17MBAPS401B SOFTWARE PROJECT MANAGEMENT 4 0 0 4

Scope:

Software project management course provides a platform for students to understand the concepts of project management. It also emphasizes on testing and acquisition process in software projects.

Objectives:

Learn to acquire a set of skills for planning and implementing software projects. Learn to acquire a set of skills for managing cost and time. Learn to access the control risk in project management.

Unit I

Introduction – Software Project Management – An overview of Project planning – Stepwise planning – Activities and characteristics of SPM – overview of COCOMO model, PERT/CPM. Rayleigh curve – Project Organization – Risk Management – Project Finance – Procurement Management – Project Scheduling.

Unit II

Software project Management – Resources planning and estimation. Different methods in brief – Different methods in brief - Function point analysis in some details - Use of CASE tools – Introduction to MS projects – Design and Development – Schedule – Resource Allocation – Progress Review – Project implementation and execution of MS office.

Unit III

Testing – Overview of Test plan- Generation of Test cases, Test data - Types of Testing - Quality concepts – ISO, CMM,- Production / Implementation – User acceptance tests,- parallel runs . – Maintenance – Types - Adaptive, Corrective, Preventive version control and configuration management – documentation methods.

Unit IV

Acquisition process - Hardware, software, Network, Infrastructure - Requirement planning-sizing - selection Methodology including Benchmarking - Documents involved IT HRM – Selection – Retention – Training – Career path planning – IT operations – Scheduling – Performance Evaluation.

Unit V

Risk Management – Nature and types of risk – managing risk – evaluating risk – A software management process frame works- Life cycle phases – software maintenance and configuration management – Maintenance characteristics – Maintenance task – Maintenance side effects.

Suggested Readings:

Text Book:

1. Roger.S.Pressman (2012) Software project management (4th Edition). New Delhi. Himalaya Publications.

Reference:

1. Donald J. Reifer (2010) Software project management (7th Edition) New Delhi. Wiley India Pvt.Ltd

2. Kelker (2012) Software project management (2nd edition) New Delhi. Prentice hall India.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

LECTURE PLAN
DEPARTMENT OF COMMERCE

STAFF NAME: V.VIVEK
SUBJECT NAME: SOFTWARE PROJECT MANAGEMENT
SUB.CODE:17MBAPS401B
SEMESTER: IV CLASS: II MBA

S.No	Lecture Duration Period	Topics to be Covered	Support Material/Page Nos
		UNIT-I	
1	1	Introduction, software project management	T1 – 8
2	1	An overview of project planning, stepwise planning	R1:18-34
3	1	Activities and characteristics of SPM	R1:4
4	1	Overview of COCOMO model, PERT/CPM	R1:96-101,112
5	1	Rayleigh curve, project organization	T1:24
6	1	Risk management, project finance	R1:138
7	1	Procurement management, project scheduling	R1:154
8	1	Recapitulation and Discussion of Important Questions	
	Total No of Hours Planned For Unit 1=8		
		UNIT-II	
1	1	SPM resource planning and estimation	T1-46,R1:79
2	1	Different methods in FPA in some details	R1:85
3	1	Use of case tools	R1:86
4	2	MS Project, Design, Development and Schedule	R3:222-226

5	1	Resource Allocation, Progress review	R1:154
6	1	Project Implementation and execution of MS office	R1:155
7	1	Recapitulation and Discussion of Important Questions	
8	1	Recapitulation and Discussion of Important Questions	
	Total No of Hours Planned For Unit II=8		
		UNIT-III	
1	1	Testing and overview of test plan	R3:346-348
2	1	Generation of test cases, test data, types of testing	R3-349-351
3	1	Quality concepts- ISO, CMM, Production/Implementation	R3:332-336
4	1	User acceptance test, Parallel runs and maintenance	R3:337-339
5	1	Types: Adaptive, Corrective	R3:439-445
6	1	Preventive version control	R5:161-163
7	2	Configuration management and documentation method	R3:464-471
8	1	Recapitulation and Discussion of Important Questions	
	Total No of Hours Planned For Unit III=8		
		UNIT-IV	
1	1	Acquisition process- Hardware, software	R4:154
2	1	Network, Infrastructure-Requirement Planning	R4:155
3	1	Sizing, Selection methodology including Bench marking	R4:156
4	1	Documents involved IT HRM	W1
5	1	Selection, Retention , Training	W1
6	1	Career path planning, IT operations	R6-87-88
7	1	Scheduling, performance evaluation	R6-88-90
8	2	Recapitulation and Discussion of Important Questions	

	Total No of Hours Planned For Unit IV=9		
		UNIT-V	
1	1	Risk management, Nature and Types of risk	R5:203-207
2	1	Managing Risk, Evaluating Risk	R5-210-216
3	1	Software management process frame work, LC Phases	R4:198-199
4	1	Software maintenance and configuration	R5:193-197
5	2	Maintanence: Characteristics, tasks and side effects	R3:197-202
8	1	Recapitulation and Discussion of Important Questions	
9	1	Discussion of Previous year ESE questions	
10	1	Discussion of Previous year ESE questions	
11	1	Discussion of Previous year ESE questions	
	Total No of Hours Planned for unit V=11		
		Total Planned Hours	**44**

TEXT BOOK

T1: Roger.S.Pressman,(2007) software project management (3rd edition). New Delhi, Himalaya Publishers

Reference:

R1:Donald.J.Reifer(2006) software project management (6th edition),New Delhi. Wiley India Pvt.Ltd

R2:Kelkar(2012)software project management(2nd edition) New Delhi, Prentice hall India

R3: K.K.Aggarwal, Yogesh Singh(2008)Software engineering (3rd edition) New Delhi, New Age International Publication.

R4:Hughes and Cotterell(2005)Software Project Management(3rd edition)New Delhi, TATA Mc-Graw Hill.

R5: B.B.Agarwal, S.Dhall & S.P.Tayal(2011) Software project Management(1st edition) New Delhi, University Science Press.

R6:NIIT(2004)Basics of Software project Management(1st edition) NIIT University Publication.

Unit – I

Introduction - Software Project Management - An overview of Project planning - Stepwise planning- Overview of COCOMO Model, PERT/CPM, Rayleigh Curve - Project Organization -Risk Management - Project Finance - Procurement Management - Project Scheduling

INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT

Management:

Management is the process of designing and maintaining an environment in which individuals, working together in groups, efficiently accomplish selected aims.

- Harold Koontz

Software Project Management

What is 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.

Need of Software Project management:

Software is said to be an intangible product. Software development is a kind of all new streams 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 most important is that 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 image above shows triple constraints for software projects. It is an essential part of software organization to deliver quality product, keeping the cost within client's budget constrain and deliver the project as per scheduled. There are several factors, both internal and external, which may impact this triple constrain triangle. Any of three factor can severely impact the other two. Therefore, software project management is essential to incorporate user requirements along with budget and time constraints.

Who is Software Project Manager?

A software project manager is a person who undertakes the responsibility of executing the software project. Software project manager is thoroughly aware of all the phases of SDLC that the software would go through. Project manager may never directly involve in producing the end product but he controls and manages the activities involved in production.

****

OVERVIEW OF PROJECT PLANNING

Project Planning: Meaning

Planning is the most difficult process in project management. This chapter describes a framework of basic steps in project planning. Many different techniques can be used but this chapter tells the overview of the steps and activities in each step of project planning

Software project planning is task, which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production.

Stepwise Project Planning

Step 0: Select project

Step 1: Identify project scope and objectives

Step 2: Identify project infrastructure

Step 3: Analyze project characteristics

Step 4: Identify project products and activities

Step 5: Estimate effort for each activity.

Step 6: Identify activity risks.

Step 7: Allocate resources

Step 8 Review / Publicize pl\an

Step 9 &10: Execute plan / lower level of planning

Each step of project planning has different activities to perform. Following are the description of each step with its activities.

Step 0: Select project

This is called step 0 because in a way of project planning, it is outside the main project planning process. Feasibility study suggests us that the project is worthwhile or not.

Step 1: Identify project scope and objectives

The activities in this step ensure that all parties to the project agree on the objectivesand are committed to the success of the project.

Step 1.1: Identify objectives and practical measures of the effectiveness in meeting those objectives

Step 1.2: Establish project authority

Step 1.3: Stakeholders analysis – Identify all stakeholders in the project and their interest.

Step 1.4: Modify objectives in the light of stakeholder analysis.

Step 1.5: Establish method of communication

Step 2: Identify project infrastructure

Projects are rarely carried out in a vacuum. There is usually some kind of infrastructure into which the project must fit. Where the project manager are new to the organization , they must find out the precise nature of this infrastructure.

