliver the technical skills that are necessary to engineer a product or application.
- ***Customers*** who specify the requirements for the software to be engineered and other stakeholders who have a peripheral interest in the outcome.
- ***End users*** who interact with the software once it is released for production use.

A Model for Project Management Leadership

Project management is a people-intensive activity, and for this reason, competent practitioners often make poor team leaders. In an excellent book of technical leadership, Jerry Weinberg suggests an MOI model of leadership:

- ***Motivation.*** The ability to encourage (by "push or pull") technical people to produce to their best ability.
- ***Organization.*** The ability to mold existing processes (or invent new ones) that will enable the initial concept to be translated into a final product.
- ***Ideas or innovation.*** The ability to encourage people to create and feel creative even when they must work within bounds established for a particular software product or application.

Characteristics of an Effective Project Manager

- ***Problem solving.*** An effective software project manager can diagnose the technical and organizational issues that are most relevant, systematically structure a solution or properly motivate other practitioners to develop the solution, apply lessons learned from past projects to new situations, and remain flexible enough to change direction if initial attempts at problem solution are fruitless.
- ***Managerial identity.*** A good project manager must take charge of the project. S/he must have the confidence to assume control when necessary and the assurance to allow good technical people to follow their instincts

Characteristics of an Effective Project Manager?

- ***Achievement.*** A competent manager must reward initiative and accomplishment to optimize the productivity of a project team. S/he must demonstrate through his/her own actions that controlled risk taking will not be punished.
- ***Influence and team building.*** An effective project manager must be able to "read" people; s/he must be able to understand verbal and nonverbal signals and react to the needs of the people sending these signals. The manager must remain under control in high-stress situations.

The Software Team

- There are almost as many human organizational structures for software development as there are organizations that develop software.
- However, the organization of the people directly involved in a new software project is within the project manager's purview.
- The “best” team structure depends on the management style of your organization, the number of people who will populate the team and their skill levels, and the overall problem difficulty.

The Software Team (Cont….)

There are seven project factors that should be considered when planning the structure of software engineering teams:

- The difficulty of the problem to be solved.
- The size of the resultant program(s) in lines of code or function points.
- The time that the team will stay together (team lifetime).
- The degree to which the problem can be modularized.
- The required quality and reliability of the system to be built.
- The rigidity of the delivery date.
- The degree of sociability (communication) required for the project.

Organizational Paradigms for Software Engineering Team

1. A ***closed paradigm*** structures a team along a traditional hierarchy of authority. Such teams can work well when producing software that is quite similar to past efforts, but they will be less likely to be innovative when working within the closed paradigm.
2. A ***random paradigm*** structures a team loosely and depends on individual initiative of the team members. When innovation or technological breakthrough is required, teams following the random paradigm will excel. But such teams may struggle when "orderly performance" is required.

Organizational Paradigms for Software Engineering Team?

3. An ***open paradigm*** attempts to work collaboratively, with heavy communication and consensus-based decision making the trademarks of open paradigm teams. Open paradigm team structures are well suited to the solution of complex problems but may not perform as efficiently as other teams.
4. A ***synchronous paradigm*** relies on the natural compartmentalization of a problem and organizes team members to work on pieces of the problem with little active communication among themselves.

Software Scope

- The first software project management activity is the determination of ***software scope***.
- Scope is defined by answering the following questions:
  - ***Context.*** How does the software to be built fit into a larger system, product, or business context, and what constraints are imposed as a result of the context?
  - ***Information objectives.*** What customer-visible data objects are produced as output from the software? What data objects are required for input?
  - ***Function and performance.*** What function does the software perform to transform input data into output? Are any special performance characteristics to be addressed?

The Process

- The framework activities that characterize the software process are applicable to all software projects.
- The problem is to select the process model that is appropriate for the software to be engineered by your project team.

