Project Management Concepts

What is Project?

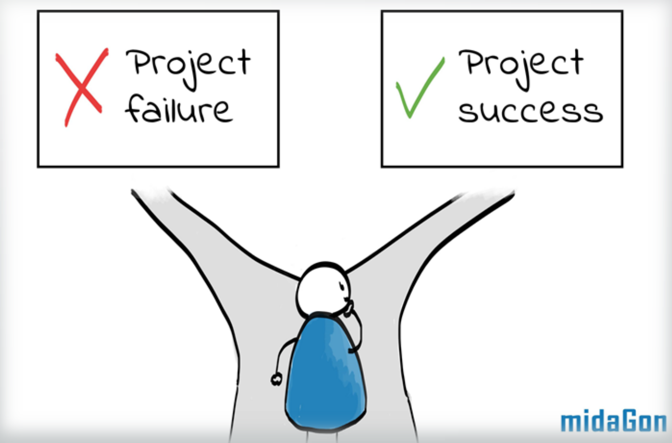
- A project is well-defined task, which is a collection of several operations done in order to achieve a goal
- A Project can be characterized as:
  - Every project may has a unique and distinct goal.
  - Project is not routine activity or day-to-day operations.
  - Project comes with a start time and end time.
  - Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
  - Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

Software Project

A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

Project without Project Management Will Fail

Project with Project Management may or may not fail



Need for Project Management

- Software development is a kind of all new stream in world business and there's very little experience in building software products.
- Most software products are tailor made to fit client's requirements
- The underlying technology changes and advances so frequently and rapidly that experience of one product may not be applied to the other one.
- All such business and environmental constraints bring risk in software development.
- Hence it is essential to manage software projects efficiently.

The Management Spectrum

- Effective software project management focuses on these items
  - **The people**
    - Deals with the cultivation of motivated, highly skilled people
    - Consists of the stakeholders, the team leaders, and the software team
  - **The product**
    - Product objectives and scope should be established before a project can be planned
  - **The process**
    - The software process provides the framework from which a comprehensive plan for software development can be established
  - **The project**
    - Planning and controlling a software project is done for one primary reason…it is the only known way to manage complexity
    - In a 1998 survey, 26% of software projects failed outright, 46% experienced cost and schedule overruns

PEOPLE

The People: The Stakeholders

Five categories of stakeholders

- **Senior managers -** Define business issues that often have significant influence on the project
- **Project (technical) managers –** plan, motivate, organize, and control the practitioners who do the work
- **Practitioners –** deliver the technical skills that are necessary to engineer a product or application
- **Customers –** specify the requirements for the software to be engineered and other stakeholders who have a peripheral interest in the outcome
- **End users –** interact with the software once it is released for production use

The People: Team Leader

- Competent practitioners often fail to make good team leaders as they just don't have the right people skills.
- Qualities to look for in a team leader
  - ○ **Motivation** – the ability to encourage 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
- Team leaders should use a problem-solving management style
  - ○ Concentrate on understanding the problem to be solved
  - ○ Manage the flow of ideas
  - ○ Let everyone on the team know, by words and actions, that quality counts and that it will not be compromised