Step 2.1: Identify relationship between the project and strategic planning

Step 2.2: Identify installation standards and procedures.

Step 2.3: Identify project team organization.

Step 3: Analyze project characteristics

The general purpose of this part of planning operation is to ensure that the appropriate methods are used for the project.

Step 3.1: Distinguish the project as either objective- product driven

Step 3.2: Analyze other project characteristics (including quality –based ones)

Step 3.3: Identify high level project risks

Step 3.4: Take into account user requirement concerning implementation.

Step 3.5: Select development methodology and life cycle approach.

Step 3.6: Review overall resources estimates

Step 4: Identify project products and activities

The more detailed planning of the individual activities now takes place. The longer term planning is broad and in outline, while the more immediate tasks are planned in some detail.

Step 4.1: Identify and describes project products (ordeliverables)

Step 4.2: Document generic product flows

Step 4.3: Record product instance

Step 4.4: produce ideal activity network

Step 4.5: Modify the ideal to take into account need for stages and checkpoints.

Step 5: Estimate effort for each activity

Step 5.1: Carry out bottom-up estimates

Step 5.2: Revise plan to create controllable activities.

Step 6: Identify activity risks

Step 6.1: Identify and quantify activity based risks

Step 6.2: Plan risk reduction and contingency measures where appropriate

Step 6.3: Adjust overall plans and estimates to take account of the risks

Step 7: Allocate resources

Step 7.1: Identify and allocate resources

Step 7.2: Revise plans and estimates to take into account resource constraints

Step 8: Review / Publicize plan

Step 8.1: Review quality aspects of the project plan.

Step 8.2: Document plans and obtain agreement.

Step 9 & 10: Execute plan / lower level of planning

Once the project is underway, plans will need to be drawn up in greater detail for each activity as it becomes due. Detailed and lower level of planning of the later stages will need to be delayed because more information will be available nearer the start of the stage. Project planning is an iterative process. As the time approaches for the particular activities to be carried out they should be re-planned in more detail.

OVERVIEW OF COCOMO MODEL

The COCOMO cost estimation model is used by thousands of software project managers, and is based on a study of hundreds of software projects. Unlike other cost estimation models, COCOMO is an open model, so all of the details are published, including:

The underlying cost estimation equations

- Every assumption made in the model (e.g. "the project will enjoy good management")
- Every definition (e.g. the precise definition of the Product Design phase of a project)
- The costs included in an estimate are explicitly stated (e.g. project managers are included, secretaries aren't)

Because COCOMO is well defined, and because it doesn't rely upon proprietary estimation algorithms, Costar offers these advantages to its users:

- COCOMO estimates are more objective and repeatable than estimates made by methods relying on proprietary models
- COCOMO can be calibrated to reflect your software development environment, and to produce more accurate estimates

Costar is a faithful implementation of the COCOMO model that is easy to use on small projects, and yet powerful enough to plan and control large projects.

Typically, you'll start with only a rough description of the software system that you'll be developing, and you'll use Costar to give you early estimates about the proper schedule and staffing levels. As you refine your knowledge of the problem, and as you design more of the system, you can use Costar to produce more and more refined estimates.

Costar allows you to define a software structure to meet your needs. Your initial estimate might be made on the basis of a system containing 3,000 lines of code. Your second estimate might be more refined so that you now understand that your system will consist of two subsystems (and you'll have a more accurate idea about how many lines of code will be in each of the subsystems). Your next estimate will continue the process -- you can use Costar to define the components of each subsystem. Costar permits you to continue this process until you arrive at the level of detail that suits your needs.

Introduction to the COCOMO Model

The most fundamental calculation in the COCOMO model is the use of the Effort Equation to estimate the number of Person-Months required to develop a project. Most of the other COCOMO results, including the estimates for Requirements and Maintenance, are derived from this quantity.

Source Lines of Code

The COCOMO calculations are based on your estimates of a project's size in Source Lines of Code (SLOC). SLOC is defined such that:

- Only Source lines that are DELIVERED as part of the product are included -- test drivers and other support software is excluded
- SOURCE lines are created by the project staff -- code created by applications generators is excluded
- One SLOC is one logical line of code
- Declarations are counted as SLOC
- Comments are not counted as SLOC

The original COCOMO 81 model was defined in terms of Delivered Source Instructions, which are very similar to SLOC. The major difference between DSI and SLOC is that a single Source

Line of Code may be several physical lines. For example, an "if-then-else" statement would be counted as one SLOC, but might be counted as several DSI.

The Scale Drivers

In the COCOMO II model, some of the most important factors contributing to a project's duration and cost are the Scale Drivers. You set each Scale Driver to describe your project; these Scale Drivers determine the exponent used in the Effort Equation.

The 5 Scale Drivers are:

- Precedentedness
- Development Flexibility
- Architecture / Risk Resolution
- Team Cohesion
- Process Maturity

Note that the Scale Drivers have replaced the Development Mode of COCOMO 81. The first two Scale Drivers, Precedentedness and Development Flexibility actually describe much the same influences that the original Development Mode did.

PROJECT EVALUATION AND REVIEW TECHNIQUE (PERT)

Project Evaluation and Review Techniques is commonly abbreviated to PERT. PERT is a method of analyzing the tasks involved in completing a given project, especially the time needed to complete each task, and to identify the minimum time needed to complete the total project. It incorporates uncertainty by making it possible to schedule a project while not knowing precisely the details and durations of all the activities. It is more of an event-oriented technique rather than start- and completion-oriented, and is used more in projects where time is the major factor rather than cost. It is applied to very large-scale, one-time, complex, non-routine infrastructure and Research and Development projects.

Program Evaluation Review Technique (PERT) offers a management tool, which relies "on arrow and node diagrams of *activities* and *events*: arrows represent the *activities* or work necessary to reach the *events* or nodes that indicate each completed phase of the total project."

PERT and CPM are complementary tools, because "CPM employs one time estimate and one cost estimate for each activity; PERT may utilize three time estimates (optimistic, expected, and pessimistic) and no costs for each activity. Although these are distinct differences, the term PERT is applied increasingly to all critical path scheduling."

History

PERT was developed primarily to simplify the planning and scheduling of large and complex projects. It was developed for the U.S. Navy Special Projects Office in 1957 to support the U.S. Navy's Polaris nuclear submarine project. It found applications all over industry. An early example was it was used for the 1968 Winter Olympics in Grenoble which applied PERT from 1965 until the opening of the 1968 Games. This project model was the first of its kind, a revival for scientific management, founded by Frederick Taylor (Taylorism) and later refined by Henry Ford (Fordism). DuPont's critical path method was invented at roughly the same time as PERT.

Initially PERT stood for *Program Evaluation Research Task,* but by 1959 was already renamed.[2] It had been made public in 1958 in two publications of the U.S. Department of the Navy, entitled *Program Evaluation Research Task, Summary Report, Phase 1.* and *Phase 2.* In a 1959 article in *The American Statistician* the main Willard Fazar, Head of the Program Evaluation Branch, Special Projects Office, U.S. Navy, gave a detailed description of the main concepts of the PERT. He explained:

Through an electronic computer, the PERT technique processes data representing the major, finite accomplishments (events) essential to achieve end-objectives; the inter-dependence of those events; and estimates of time and range of time necessary to complete each activity between two successive events. Such time expectations include estimates of "most likely time", "optimistic time", and "pessimistic time" for each activity.

The technique is a management control tool that sizes up the outlook for meeting objectives on time; highlights danger signals requiring management decisions; reveals and defines both methodicalness and slack in the flow plan or the network of sequential activities that must be performed to meet objectives; compares current expectations with scheduled completion

dates and computes the probability for meeting scheduled dates; and simulates the effects of options for decision — before decision.The concept of PERT was developed by an operations research team staffed with representatives from the Operations Research Department of Booz, Allen and Hamilton; the Evaluation Office of the Lockheed Missile Systems Division; and the Program Evaluation Branch, Special Projects Office, of the Department of the Navy.

***PERT Guide for management use*, June 1963**

Ten years after the introduction of PERT in 1958 the American librarian Maribeth Brennan published a selected bibliography with about 150 publications on PERT and CPM, which had been published between 1958 and 1968. The origin and development was summarized as follows:

PERT originated in 1958 with the... Polaris missile design and construction scheduling. Since that time, it has been used extensively not only by the aerospace industry but also in many situations where management desires to achieve an objective or complete a task within a scheduled time and cost expenditure; it came into popularity when the algorithm for calculating a maximum value path was conceived. PERT and CPM may be calculated manually or with a computer, but usually they require major computer support for detailed projects. A number of colleges and universities now offer instructional courses in both.