The Process

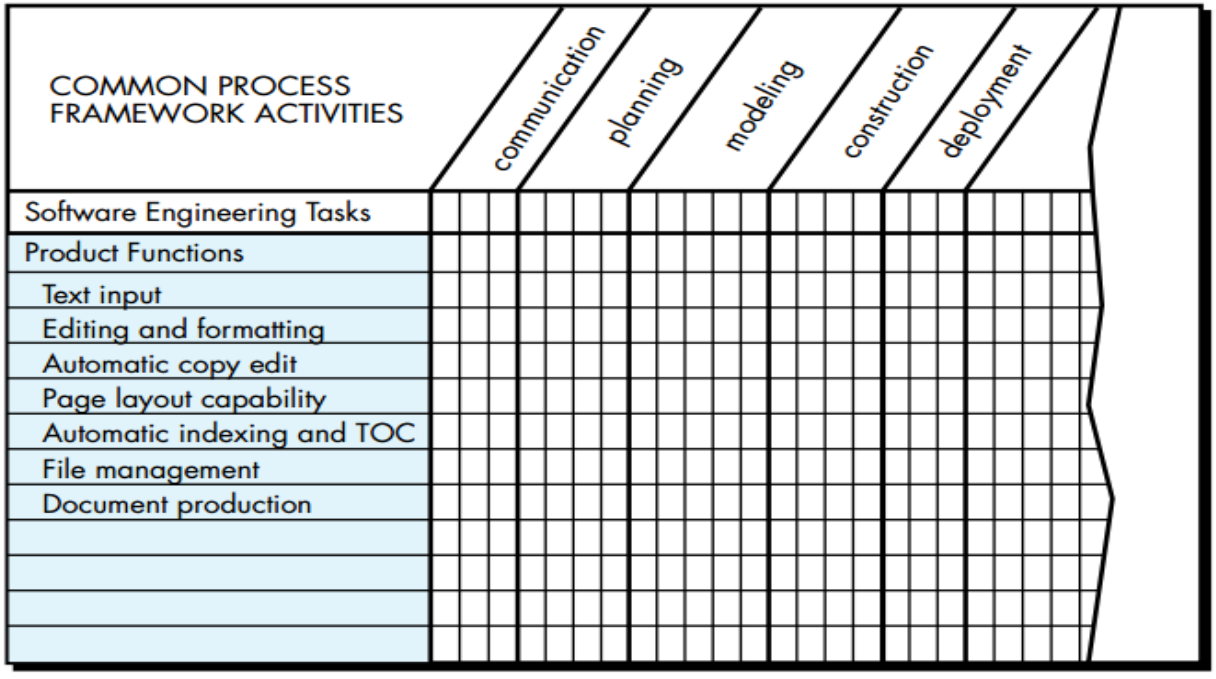
- Your team must decide which process model is most appropriate for
  - (1) the customers who have requested the product and the people who will do the work,
  - (2) the characteristics of the product itself, and
  - (3) the project environment in which the software team works.
- When a process model has been selected, the team then defines a preliminary project plan based on the set of process framework activities.
- Once the preliminary plan is established, process decomposition begins. That is, a complete plan, reflecting the work tasks required to populate the framework activities, must be created.

Melding the Product and Process

- Project planning begins with the melding of the product and the process.
- Each function to be engineered by your team must pass through the set of **framework activities** that have been defined for your software organization.
- Assume that the organization has adopted the generic **framework activities**—**communication**, **planning**, **modeling**, **construction**, and **deployment**.
- The team members who work on a product function will apply each of the framework activities to it.

Melding the Product and Process

- In essence, a matrix similar to the one shown in Figure is created.



- Each major product function (the figure notes functions for the word-processing software) is listed in the left-hand column.
- Framework activities are listed in the top row.
- The job of the project manager (and other team members) is to estimate resource requirements for each matrix cell, start and end dates for the tasks associated with each cell, and work products to be produced as a consequence of each task.

Software Process and Project Metrics

What is it?

- Software process and project metrics are quantitative measures that enable you to gain insight into the efficacy of the software process and the projects that are conducted using the process as a framework.
- Basic quality and productivity data are collected. These data are then analyzed, compared against past averages, and assessed to determine whether quality and productivity improvements have occurred.
- Metrics are also used to pinpoint problem areas so that remedies can be developed and the software process can be improved.

Software Process and Project Metrics

Who does it?

- Software metrics are analyzed and assessed by software managers. Measures are often collected by software engineers.

Why it is important?

- If you don’t measure, judgment can be based only on subjective evaluation. With measurement, trends (either good or bad) can be spotted, better estimates can be made, and true improvement can be accomplished over time.

Software Process and Project Metrics

What are the steps?

- Begin by defining a limited set of process, project, and product measures that are easy to collect.
- These measures are often normalized using either size- or function-oriented metrics.
- The result is analyzed and compared to past averages for similar projects performed within the organization.
- Trends are assessed and conclusions are generated.

Estimation for Software Project

What is it?

- A real need for software has been established; stakeholders are onboard, software engineers are ready to start, and the project is about to begin.
- But how do you proceed? Software project planning encompasses five major activities—**estimation**, **scheduling**, **risk analysis**, **quality management planning**, and **change management planning**.
- Consider only **estimation**—attempt to determine how much money, effort, resources, and time it will take to build a specific software-based system or product.

Estimation for Software Project

Who does it?

- Software project managers—using information solicited from project stakeholders and software metrics data collected from past projects.

Why it is important?

- Would you build a house without knowing how much you were about to spend, the tasks that you need to perform, and the time line for the work to be conducted?
- Of course not, and since most computer-based systems and products cost considerably more to build than a large house, it would seem.

Estimation for Software Project

What are the steps?

- Estimation begins with a description of the scope of the product.
- The problem is then decomposed into a set of smaller problems, and each of these is estimated using historical data and experience as guides.
- Problem complexity and risk are considered before a final estimate is made.