CS615 - Software Project Management
Reference Short Notes for Mid Term Papers
Muhammad Faisal Dar
MIT 4th Semester
faisalgrw123@gmail.com

The 80:20 Rule: (page 11)

The 80:20, rule was originated by Vilfredo Pareto, an Italian economist who studies the distribution of wealth in a variety of countries around 1900. He discovered a common phenomenon: about 80% of the wealth in most countries was controlled by a consistent minority -- about 20% of the people. Pareto called this a "predictable imbalance." His observation eventually became known as either the "80:20 rule" or "Pareto's Principle."

A project has five phases: (page 13)

$\Rightarrow$ Initiation
$\Rightarrow$ Planning
$\Rightarrow$ Executing
$\Rightarrow$ Controlling
$\Rightarrow$ Closing

Project Life Cycle includes the following Phases: (page 16)

A. Concept Phase
B. Requirements Phase
C. Design Phase
D. Development Phase
E. Operation Phase

(page 29)
In managing competing project requirements Project managers often talk of a triple constraint:

– Project scope
– Time and
– Cost

The role of the Leader in project management is one of great responsibility. It's the project manager's job to direct and supervise the project from beginning to end. Here are some other roles:

(a) Leadership
(b) Communications
(c) Problem Solving
(d) Negotiating
(e) Influencing the Organization
(f) Mentoring
(g) Process and technical expertise

PM's nine Knowledge Areas

1. Project Integration Management
2. Project Scope Management
3. Project Time Management
4. Project Cost Management
5. Project Quality Management
6. Project Human Resource Management
7. Project Communications Management
8. Project Risk management
9. Project Procurement Management

Difference between Leaders and Managers:

Leadership is path finding
Management is path following
Management is about doing things right
Leadership is about doing the right things

Supporting detail

Supporting detail for organizational planning varies by application area and project size. Information frequently supplied as supporting detail.

Software engineering practices: **(page58)**

> ➢ Span of control
> ➢ Authority
> ➢ Responsibility
> ➢ Delegation
> ➢ Chain of command
> ➢ Accountability
> ➢ Line authority
> ➢ Staff departments

The Software Crisis: **(page60)**
Some of the reasons to which you can attribute the software crisis include:

– Software developers used-multiple programming languages.
– Most of the requirements were complex with regard to the existing capabilities,
– Software developers poorly mapped requirements to the actual product.
– Software maintenance was costly.
– Hardware developed at a faster rate than software.

Status reports: **(page62)**

Status reports should be required from every member of the development team, without exception. The reports should be submitted periodically, usually weekly or bi-weekly, and should contain at least the following three sections.

1. Activities during the report period
2. Planned activities for the next report period
3. Problems

A typical software project includes the following phases: **(page67)**

> ❖ Software requirement analysis phase
> ❖ Software Design Phase
> ❖ Software Planning Phase
> ❖ Software construction phase
> ❖ Software testing phase
> ❖ Software acceptance and maintenance phase

The primary software project management areas that you need to concentrate on include:

- o Managing Resources
- o Managing Cost
- o Managing Risk
- o Managing Schedule
- o Managing the project plan
- o Managing quality

As a **project manager**, you perform **two primary activities to manage risks for software project**:

- o Risk Analysis
  - Risk identification
  - Risk quantification

- o Risk management

The project management phases can be broadly categorized as follows;

- i. Project initiation
- ii. Project closedown
- iii. Project planning, control, and tracking
- iv. Product implementation

Product Implementation

Product implementation activities involve defining processes related to the implementation of the software product at the customer site. Some of the tasks that you perform for product implementation are mentioned below:

- I. Implementation plan creation
- II. Support plan creation
- III. Training plan creation
- IV. User acceptance plan

Project Closedown   (page80)

The final activity for a project manager is project closedown. For most software projects, the project closedown activities take place in the post-implementation phase. The tasks that you perform in project closedown are mentioned below:

   i.   Prepare closedown report
  ii.   Identify learning
 iii.   Identify reusable software components
 iv.   Create reference material

Problems in Software Projects   (page85)

You can classify the problems that affect software projects into the following four categories:

   I. People-related problems
  II. Process-related problems
 III. Product-related problems
 IV. Technology-related problems

⇒ **People-related problems   (page85)**

- Low motivation
- Problem employees
- Unproductive work environment
- Inefficient project management style
- Lack of stakeholder interest
- Ineffective project sponsorship by management

⇒ **Process- related Problems   (page86)**

- Unrealistic schedule
- Insufficient identification
- Unsuitable life cycle model selection
- Abandoning quality under pressure of deadlines
- Unstructured and hurried software development

⇒ **Product-related Problems   (page87)**

- Product scope changed toward the end of the project life cycle
- Research-oriented software development
- III-defined scope
- Fuzzy users

⇒ **Technology-related problems**

- ➢ Overestimated savings from reusable components and new tools and methods
- ➢ Switching tools in mid way
- ➢ Integrating different software products in cross-platform implementation