For the subdivision of work units in PERT another tool was developed: the Work Breakdown Structure. The Work Breakdown Structure provides "a framework for complete networking, the Work Breakdown Structure was formally introduced as the first item of analysis in carrying out basic PERT/COST."

Advantages

- PERT chart explicitly defines and makes visible dependencies (precedence relationships) between the work breakdown structure (commonly WBS) elements.
- PERT facilitates identification of the critical path and makes this visible.
- PERT facilitates identification of early start, late start, and slack for each activity.
- PERT provides for potentially reduced project duration due to better understanding of dependencies leading to improved overlapping of activities and tasks where feasible.

- The large amount of project data can be organized and presented in diagram for use in decision making.
- PERT can provide a probability of completing before a given time.

Disadvantages

- There can be potentially hundreds or thousands of activities and individual dependency relationships.
- PERT is not easily scalable for smaller projects.
- The network charts tend to be large and unwieldy requiring several pages to print and requiring specially sized paper.
- The lack of a timeframe on most PERT/CPM charts makes it harder to show status although colours can help (e.g., specific colour for completed nodes).

CRITICAL PATH METHOD (CPM)

The critical path method (CPM) is a step-by-step project management technique for process planning that defines critical and non-critical tasks with the goal of preventing time-frame problems and process bottlenecks. The CPM is ideally suited to projects consisting of numerous activities that interact in a complex manner.

History

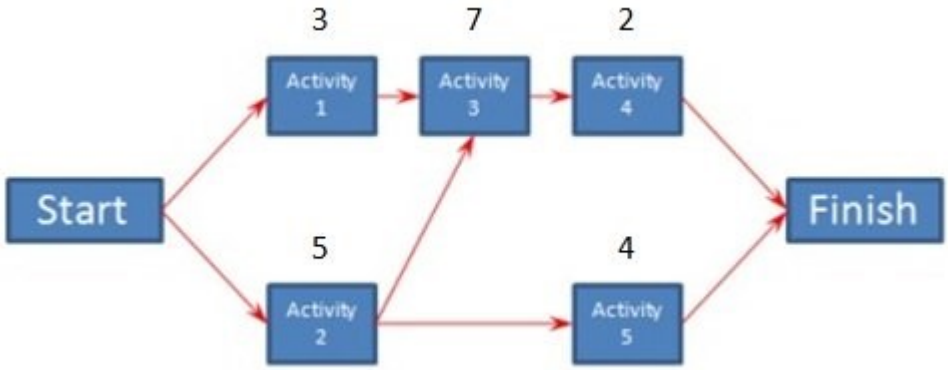
The critical path method (CPM) is a project modeling technique developed in the late 1950s by Morgan R. Walker of DuPont and James E. Kelley Jr. of Remington Rand. Kelley and Walker related their memories of the development of CPM in 1989. Kelley attributed the term "critical path" to the developers of the Program Evaluation and Review Technique which was developed at about the same time by Booz Allen Hamilton and the U.S. Navy. The precursors of what came to be known as Critical Path were developed and put into practice by DuPont between 1940 and 1943 and contributed to the success of the Manhattan Project.

Critical Path Method (CPM)

CPM is commonly used with all forms of projects, including construction, aerospace and defense, software development, research projects, product development, engineering, and plant maintenance, among others. Any project with interdependent activities can apply this method of

mathematical analysis. The first time CPM was used for major skyscraper development was in 1966 while constructing the former World Trade Center Twin Towers in NYC. Although the original CPM program and approach is no longer used, the term is generally applied to any approach used to analyze a project network logic diagram.

The critical path is the sequence of activities with the longest duration. A delay in any of these activities will result in a delay for the whole project. Below are some *critical path examples* to help you understand the key elements.



Using the Critical Path Method (CPM), the duration of each activity is listed above each node in the diagram. For each path, add the duration of each node to determine it's total duration. The critical path is the one with the longest duration.

PUTNAM, NORDEN, RAYLEIGH CURVES (OR) RAYLEIGH CURVES

Introduction

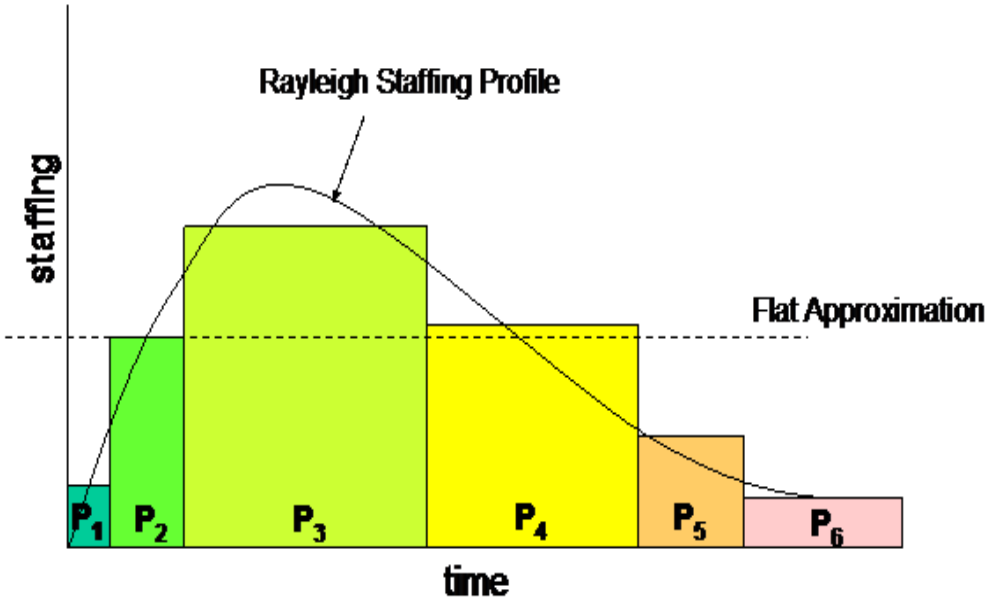
In the 1960s and 1970s, people actively sought ways to mathematically predict the time and cost of a project as a function of the requirements and other parameters of the problem. The most advanced fruit of these efforts was Barry Boehm's COnstructive COst Model. While the formulae derived in those efforts are not (today) widely used, there is still value in understanding the relationships they capture.

This paper is a brief introduction to (and commentary on) Putnam, Norden, Rayleigh (PNR) curves and what they tell us about project staffing.

Staffing and the Rayleigh Curve

1. It is very natural for a project manager to want to:
2. Figure out how many people will be needed to complete a project.
3. Assign them to it.
4. Divide the number of estimated staff months by the head-count to get an expected completion time.

Rayleigh Staffing Profile Curve



We can attempt to force a flat staffing profile on such a project, but it will result in considerable waste:

- In the early stages, there will be people standing around with nothing to do, because the requirements and architecture development process is more gated by time than by labor.
- In the mid stages, work may be delayed by not having enough people available to perform all of the defined tasks.

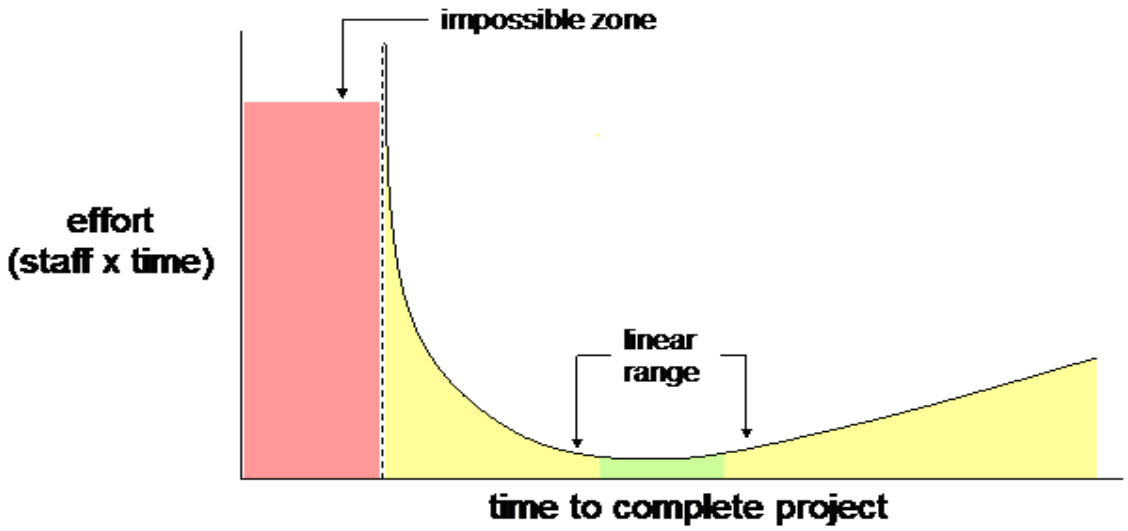
- In the late stages, over-staffing will not bring the project to completion any sooner, because the collection of feedback and shaking out of final problems is (again) more gated by time than available labor.