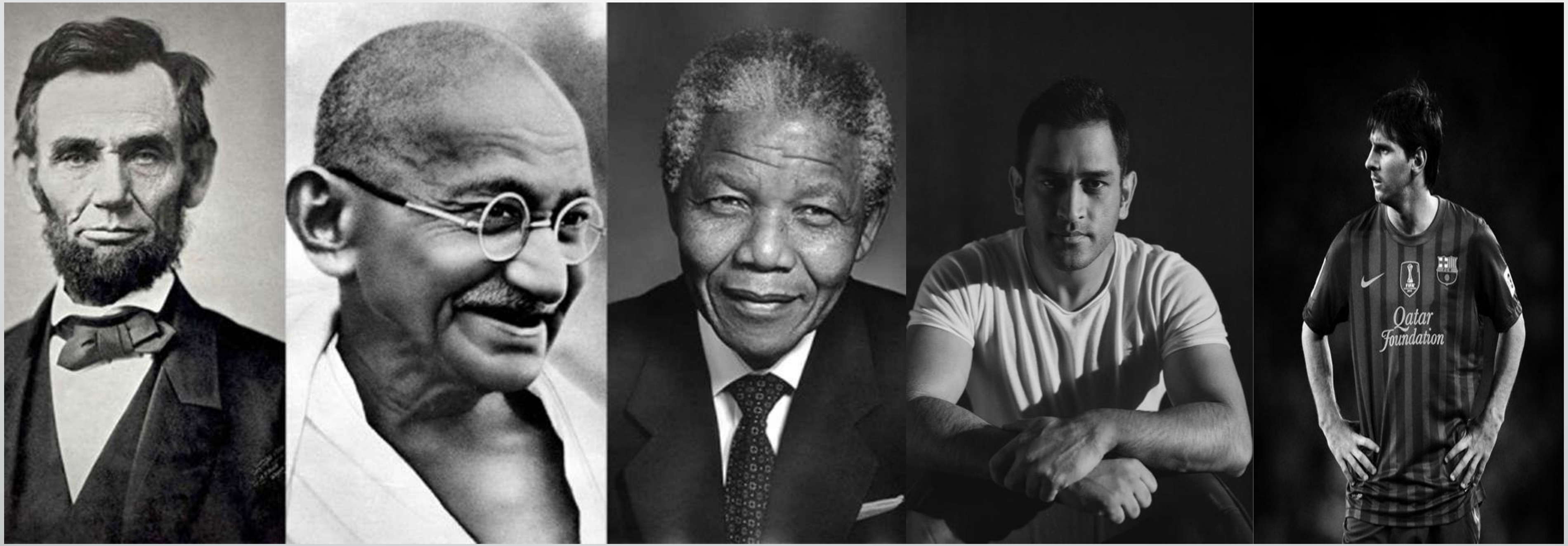
The People: Team Leader (Contd.,)

- Set of useful leadership traits
  - ○ **Problem solving –** diagnose, structure a solution, apply lessons learned, remain flexible.
  - ○ **Managerial identity –** take charge of the project, have confidence to assume control, have assurance to allow good people to do their jobs.
  - ○ **Achievement –** reward initiative, demonstrate that controlled risk taking will not be punished.
  - ○ **Influence and team building –** be able to "read" people, understand verbal and nonverbal signals, be able to react to signals, remain under control in high-stress situations.

Activity

Think about any of your favourite leader that you come across ?

Some suggested Leaders….



Activity

Does your leader have the good qualities of a leader?

The People: The Software Team

- Seven project factors to consider when structuring a software development team
  - ○ The **difficulty** of the problem to be solved
  - ○ The **size** of the resultant program(s) in source lines of code
  - ○ The **time** that the team will stay together
  - ○ 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

The People: The Software Team (Contd.,)

- Four organizational paradigms for software development teams
  - **Closed paradigm –** traditional hierarchy of authority; works well when producing software similar to past efforts; members are less likely to be innovative
  - **Random paradigm –** depends on individual initiative of team members; works well for projects requiring innovation or technological breakthrough; members may struggle when orderly performance is required
  - **Open paradigm –** hybrid of the closed and random paradigm; works well for solving complex problems; requires collaboration, communication, and consensus among members
  - **Synchronous paradigm –** organizes team members based on the natural pieces of the problem; members have little communication outside of their subgroups

Activity

Suggest one example for each category of Organization paradigm .....

The People: The Software Team (Contd.,)

- Five factors that cause team **toxity** (i.e., a toxic team environment)
  - A **frenzied** work atmosphere
  - **High frustration** that causes friction among team members
  - A **fragmented** or **poorly coordinated** software process
  - An **unclear** definition of roles on the software team
  - Continuous and repeated **exposure to failure**

The People: The Software Team (Contd.,)

- How to avoid these problems
  - ○ Give the team **access to all information** required to do the job
  - ○ **Do not modify** major goals and objectives, once they are defined, unless absolutely necessary
  - ○ Give the team as much **responsibility** for decision making as possible
  - ○ Let the team **recommend** its own process model – Let the team establish its own mechanisms for accountability (i.e., reviews)
  - ○ Establish team-based techniques for **feedback** and **problem solving**

The People: Coordination and Communication Issues

- Key characteristics of modern software make projects fail
  - Scale, Uncertainty, Interoperability

- To better ensure success
  - Establish effective methods for coordinating the people who do the work
  - Establish methods of formal and information communication among team members

Group Dynamics

- Based on studies published by B. Tuckman in 1965
- Describes a four-stage model
  - Forming
  - Storming
  - Norming
  - Performing