Requirements analysis

Requirements analysis is a software engineering task that bridges the gap between system level requirements engineering and software design. Software requirements analysis may be divided into five areas of effort:

1. Problem recognition,
2. Evaluation and synthesis,
3. Modeling
4. Specification, and
5. Review

There is also Requirements Analysis includes:

Evaluation and Synthesis, Models, Specification & Concerns for Review

Requirements Elicitation for Software

(1) Initiating the Process
(2) Facilitated Application Specification Techniques
(3) Quality Function Deployment
(4) Use Cases
(5) Analysis Principles
(6) Software Prototyping

The Software Requirements Specification

– Quality
– Timeliness and
– Completeness of SW product

The **bibliography** contains references to all documents that relate to the software. These include other software engineering documentation, technical references, vendor literature, and; standards.

The **appendix** contains information that supplements the specifications. Tabular data, detailed description of algorithms, charts, graphs and other material, are presented as appendixes.

Attributes of SW Design:

Software design is actually a multi step process that focuses on four distinct attributes of a program:

- o Data structure,
- o Software architecture,
- o Interface representations, and procedural (algorithmic) detail.

Modularity

Modularity (in both program and data) and the concept of abstraction enable the designer to simplify and reuse software components.

Standard design approaches

- ➤ Exception handling,
- ➤ Localization
- ➤ Portability
- ➤ Reuse
- ➤ Input/output
- ➤ Memory management,
- ➤ Performance

Software Configuration Management

Software configuration management (SCM) takes care of changes in a software process. SCM identifies controls, audits, and reports modifications that occur during software development. SCM helps maintain the integrity of configurable items produced during software development. SCM is an integral part of Software Quality Assurance (SQA). SCM involves assessing the impact of the changes made during SQA activities and making decisions based on cost and benefit analysis.

Software Configuration Management Activities   **(page 110)**

SCM is used to establish and maintain integrity of software items and ensure that they can be traced easily. Using SCM, you can define a library structure for storage and retrieval of software items. SCM needs to be performed at all phases in the SDLC of a software project. The various SCM activities are:

1. Identifying Objects
2. Controlling Versions
3. Controlling Changes
4. Auditing
5. Communicating Changes

Software Configuration management important topics: **(page 115)**

 Read these topics from handout

> Controlling Changes
> Communicating Changes
> Software Configuration Management Vs Software Maintenance
> Baselines vs. Interim Versions

Quality Control   **(page 119)**

Quality control is a series of review activities, such as:

- Inspections
- Reviews and
- Tests, used throughout the SDLC of the software product

Quality Factors   **(page 120)**

McCall (MCC77) and his colleagues proposed some software quality factors based on three most important aspects of a software product:

1. Product operation
2. Product revision and
3. Product transition

1. Product Operation Factors (page120)

Correctness	Accuracy of the program and the extent to which it fulfills design specifications
Reliability	Extent to which the program is secure and its ability to recover quickly from failure.
Efficiency	Performance of the program and its ability to perform tasks within a time frame
Integrity	Ability of the program to take care of security and the extent to which it can prevent unauthorized.
Usability	Ease with which a user can learn, operates, and uses the program.

2. Product Revision Factors (page121)

Maintainability	Ease with which a program is debugged.
Flexibility	Ease with which a program is modified.
Testability	Ease with which a program is tested.

3. Product Transition Factors (page121)

Portability	Efficiency with which a program runs on different platforms or operating systems
Reusability	Extent to which the program can be used in more than one program or system
Interoperability	Effort needed to transfer a program to another system
Configurability	Ability of the program to be installed at more than one location with different features at each location
Expandability	Ability of the program to support an increase in data and users

Software Quality Assurance Activities (page122)

SQA is the process of evaluating the quality of a product and enforcing adherence to software product standards and procedures. It is an umbrella activity that ensures conformance to standards and procedures throughout the SDLC of a software product. There are a large number of tasks involved in SQA activities. These include:

- o Formulating a quality management plan
- o Applying software engineering techniques
- o Conducting formal technical reviews
- o Applying a multi-tiered testing strategy
- o Enforcing process adherence
- o Controlling change
- o Measuring impact of change
- o Performing SQA audits
- o Keeping records and reporting

Software Review **(page124)**

Software review is an effective way of filtering errors in a software product. Typically, an error found after the product release costs 50 times as much to correct as one detected during the design phase. Software review is used as a filter at various points of software development. Reviews conducted at each of these phases, analysis, design, coding, and testing reveal areas of improvement in the product. Some of the most common software review techniques, practiced across software organizations include:

a) Inspection
b) Walkthrough
c) Formal technical reviews

Software Process **(page128)**

When you build a product or system, it's important to go through a series of predictable steps – a road map that helps you create a timely, high-quality result, The road map that you follow is called a 'software process'.

Software Process Vs Software Engineering **(page129)**

Software engineering is performed by creative, knowledgeable people who should work within a defined and mature software process that is appropriate for the products they build, and the demands of their marketplace. A software process defines the approach that is taken as software is engineered. But software engineering also encompasses technologies that populate the process-technical methods and automated tools.