If we want to accomplish a project efficiently, we much understand the distinct phases, along with their respective skill and activity requirements. The notion that we can prepare an estimate in staff months and predict completion time by choosing a staffing level and dividing that into the total project size is a canard.

Time/Effort and the PNR Curve

Lawrence Putnam (of RADC) applied Norden and Rayleigh's work to the effort to quantitatively predict the work associated with software projects. After studying numerous projects, he concluded that the effort required to deliver a project varied inversely as the fourth power of the time allotted to its completion.

PNR Effort/Time Curve



The clear message here is that every project has an optimal staffing level, and (correspondingly) an optimal time in which it will be completed. Going significantly above or below the optimal staffing level will reduce work efficiency and there may be a point beyond which adding people actually delays the project.

This curve breaks down into four zones:

- **An impossible zone**

  The project cannot be accomplished in less time than this, no matter how many people are applied to the problem. This situation is classically summarized as: "Nine women cannot have a baby in one month".

- **An "Haste makes waste" zone**

  Adding people does accelerate delivery, but not in proportion to the added effort. Each additional person added to the project lowers our productivity (they have to be trained, more time goes into communication and coordination, more misunderstandings). This is a very inefficient way to operate.

- **A linear range**

  This is the range of efficient staffing, and within this range it is possible to trade man-power for time, or vice versa.

- **An under-staffed/over-staffed zone**

  This curve does not yield completion time as a function of staffing, but merely shows the relationship between staffing level and completion times. What we can clearly see on the right of the curve is that productivity is dropping. Why might this be?

  - ✓ If the project is critically under-staffed, productivity will suffer because there aren't enough people to deal with the problems.
  - ✓ If the project is greatly over-staffed communications overhead will reduce efficiency and misunderstandings will create problems and result in wasted work.

PROJECT ORGANIZATION

Define Project Organization

The Project Organization defines the human infrastructure of the project. This task is designed to define the project organization chart, the roles, and the relationships of the project team. The organizational structure clearly identifies roles and responsibilities of each position, augmenting the existing role definitions where necessary to cover all of the responsibilities. The Project Organization technique that is used in this step provides a standard set of roles and responsibilities which can be customized for a particular project. This should cover all personnel resources required, both full and part time.

If not selected in an earlier activity, it is now time to recruit the members of the Project Board. Project Board members are drawn from those in management who are senior enough to be able to commit all necessary resources to the project. Recruit project team members. It will also be necessary to determine outsourcing requirements that will drive the project plan by identifying types of external contractors required.

Identify clients/users/business partners (stakeholders) to be associated with the project and determine the level of participation required and available. Identify any stakeholders who are to be consulted and informed of any status and organizational change regarding the project. Identify any additional technical or business specialists required to support the project. This may include such areas as training, human resources, help desk, learning center, and quality assurance. The roles and responsibilities of these resources should be clearly defined along with the level of commitment over the duration of the project. For example, the whole-product solution, including the training and the new technology, will need to live on beyond deployment. This means that someone will need to "own" those solution components and the technology.

The person/function who is knowledgeable about the technology and responsible for its welfare after deployment should be tasked with staying current and watching the marketplace for significant changes in the technology area. This responsibility usually falls to a senior person (identified earlier) and someone who has been involved in the rollout effort.

Merits and Limitations of the Project Organization

Merits:

(i) Concentrated attention on project work:

In a project organisation, there is full and concentrated attention of the project manager on project work; as the project manager has no work other than attending to project management. He has full powers to co-ordinate and control project activities. In fact, during continuance of the project, functional managers renounce their authority over their project-team personnel, in favour of the project manager.

(ii) Advantages of team specialization:

The project team formed for purposes of undertaking project work consists of specialists drawn from many functional areas. This phenomenon makes available to the project organisation, the advantages of team specialisation.

(iii) Ability to cope with environmental influences:

Due to the leadership of the project manager coupled with specialised knowledge of project team members, the project organisation is in a better position to cope with environmental challenges. In fact, one of the reasons for creating a project structure is to successfully combat environmental forces.

(iv) Timely completion of the project:

The project organisation ensures a timely completion of projects; without disturbing the normal functioning of the whole organisation.

Limitations:

(i) Accentuated problems of co-ordination:

In a project organisation, there are increased problems of co-ordination; because of the diverse viewpoints of team specialists. As a matter of fact, specialists have a tendency to over-emphasize on their specialised viewpoints vis-a-vis the manner of project designing and implementation.

This tendency of specialists creates a serious headache for the project manager; who, all the time, may be found busy in reconciling conflicting viewpoints of specialists getting little time for attention towards project progress.

(ii) Unclearly defined relationship:

Usually, in a project organisation, the relationships between the project manager and functional specialists are not very clearly defined. This situation may lead to tension between them; resulting in poor human relations, in the project organisation. Ultimately, the project work efficiency may be considerably reduced.

(iii) Feeling of insecurity among personnel:

Usually, there is a feeling of uncertainty in the minds of the project team personnel as to where they will seek shelter; after a particular project (on which they were engaged) is over. This feeling of uncertainty about assignment creates feeling of insecurity among personnel; and then they tend to unduly stretch the existing project work-causing delays in timely completion of the project.

(iv) Duplication of efforts:

A project organisation suffers from the limitation of duplication of efforts, involved in the completion of project activities. When e.g. in a project organisation more than one or two projects is/are undertaken; it is quite likely that the same types of activities might be duplicated, during the completion of various projects. This phenomenon ultimately tells upon the overall organisational efficiency and profitability.

PROJECT FINANCE

What is 'Project Finance'?

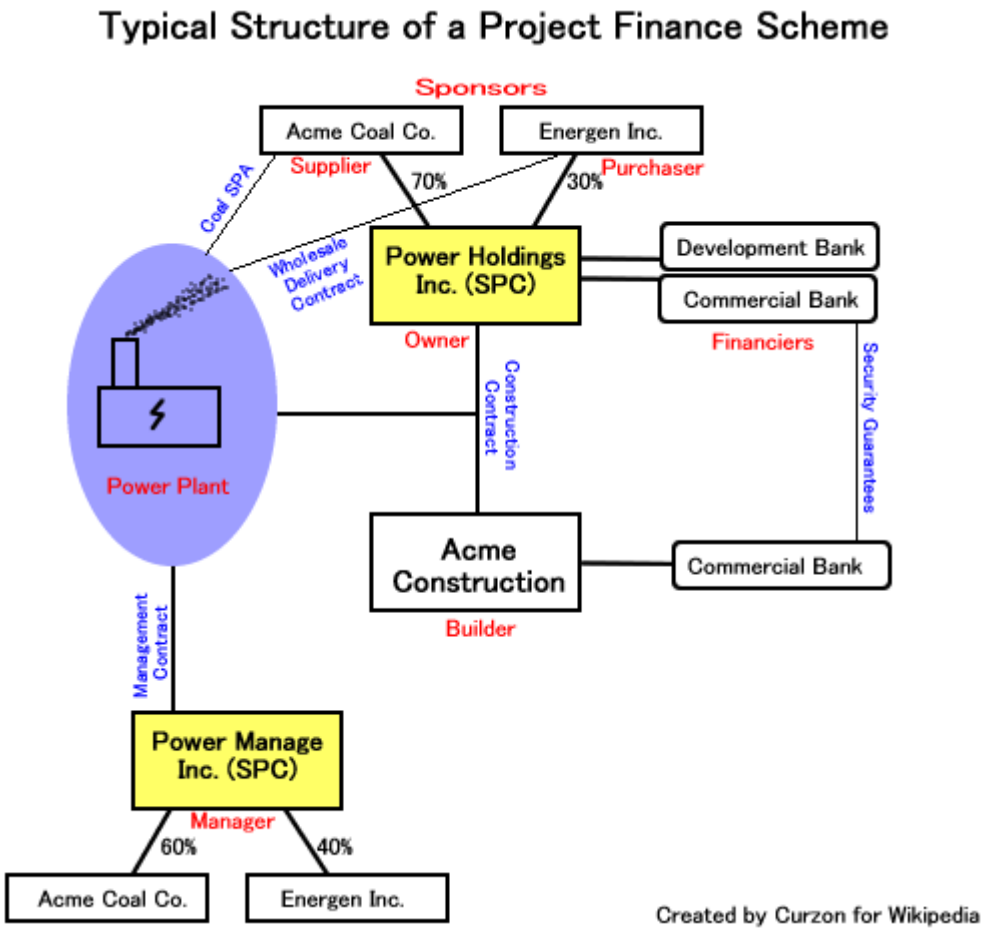
Project finance is the financing of long-term infrastructure, industrial projects and public services based upon a non-recourse or limited recourse financial structure, in which project debt and equity used to finance the project are paid back from the cash flow generated by the project. Project financing is a loan structure that relies primarily on the project's cash flow for repayment, with the project's assets, rights and interests held as secondary security or collateral. Project finance is especially attractive to the private sector because companies can fund major projects off balance sheet.