Group Dynamics Model

- Forming
  - Group members rely on safe, patterned behavior and look to the group leader for guidance and direction
  - Impressions are gathered and similarities and differences are noted
  - Serious topics and feelings are avoided
  - To grow, members must relinquish the comfort of non-threatening topics and risk the possibility of conflict

Group Dynamics Model (Contd.,)

- Storming
  - ○ As group members organize for the tasks, conflict inevitably results in their personal relations and cliques start to form
  - ○ Individuals have to bend and mold their feelings to fit the group
  - ○ Fear of exposure or fear of failure causes an increased desire for structural clarification and commitment
  - ○ Conflicts arise over leadership, structure, power, and authority
  - ○ Member behavior may have wide swings based on emerging issues of competition and hostilities
  - ○ Some members remain silent while others attempt to dominate

Group Dynamics Model (Contd.,)

- Norming
  - Members engage in active acknowledgement of all members' contributions community building, and solving of group issues
  - Members are willing to change their preconceived ideas or opinions based on facts presented by the group
  - Leadership is shared, active listening occurs, and cliques dissolve
  - Members began to identify with one another, which leads to a level of trust in their personal relations and contributes to cohesion
  - Members begin to experience a sense of group belonging

Group Dynamics Model (Contd.,)

- Performing
  - ○ The capacity, range, and depth of personal relations in the group expand to true interdependence
  - ○ Members can work independently, in subgroups, or altogether with equal ability and success
  - ○ The group is most productive, members become self-assuring, and the need for group approval is past
  - ○ Genuine problem solving can occur leading towards optimal solutions

PRODUCT

The Product

- The scope of the software development must be established and bounded
  - **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 functions does the software perform to transform input data into output? Are there any special performance characteristics to be addressed?
- Software project scope must be unambiguous and understandable at both the managerial and technical levels

The Product (Contd.,)

- Problem decomposition
  - ○ Also referred to as partitioning or problem elaboration
  - ○ Sits at the core of software requirements analysis

- Two major areas of problem decomposition
  - ○ The functionality that must be delivered
  - ○ The process that will be used to deliver it