Software engineering layers **(page130)**

Process
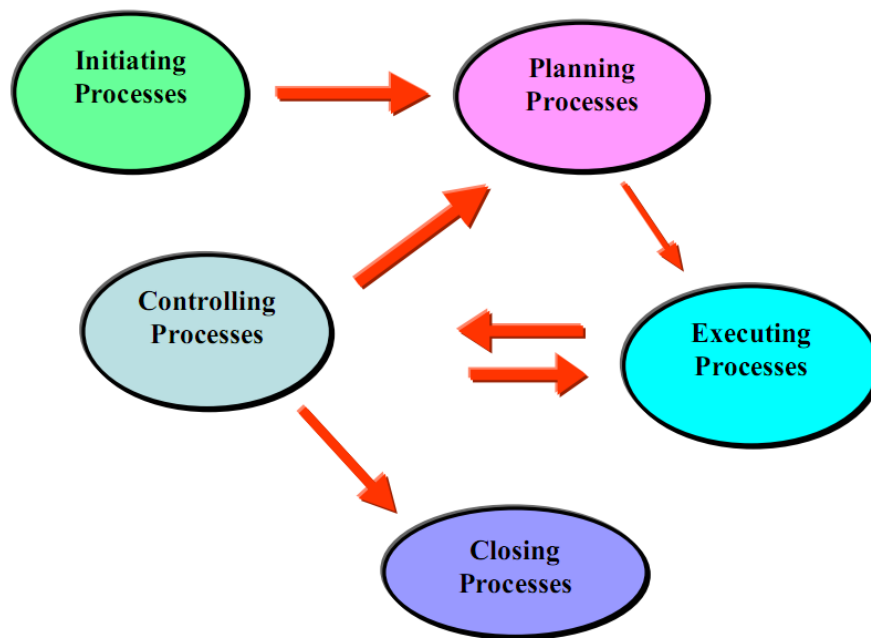
A quality focus

Tools

Metho
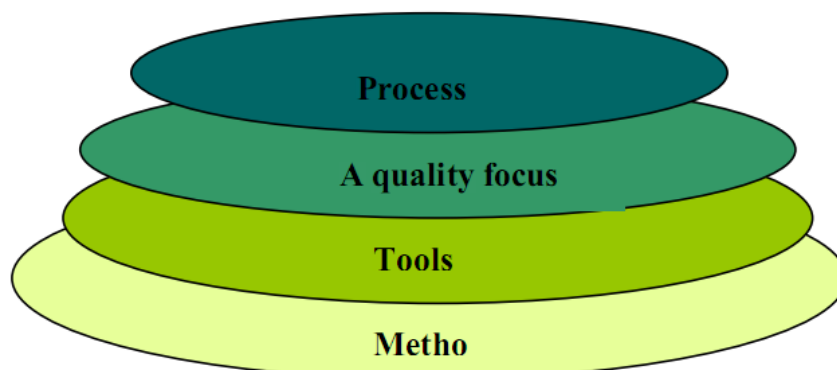
PM Process Groups **(page131)**

Project management processes can be organized into five groups of one or more processes each:

1. Initiating processes
2. Planning processes
3. Executing processes
4. Controlling processes
5. Closing processes

PM Phase interactions **(page132)**

Initiating Process Inputs **(page134)**

➢ Product Description
➢ Strategic Plan
➢ Selection Criteria
➢ Historical Information

Initiating Process Outputs   (page134)

- Project Charter
- Project Manager assignments
- Constraints
- Assumptions

Initiating Process Tools and Techniques   (page134)

- Project selection methods
- Expert judgment

The tasks performed for project initiation are: (page134)

- Requirement gathering
- Scope determination
- Resource allocation
- Initial project plan

SMARTER is abbreviated as: (page136)

Specific
Measurable
Acceptable
Realistic
Time frame
Extending
Rewarding

Planning Process Tasks   (page138)

1. Scope Planning
2. Scope Definition
3. Activity Definition
4. Activity Sequencing
5. Activity Duration Estimating
6. Resource Planning
7. Cost Estimating

8. Cost Budgeting Risk Planning
9. Schedule Development
10. Quality Planning
11. Communications Planning
12. Organization Planning
13. Staff Acquisition
14. Procurement Planning
15. Project Plan Development

Executing Process Tasks   **(page139)**

1. Project Plan Execution
 2. Quality Assurance
3. Team Development
 4. Information Distribution
5. Solicitation
6. Source Selection
7. Contract Administration

Controlling Process Tasks   **(page140)**

1. Integrated Change Control
2. Scope Verification
3. Scope Change Control
4. Schedule Control
5. Cost Control
6. Quality Control
7. Performance Reporting
8. Risk Response Control

Management Aspect   **(page107)**

– Major design styles
• Object, structured, data-structured design

– Foundational design concepts
• Information hiding, abstraction, encapsulation, Inheritance, basic algorithms & data structures, ...