Parties to Project Finance

There are several parties in a project financing depending on the type and the scale of a project. The most usual parties to a project financing are;

1. Sponsor (typically also an Equity Investor)
2. Lenders (including senior lenders and/or mezzanine)
3. Off-taker(s)
4. Contractor and equipment supplier
5. Operator
6. Financial Advisors
7. Technical Advisors
8. Legal Advisors
9. Equity Investors
10. Regulatory Agencies
11. Multilateral Agencies / Export Credit Agencies
12. Insurance Providers
13. Hedge providers

Structure of Project Finance



For example, the Acme Coal Co. imports coal. Energen Inc. supplies energy to consumers. The two companies agree to build a power plant to accomplish their respective goals. Typically, the first step would be to sign a memorandum of understanding to set out the intentions of the two parties. This would be followed by an agreement to form a joint venture. Acme Coal and Energen form an SPC (Special Purpose Corporation) called Power Holdings Inc. and divide the shares between them according to their contributions.

Acme Coal, being more established, contributes more capital and takes 70% of the shares. Energen is a smaller company and takes the remaining 30%. The new company has no assets. Power Holdings then signs a construction contract with Acme Construction to build a

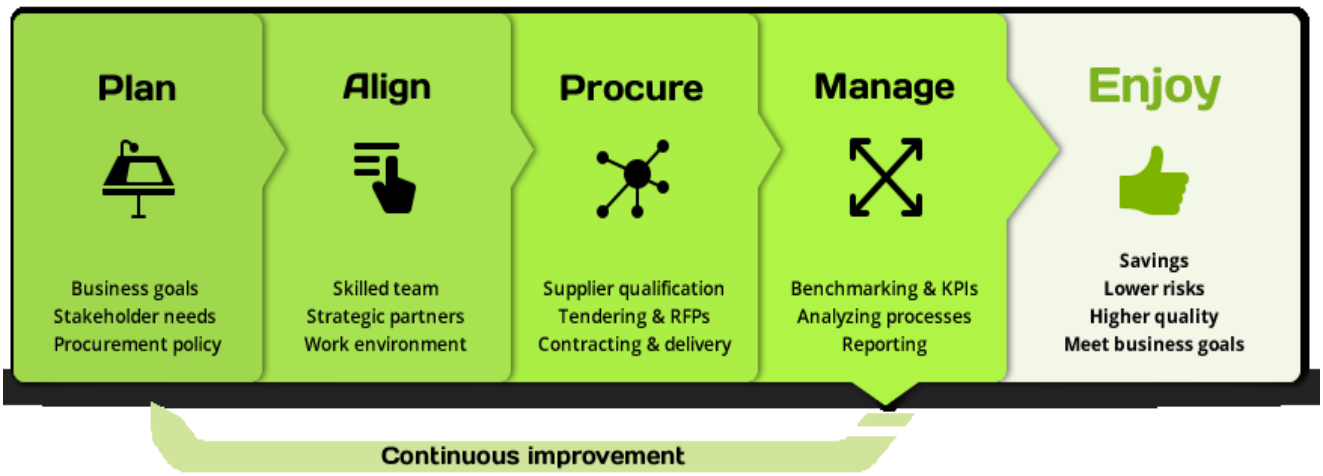
power plant. Acme Construction is an affiliate of Acme Coal and the only company with the know-how to construct a power plant in accordance with Acme's delivery specification.

A power plant can cost hundreds of millions of dollars. To pay Acme Construction, Power Holdings receives financing from a development bank and a commercial bank. These banks provide a guarantee to Acme Construction's financier that the company can pay for the completion of construction. Payment for construction is generally paid as such: 10% up front, 10% midway through construction, 10% shortly before completion, and 70% upon transfer of title to Power Holdings, which becomes the owner of the power plant.

Acme Coal and Energen form Power Manage Inc., another SPC, to manage the facility. The ultimate purpose of the two SPCs (Power Holding and Power Manage) is primarily to protect Acme Coal and Energen. If a disaster happens at the plant, prospective plaintiffs cannot sue Acme Coal or Energen and target their assets because neither company owns or operates the plant. However project financiers may recognize this and require some sort of parent guarantee for up to negotiated amounts of operational liabilities. A Sale and Purchase Agreement (SPA) between Power Manage and Acme Coal supplies raw materials to the power plant. Electricity is then delivered to Energen using a wholesale delivery contract. The net cash flow of the SPC Power Holdings (sales proceeds less costs) will be used to repay the financiers.

PROCUREMENT MANAGEMENT

This Procurement Management process will help you to purchase goods and services from external suppliers. It gives you a complete *procurement process* and *procurement procedures*, which explain step-by-step, how to purchase from suppliers.



Procurement Management Process

1. **Identification of need** – When the company needs goods/services, the company's needs must be identified in order to choose which type of service or product will fit best. Then the buyer needs to find them at the best quality for the best price while also making sure the supplier is able to deliver.
2. **Finding and qualifying the suppliers** – Using the Internet or your supplier databases, you assemble a list of all potential product and/or service providers. Sometimes it's useful to run RFIs or gather additional information, like years on the market and turnover rates, to pre-qualify suppliers.
3. **Requesting proposals** – To make sure you buy the products or services under the best conditions – price, quality, etc. – you request proposals. Based on the results, you know with whom to start negotiations. The RFP results can be considered an overview of current market capability.
4. **Negotiating with suppliers** – To achieve the best conditions regarding prices, terms and delivery, you negotiate with suppliers. It's especially essential if it's the first time you are working with them. This process can help you evaluate their trustworthiness as well.
5. **Contracting** – If both parties, the buyer and the supplier, agree on all terms (pricing, delivery, quality, etc.) you can make it official by signing a contract.
6. **Delivery** – Throughout the delivery process, you need to evaluate the products and services delivered to ensure they are what you had planned to buy, they meet your quality standards, they arrive on schedule and you are charged the prices outlined in the contract.
7. **Analyzing results** – Once the project is complete, it is essential to analyze the process and evaluate its success as well as record observations for future projects. You may need to present the outcomes to company management or relevant stakeholders. The results can be used the next time you need to make a similar purchase.

KARPAGAM ACADEMY OF HIGHER EDUCATION - COIMBATORE 21

SUBJECT CODE:17MBAPS401B SUBJECT: SOFTWARE PROJECT MANAGEMENT

	Unit 1						
S.NO	**Question**	**option 1**	**option2**	**option3**	**option4**		**Answer**
1	SPM stands for _____	Software Project Management	System Process Monitoring.	Software Program Manipulation.	System Program Model.		Software Project Management
2	The _____ is a set of instruction or programs.	Data	Software	Command	Hardware		Software
3	In general any project in real time system must reach a certain _____.	Goal	Result	Output	Objectives		Objective
4	There are _____ numbers of steps in Stepwise project planning for real time projects.	6	7	8	10		10
5	The _____ costs include the salaries and other employment costs of the staff involved in the development project and related tasks.	Operational	setup	Development	Functional		Development
6	In _____ step the identification of project scope and objectives.	7	5	1	0		1
7	_____ step to make the process of making the new system operational	Execution.	Integration.	Installation.	Planning		Planning
8	Groups of methods or techniques are called ____.	plans.	methodologies.	objectives.	products.		plans.
9	Physical part of a computer is called_____	peripherals.	CPU.	Hardware.	Software		Hardware.
10	_____ is a specific plan or design	Project.	Diagram.	Chart.	Management.		Project.
11	In stepwise project planning the zero step deals with _____.	Select project	Define project	Identify project	project authority and rights.		Select project
12	In stepwise project planning the third step deals with _____.	Analyse project characteristics.	identify project scope.	team member..	project manager.		Analyse project characteristics
13	In stepwise project planning the first step deals with _____.	feasibility study.	project design.	identify project scope and objectives.	Identify project infrastructure.		feasibility study.
14	The people / client who have interest in project are called as ____.	project holders.	stakeholders.	project leader	team member		stakeholders.
15	The ____ step in stepwise planning deals for Execute plan.	9	10	1	3		9
16	The task called _____ will produce practical requirements to be fed into the next stage of the planning process.	project analysis	system requirements	feasibility study.	functional requirements		project analysis
17	The data _____ transforms the raw data into useful information	processing	transformation	management	conversion		management
18	The _____requirements is a record of how much the organization is willing to spend on the system	Functional	System	Quality	Process		Functional
19	The ____ step in stepwise planning deals for allocation of resources	5	6	7	8		7
20	_____ is a planned set of interrelated tasks to be executed over a fixed period and within certain cost and other limitations	Project	Idea	Product	Information		Project.
21	_____ is the amount of work that needs to be done in any project.	Effort.	Pending SLOC.	Elapsed work	Expended work.		Effort.
22	The collection of projects that an organization undertakes within a particular planning cycle is called as ……..	Motto	Mission	Programme	portfolio		Portfolio
23	The _____ risks affect the organisation developing or procuring the software	product	process	business	project		business
24	_____takes into account the profitability of a project and the timing of the cash flows.	Net present value.	Internal rate of return	Return on investment	Accounting rate of return		Net present value