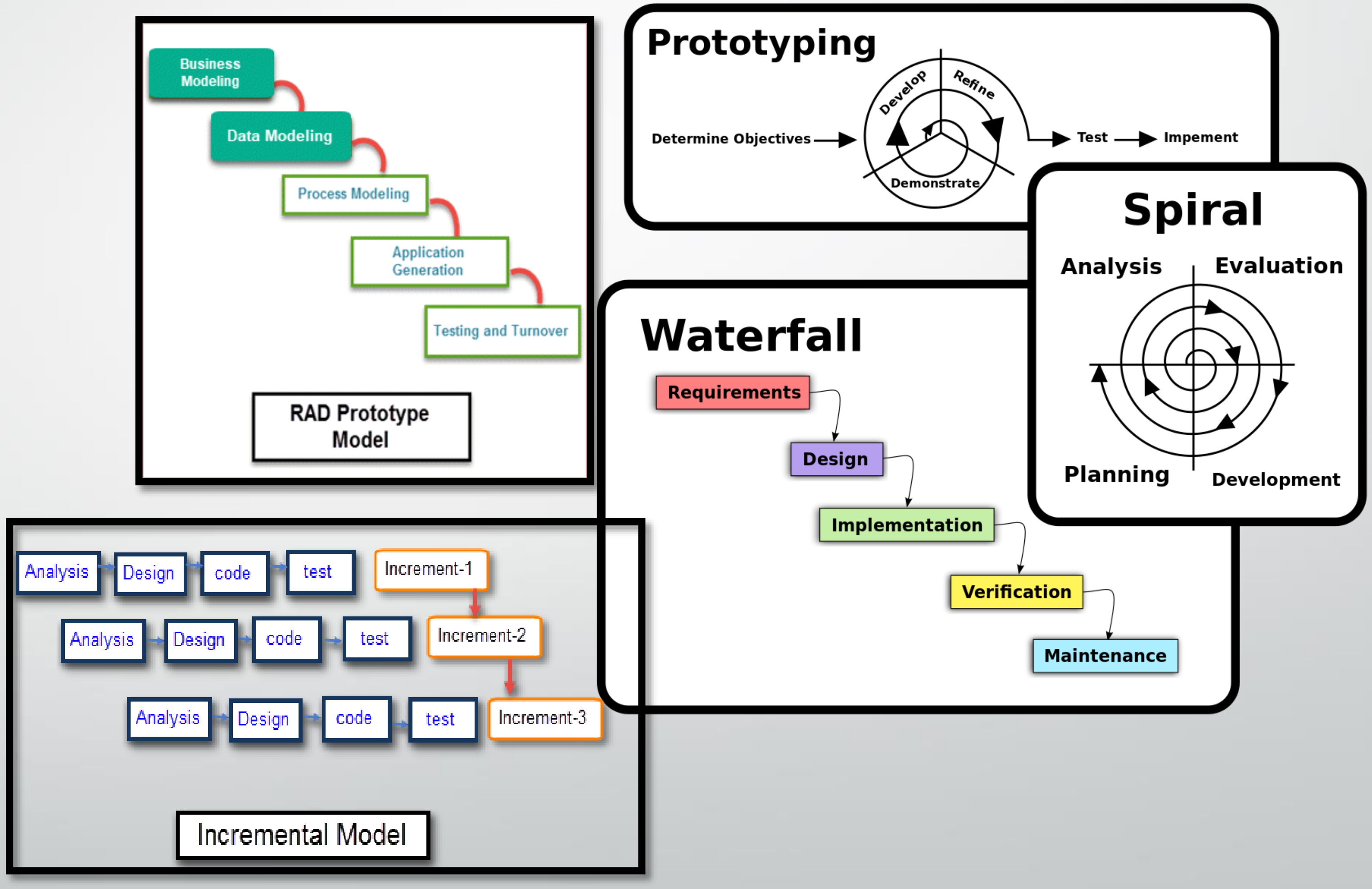
PROCESS

The Process

- Getting Started
  - The project manager must decide which **process model is most appropriate** based on
    - The customers who have requested the product and the people who will do the work
    - The characteristics of the product itself
    - The project environment in which the software team works
  - Once a process model is selected, a preliminary **project plan** is established based on the process framework activities
  - Process **decomposition** then begins
  - The result is a complete plan reflecting the work tasks required to populate the framework activities
- Project planning begins as a melding of the product and the process based on the various framework activities

Selection of Process Model

- Waterfall Model
- Incremental Model
- RAD Model
- Prototyping Model
- Spiral Model



Activity

- Identify a Problem
- Suggest a suitable Process model
  - Also List why the other models are not appropriate.

The Project

Software Project Versus Others types of Project

The following characteristics will make the software project different from others.

- Invisibility

  Software is not immediately visible as like the construction of bridges.

- Complexity

  We cant measure the complexity of software project until we actually work on it.

- Conformity

  Software project are based on logical work, while other are based on physical work. Software developers have to conform to the requirements of human clients.

- Flexibility

  The Ease in which the software can be changed is usually seen as its strength.

Some ways of categorizing software projects

- Information systems versus embedded systems
  - The difference is that in the former case the system interfaces with the organization, whereas in the latter case the system interfaces with a machine!
  - A stock control system would be an information system that controls when the organization reorders stock.
  - An embedded, or process control, system might control the air conditioning equipment in a building.
  - Some systems may have elements of both so that the stock control system might also control an automated warehouse.

Some ways of categorizing software projects

Information systems versus embedded systems

- Would an operating system on a computer be an information system or an embedded system?

Some ways of categorizing software projects

Information systems versus embedded systems

- Would an operating system on a computer be an information system or an embedded system?
  - An **operating system** be categorized as an **embedded system** and this is due to the fact that **operating system** is a software that manages a **computer** hardware and so, it means that the hardware as a machine is supported by the **operating system**.

Some ways of categorizing software projects

- Objectives versus products
  - Projects may be distinguished by whether their aim is to produce a product or to meet certain objectives.
  - A project might be to create a product the details of which have been specified by the client. The client has the responsibility for justifying the product.
  - On the other hand, the project might be required to meet certain objectives. There might be several ways of achieving these objectives in contrast to the constraints of the product-driven project.

Some ways of categorizing software projects

- Objectives versus products
  - Example of this is where a new information system is implemented to improve some service to users inside or outside an organization.
  - The subject of an agreement would be the level of service rather than the characteristics of a particular information system.

Problems with software projects

- A survey of managers published by Thayer, Pyster and Wood identified the following commonly experienced problems:
  - Poor estimates and plans;
  - Lack of quality standards and measures;
  - Lack of guidance about making organizational decisions;
  - Lack of techniques to make progress visible;
  - Poor role definition - who does what?
  - Incorrect success criteria.

Managerial Perspective

Problems with software projects

- **Below is a list of the problems identified by a number of students on a degree course in Computing and Information Systems who had just completed a year's industrial placement:**
  - ○ Inadequate specification of work;
  - ○ Lack of knowledge of application area;
  - ○ Lack of commitment - especially w hen a project is tied to one person who then moves;
  - ○ Lack of up-to-date documentation;
  - ○ Lack of communication between users and technicians;
  - ○ Lack of communication leading to duplication of work;
  - ○ Changing software environment;
  - ○ Deadline pressure;
  - ○ Lack of quality control;
  - ○ Remote management;

Staff Perspective

The Project: A Common Sense Approach

- Start on the right foot
  - Understand the problem; set realistic objectives and expectations; form a good team
- Maintain momentum
  - Provide incentives to reduce turnover of people; emphasize quality in every task; have senior management stay out of the team's way
- Track progress
  - Track the completion of work products; collect software process and project measures; assess progress against expected averages
- Make smart decisions
  - Keep it simple; use COTS or existing software before writing new code; follow standard approaches; identify and avoid risks; always allocate more time than you think you need to do complex or risky tasks
- Conduct a post mortem analysis
  - Track lessons learned for each project; compare planned and actual schedules; collect and analyze software project metrics; get feedback from teams members and customers; record findings in written form

W5HH Principle

- Boehm suggests an approach that addresses project objectives, milestones and schedules, responsibilities, management and technical approaches, and required resources.
- He calls it the W5HH Principle, after a series of questions that lead to a definition of key project characteristics and the resultant project plan
- **Why , What , When , Who , Where , How , How .**

W5HH Principle

W5HH	The Question	What It Means
Why?	Why is the system being developed?	This focuses a team on the business reasons for developing the software.
What?	What will be done?	This is the guiding principle in determining the tasks that need to be completed.
When?	When will it be completed?	This includes important milestones and the timeline for the project.
Who?	Who is responsible for each function?	This is where you determine which team member takes on which responsibilities. You may also identify external stakeholders with a claim in the project.
Where?	Where are they organizationally located?	This step gives you time to determine what other stakeholders have a role in the project and where they are found.

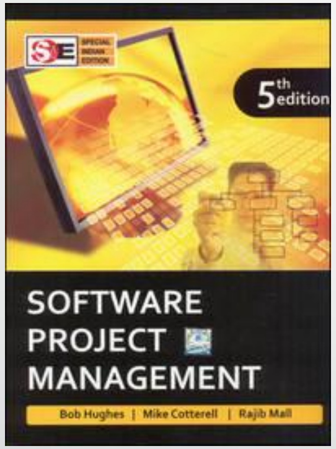
W5HH Principle

W5HH	The Question	What It Means
How?	How will the job be done technically and managerially?	In this step, a strategy for developing the software and managing the project is concluded upon.
How Much?	How much of each resource is needed?	The goal of this step is to figure out the amount of resources necessary to complete the project.

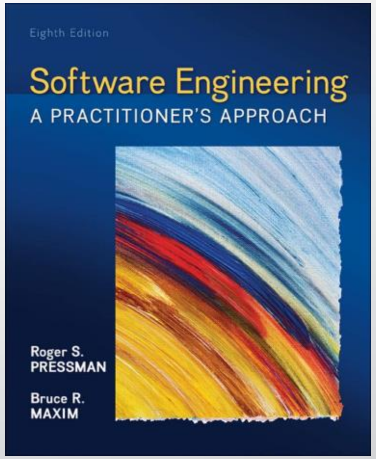
Activity

- **For the Identified Problem**
  - ○ Visualize and prepare W5HH Principle

Reference


SE SPECIAL INDIAN EDITION
5th edition
SOFTWARE PROJECT MANAGEMENT
Bob Hughes | Mike Cotterell | Rajib Mall

tutorialspoint
SIMPLYEASYLEARNING


Eighth Edition
Software Engineering
A PRACTITIONER'S APPROACH
Roger S. PRESSMAN
Bruce R. MAXIM