25	IRR stands for _____	Internal Rate of Return.	Intermediate Rate of Return	Individual Rate of Return.	none of the above.		Internal Rate of Return.
26	Lack of guidance about making organizational decisions is the problem faced by the ____.	Customer	Team members	Project Managers	Project boards		Project Managers
27	The _____ will look after many projects simultaneously.	project manager	programme manager.	team manger.	resource manager		programme manager.
28	_____ project means the developers and the users belong to the same organization	1nformation system.	Embedded system.	Product based.	In-house		In-house
29	The _____ analysis demands that vary each factor one at a time.	case	sensitivity	Group	Overall		Case
30	Risk _____is a action taken to ensure that the impact of the risk is lessened when it occurs	reduction	avoidance	mitigation.	planning		mitigation.
31	The _____ is also known as the one-shot or once-through model.	waterfall model	spiral model	V process model	evolutionary model		waterfall model.
32	Project may be distinguished by whether their aim is to produce a product and _____.	Goals	Objectives	Mission	Profit		Objectives
33	Risk _____ attempts to minimize the likelihood of the risk occurring	identification.	reduction	avoidance	.transfer		reduction
34	In software prototyping, _____ changes affect more than one part of processing	cosmetic	Local	Global	intermediate		intermediate
35	The outcome of project analysis will be the selection of the most appropriate methods and _____.	science	technique.	formulas	technology.		technology.
36	The process in which the later increments might require modification to earlier increments in known as	software breakage.	hardware breakage.	incremental breakage	product breakage		product breakage
37	In _____the scope of deliverables for an increment is rigidly constrained by an agreed deadline	increment boxing.	time boxing.	deadline verification	checkpoint verification		time boxing.
38	_____ is the set of programs and other operating information used by a computer	Application	Software	Hardware.	middleware		Software
39	An _____ may cause the project to take longer than it would otherwise.	over-estimate	under-estimate	estimate.	non-estimate		over-estimate
40	The software project management deals with _____ type of projects.	Offline	Real-time	Static	Fixed		Real-time
41	A _____ is any item or person required for the execution of the project. Algorithmic model	Data	Resource	Items	Materials		Resource
42	_____ is the process of dealing with or controlling things or people.	Service	Motivation.	management	concern		management
43	In _____ estimation method component tasks are identified and sized and these individual estimates are aggregated	top-down.	bottom up.	procedural oriented	algorithmic model		bottom up.
44	The another name of water fall model is called _____.	classical.	General	Idea	Systematic		Classical
45	The use of estimating by analogy technique is also called _____.	knowledge based reasoning	case base reasoning	decision based reasoning	decision support systems		knowledge based reasoning
46	The V-process model deals with expanding the activity _____.	Tasks	Efforts	testing	Process		Process
47	The critical path is the _____ path through the network	shortest	nearest.	path with interference float	longest		longest
48	In traffic light method of risk reporting, _____indicates that all the activities are on target	green	amber	red	blue		green
49	A ____is an uncertain event or condition that, if it occurs, has a positive or negative effect on a projects objectives	Risk	Activity	critical path.	dangling activity		Risk

50	_____ is one of the ways of identifying possible threats to the success of a project and the measures that might eliminate or reduce them.	Causal mapping	Checklists	Brainstorming	Risk assessment		Causal mapping
51	The _____ prototype is used to modify some ideas.	throw away	evolutionary	general	technique		general
52	_____ is a planned action to be carried out if the particular risk materializes	Contingency plan	Risk transfer.	Integration plan.	Risk mitigation		Contingency plan
53	In _____the risk analysis is carried out immediately after the estimation of effort for each activity.	stepwise approach	product based approach.	process based approach	cash based Approach		stepwise approach
54	Listing of all risks that affect the project is called as _____.	risk Planning.	risk Estimation	risk Evaluation	risk Identification		risk Identification
55	The _____ indicates the planned start and completion dates for each activity	resource schedule	cost schedule	activity schedule	planned schedule		activity schedule
56	A _____ is any item or person required for the execution of the project	resource	project services.	implementation phase	project products		resource
57	The _____ chart is a method of recording and displaying the way in which targets have changed throughout the duration of the project	ball	slip	timeline	gantt		timeline
58	A _____ is likely to have very high priority for close monitoring.	critical path activity	activity	activity with free float	activity with float		critical path activity
59	The _____ chart provides a simple method of comparing actual and planned expenditure	Gantt	PERT	pie	bar		pie
60	The _____ chart need to be redrawn each time when target dates are revised.	ball	slip	timeline	pie		ball

Unit – II

Software Project Management - Resources Planning and Estimation, Different Methods in brief- Function Point Analysis in some details - Use of CASE Tools - Introduction to MS Projects- Design and Development – Schedule - Resource Allocation - Progress Review

INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT

Software project management is the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, monitored and controlled.

History

The history of software project management is closely related to the history of software. Software was developed for dedicated purposes for dedicated machines until the concept of object-oriented programming began to become popular in the 1960's, making *repeatable solutions* possible for the software industry. Dedicated systems could be adapted to other uses thanks to component-based software engineering. Companies quickly understood the relative ease of use that software programming had over hardware circuitry, and the software industry grew very quickly in the 1970's and 1980's.

To manage new development efforts, companies applied proven project management methods, but project schedules slipped during test runs, especially when confusion occurred in the gray zone between the user specifications and the delivered software. To be able to avoid these problems, software project management methods focused on matching user requirements to delivered products, in a method known now as the waterfall model. Since then, analysis of software project management failures has shown that the following are the most common causes.

1. Unrealistic or unarticulated project goals
2. Inaccurate estimates of needed resources
3. Badly defined system requirements
4. Poor reporting of the project's status

5. Unmanaged risks
6. Poor communication among customers, developers, and users
7. Use of immature technology
8. Inability to handle the project's complexity
9. Sloppy development practices
10. Poor project management
11. Stakeholder politics
12. Commercial pressures

The first three items in the list above show the difficulties articulating the needs of the client in such a way that proper resources can deliver the proper project goals. Specific software project management tools are useful and often necessary, but the true art in software project management is applying the correct method and then using tools to support the method. Without a method, tools are worthless. Since the 1960's, several proprietary software project management methods have been developed by software manufacturers for their own use, while computer consulting firms have also developed similar methods for their clients. Today software project management methods are still evolving, but the current trend leads away from the waterfall model to a more cyclic project delivery model that imitates a Software release life cycle.

Software development process.

A software development process is concerned primarily with the production aspect of software development, as opposed to the technical aspect, such as software tools. These processes exist primarily for supporting the management of software development, and are generally skewed toward addressing business concerns. Many software development processes can be run in a similar way to general project management processes. Examples are:

- ✓ Risk management is the process of measuring or assessing risk and then developing strategies to manage the risk. In general, the strategies employed include transferring the risk to another party, avoiding the risk, reducing the negative effect of the risk, and accepting some or all of the consequences of a particular risk. Risk management in software project management begins with the business case for starting the project, which

includes a cost-benefit analysis as well as a list of fallback options for project failure, called a contingency plan.

- ✓ A subset of risk management that is gaining more and more attention is "Opportunity Management", which means the same thing, except that the potential risk outcome will have a positive, rather than a negative impact. Though theoretically handled in the same way, using the term "opportunity" rather than the somewhat negative term "risk" helps to keep a team focused on possible positive outcomes of any given risk register in their projects, such as spin-off projects, windfalls, and free extra resources.
- ✓ Requirements management is the process of identifying, eliciting, documenting, analyzing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders. New or altered computer system[1] Requirements management, which includes Requirements analysis, is an important part of the software engineering process; whereby business analysts or software developers identify the needs or requirements of a client; having identified these requirements they are then in a position to design a solution.
- ✓ Change management is the process of identifying, documenting, analyzing, prioritizing and agreeing on changes to scope (project management) and then controlling changes and communicating to relevant stakeholders. Change impact analysis of new or altered scope, which includes Requirements analysis at the change level, is an important part of the software engineering process; whereby business analysts or software developers identify the altered needs or requirements of a client; having identified these requirements they are then in a position to re-design or modify a solution. Theoretically, each change can impact the timeline and budget of a software project, and therefore by definition must include risk-benefit analysis before approval.
- ✓ Software configuration management is the process of identifying, and documenting the scope itself, which is the software product underway, including all sub-products and changes and enabling communication of these to relevant stakeholders. In general, the processes employed include version control, naming convention (programming), and software archival agreements.

- ✓ Release management is the process of identifying, documenting, prioritizing and agreeing on releases of software and then controlling the release schedule and communicating to relevant stakeholders. Most software projects have access to three software environments to which software can be released; Development, Test, and Production. In very large projects, where distributed teams need to integrate their work before release to users, there will often be more environments for testing, called unit testing, system testing, or integration testing, before release to User acceptance testing (UAT).
- ✓ A subset of release management that is gaining more and more attention is Data Management, as obviously the users can only test based on data that they know, and "real" data is only in the software environment called "production". In order to test their work, programmers must therefore also often create "dummy data" or "data stubs". Traditionally, older versions of a production system were once used for this purpose, but as companies rely more and more on outside contributors for software development, company data may not be released to development teams. In complex environments, data-sets may be created that are then migrated across test environments according to a test release schedule, much like the overall software release schedule.

RESOURCE PLANNING

What is Resource Planning?

The short answer is that resource planning is the step in writing a business plan where you identify all the resources in a proposed project.

What Are Resources?

The resources can be anything from equipment to project sites to people. Here's a short list of some resources you'll have to identify when planning your project:

- ✓ Type of team you'll need
- ✓ Roles and key responsibilities for each team member
- ✓ Number of people required to fill each role
- ✓ What equipment they'll need and its purposes
- ✓ Job locations or meeting rooms required
- ✓ Types and number of equipment needed

- ✓ Total amount of material needed

How to Create a Resource Plan

A resource plan also has to

- Schedule the dates for using the planned resources. That includes when and for how long you'll need the people assigned to your team, equipment rental, project site rental and anything else.
- Identify the amount of resource required per project activity. Each day you'll be using many resources. Use this part of the plan to detail them on a daily basis.
- Create a detailed resource use schedule. Now take those durations and amounts and collect them on a calendar or timeline to make sure you've allocated those resources correctly.

In order to include all that information you want to have a process in place. Basically, it's a three-step process of listing, estimating and then constructing. Let's go through each of those steps in more detail.

Step 1: List the Resources

Simply start a list. Write down all the different resources you'll need. Use the above bullet points as a structure. Who do you need to do the tasks that make up the project? Identify all of those roles. That includes full-time, part-time and contractors.

Now, what about equipment? Do the same as you did with the labor, identify all the equipment you're going to need to get the project completed. That list should include anything from office equipment such as computers, photocopies and other devices to telecommunications and machinery. The next item on the list is the materials. What are the non-consumable materials you'll need to complete the project activities? These can be materials necessary to build physical deliverables, such as wood, steel, and concrete.

Step 2: Estimate How Many Resources

Now you want to figure out, the best you can, how many of of the listed resources you'll need throughout the life cycle of the project. For example, how many hours are you going to need from your team? Break that down per role. Do the same for your equipment. How many pieces of equipment are going to be necessary?

The same goes with your material. Estimate what about of material, in terms of square footage, number of units, etc., is going to be necessary for the project. How much hardware do you need to buy, or will you have to license software? Get as accurate an estimate as you can. If possible, try to note the date the resources are needed and what the consumption rate per day, week or month is. The more data you have, the better you can allocate various resources.

Step 3: Construct a Resource Schedule

Use the information you've collected in the first two steps to build a detailed resource schedule. You'll want to specify the resources required to complete the project, the time-frames for the consumption of each of those resources and the quantity of time each resource is going to require per week and/or month. Then you want to add up the total quantity of resources consumed per week and/or month. Don't forget to identify the assumptions and constraints you feel are going to arise over the life cycle of the project.

PROJECT COST ESTIMATION

Cost estimations

Cost estimations depend upon accurate estimations of time and resources. There are various methods for estimating cost – analogy (looking at similar projects), parametric (derived from a mathematical formula) or bottom-up (adding up the costs of all the parts to obtain the cost of the whole, etc. What is important is that the method used for estimating project cost produces an accurate result.

Tools and Techniques for Project Cost Estimation

Expert Judgment	**Expert judgment** uses the experience and knowledge of experts to estimate the cost of the project. This technique can take into account unique factors specific to the project. However, it can also be biased.
Analogous Estimating	**Analogous estimating** uses historical data from similar projects as a basis for the cost estimate. The estimate can be adjusted for known differences between the projects. This type of estimate is usually used in the early phases of a project and is less accurate than other methods.
Parametric Estimating	**Parametric estimating** uses statistical modeling to develop a cost estimate. It uses historical data of key cost drivers to calculate an estimate for different parameters such as cost and duration. For example, square footage is used in some construction projects.

Bottom-Up Estimating	**Bottom-up estimating** uses the estimates of individual work packages which are then summarized or "rolled up" to determine an overall cost estimate for the project. This type of estimate is generally more accurate than other methods since it is looking at costs from a more granular perspective.
Three-Point Estimates	**Three-point estimates** originated with the *Program Evaluation and Review Technique (PERT)*. This method uses three estimates to define an approximate range for an activities cost: Most Likely (Cm), Optimistic (Co), and Pessimistic (Cp). The cost estimate is calculated using a weighted average: Cost Estimate = (Co + 4Cm + Cp)/6
Reserve Analysis	**Reserve analysis** is used to determine how much contingency reserve, if any, should be allocated to the project. This funding is used to account for cost uncertainty.
Cost of Quality	**Cost of Quality (COQ)** includes money spent during the project to avoid failures and money spent during and after the project due to failures. During cost estimation, assumptions about the COQ can be included in the project cost estimate.
Project Management Estimating Software	**Project management estimating software** includes cost estimating software applications, spreadsheets, simulation applications, and statistical software tools. This type of software is especially useful for looking at cost estimation alternatives.
Vendor Bid Analysis	**Vendor analysis** can be used to estimate what the project should cost by comparing the bids submitted by multiple vendors.

AN INTRODUCTION TO FUNCTION POINT ANALYSIS

Brief History

Function Point Analysis was developed first by Allan J. Albrecht in the mid 1970s. It was an attempt to overcome difficulties associated with lines of code as a measure of software size, and to assist in developing a mechanism to predict effort associated with software development. The method was first published in 1979, then later in 1983 . In 1984 Albrecht refined the method and since 1986, when the International Function Point User Group (IFPUG) was set up, several versions of the Function Point Counting Practices Manual have been published by IFPUG. The current version of the IFPUG Manual is 4.1.

Introduction to Function Point Analysis

One of the initial design criteria for function points was to provide a mechanism that both software developers and users could utilize to define functional requirements. It was determined that the best way to gain an understanding of the users' needs was to approach their problem from the perspective of how they view the results an automated system produces. Therefore, one of the primary goals of Function Point Analysis is to evaluate a system's capabilities from a user's point of view. To achieve this goal, the analysis is based upon the various ways users interact with computerized systems. From a user's perspective a system assists them in doing their job by providing five (5) basic functions. Two of these address the data requirements of an end user and are referred to as Data Functions. The remaining three address the user's need to access data and are referred to as Transactional Functions.

The Five Components of Function Points

- **Data Functions**
    1. Internal Logical Files
    2. External Interface Files
- **Transactional Functions**
    1. External Inputs
    2. External Outputs
    3. External Inquiries

Internal Logical Files - The first data function allows users to utilize data they are responsible for maintaining. For example, a pilot may enter navigational data through a display in the cockpit prior to departure. The data is stored in a file for use and can be modified during the mission. Therefore the pilot is responsible for maintaining the file that contains the navigational information. Logical groupings of data in a system, maintained by an end user, are referred to as Internal Logical Files (ILF).

External Interface Files - The second Data Function a system provides an end user is also related to logical groupings of data. In this case the user is not responsible for maintaining the data. The data resides in another system and is maintained by another user or system.

The user of the system being counted requires this data for reference purposes only. For example, it may be necessary for a pilot to reference position data from a satellite or ground-

based facility during flight. The pilot does not have the responsibility for updating data at these sites but must reference it during the flight. Groupings of data from another system that are used only for reference purposes are defined as External Interface Files (EIF).

The remaining functions address the user's capability to access the data contained in ILFs and EIFs. This capability includes maintaining, inquiring and outputting of data. These are referred to as Transactional Functions.

External Input - The first Transactional Function allows a user to maintain Internal Logical Files (ILFs) through the ability to add, change and delete the data. For example, a pilot can add, change and delete navigational information prior to and during the mission. In this case the pilot is utilizing a transaction referred to as an External Input (EI). An External Input gives the user the capability to maintain the data in ILF's through adding, changing and deleting its contents.

External Output - The next Transactional Function gives the user the ability to produce outputs. For example a pilot has the ability to separately display ground speed, true air speed and calibrated air speed. The results displayed are derived using data that is maintained and data that is referenced. In function point terminology the resulting display is called an External Output (EO).

External Inquiries - The final capability provided to users through a computerized system addresses the requirement to select and display specific data from files. To accomplish this a user inputs selection information that is used to retrieve data that meets the specific criteria. In this situation there is no manipulation of the data. It is a direct retrieval of information contained on the files.

For example if a pilot displays terrain clearance data that was previously set, the resulting output is the direct retrieval of stored information. These transactions are referred to as External Inquiries (EQ). In addition to the five functional components described above there are two adjustment factors that need to be considered in Function Point Analysis.

Functional Complexity - The first adjustment factor considers the Functional Complexity for each unique function. Functional Complexity is determined based on the combination of data groupings and data elements of a particular function. The number of data elements and unique

groupings are counted and compared to a complexity matrix that will rate the function as low, average or high complexity. Each of the five functional components (ILF, EIF, EI, EO and EQ) has its own unique complexity matrix.

Benefits of Function Point Analysis

- Function Points can be used to size software applications accurately. Sizing is an important component in determining productivity (outputs/inputs).
- They can be counted by different people, at different times, to obtain the same measure within a reasonable margin of error.
- Function Points are easily understood by the non technical user. This helps communicate sizing information to a user or customer.
- Function Points can be used to determine whether a tool, a language, an environment, is more productive when compared with others.

CASE (Computer-aided software engineering) TOOLS

Meaning of CASE Tools

Computer-aided software engineering (**CASE**) is the domain of software tools used to design and implement applications. CASE tools are similar to and were partly inspired by computer-aided design (CAD) tools used for designing hardware products. CASE tools are used for developing high-quality, defect-free, and maintainable software.

Introduction to CASE

CASE stands for **C**omputer **A**ided **S**oftware **E**ngineering. It means, development and maintenance of software projects with help of various automated software tools.

CASE Tools

- CASE tools are set of software application programs, which are used to automate SDLC activities. CASE tools are used by software project managers, analysts and engineers to develop software system.

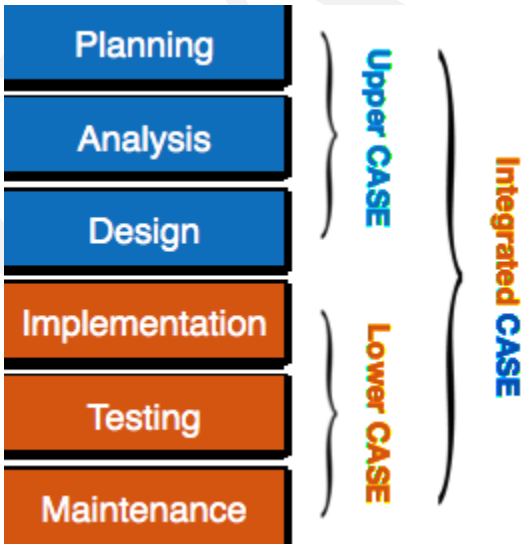
- There are number of CASE tools available to simplify various stages of Software Development Life Cycle such as Analysis tools, Design tools, Project management tools, Database Management tools, Documentation tools are to name a few.
- Use of CASE tools accelerates the development of project to produce desired result and helps to uncover flaws before moving ahead with next stage in software development.

Components of CASE Tools

CASE tools can be broadly divided into the following parts based on their use at a particular SDLC stage:

1. **Central Repository**

   CASE tools require a central repository, which can serve as a source of common, integrated and consistent information. Central repository is a central place of storage where product specifications, requirement documents, related reports and diagrams, other useful information regarding management is stored. Central repository also serves as data dictionary.



2. **Upper Case Tools**

   Upper CASE tools are used in planning, analysis and design stages of SDLC.

3. **Lower Case Tools**

Lower CASE tools are used in implementation, testing and maintenance.

4. **Integrated Case Tools**

   Integrated CASE tools are helpful in all the stages of SDLC, from Requirement gathering to Testing and documentation.

CASE tools can be grouped together if they have similar functionality, process activities and capability of getting integrated with other tools.

Case Tools Types

1. **Diagram tools**

These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form. For example, Flow Chart Maker tool for creating state-of-the-art flowcharts.

2. **Process Modeling Tools**

Process modeling is method to create software process model, which is used to develop the software. Process modeling tools help the managers to choose a process model or modify it as per the requirement of software product. For example, EPF Composer

3. **Project Management Tools**

These tools are used for project planning, cost and effort estimation, project scheduling and resource planning. Managers have to strictly comply project execution with every mentioned step in software project management. Project management tools help in storing and sharing project information in real-time throughout the organization. For example, Creative Pro Office, Trac Project, Basecamp.

4. **Documentation Tools**

Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project.

Documentation tools generate documents for technical users and end users. Technical users are mostly in-house professionals of the development team who refer to system manual, reference manual, training manual, installation manuals etc. The end user documents describe the functioning and how-to of the system such as user manual. For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

5. Analysis Tools

These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies or erroneous omissions. For example, Accept 360, Accompa, CaseComplete for requirement analysis, Visible Analyst for total analysis.

6. Design Tools

These tools help software designers to design the block structure of the software, which may further be broken down in smaller modules using refinement techniques. These tools provides detailing of each module and interconnections among modules. For example, Animated Software Design

7. Configuration Management Tools

An instance of software is released under one version. Configuration Management tools deal with –

Version and revision management

Baseline configuration management

Change control management

CASE tools help in this by automatic tracking, version management and release management. For example, Fossil, Git, Accu REV.

8. Change Control Tools

These tools are considered as a part of configuration management tools. They deal with changes made to the software after its baseline is fixed or when the software is first released. CASE tools automate change tracking, file management, code management and more. It also helps in enforcing change policy of the organization.

9. Programming Tools

These tools consist of programming environments like IDE (Integrated Development Environment), in-built modules library and simulation tools. These tools provide comprehensive aid in building software product and include features for simulation and testing. For example, Cscope to search code in C, Eclipse.

10. Prototyping Tools

Software prototype is simulated version of the intended software product. Prototype provides initial look and feel of the product and simulates few aspect of actual product.

Prototyping CASE tools essentially come with graphical libraries. They can create hardware independent user interfaces and design. These tools help us to build rapid prototypes based on existing information. In addition, they provide simulation of software prototype. For example, Serena prototype composer, Mockup Builder.

11. Web Development Tools

These tools assist in designing web pages with all allied elements like forms, text, script, graphic and so on. Web tools also provide live preview of what is being developed and how will it look after completion. For example, Fontello, Adobe Edge Inspect, Foundation 3, Brackets.

12. Quality Assurance Tools

Quality assurance in a software organization is monitoring the engineering process and methods adopted to develop the software product in order to ensure conformance of quality as per organization standards. QA tools consist of configuration and change control tools and software testing tools. For example, SoapTest, AppsWatch, JMeter.

13. Maintenance Tools

Software maintenance includes modifications in the software product after it is delivered. Automatic logging and error reporting techniques, automatic error ticket generation and root cause Analysis are few CASE tools, which help software organization in maintenance phase of SDLC. For example, Bugzilla for defect tracking, HP Quality Center.

CASE Risk Factors

Some of the most significant risk factors for organizations adopting CASE technology include:

- ✓ Inadequate standardization. Organizations usually have to tailor and adopt methodologies and tools to their specific requirements. Doing so may require significant effort to integrate both divergent technologies as well as divergent methods. For example, before the adoption of the UML standard the diagram conventions and methods for designing object-oriented models were vastly different among followers of Jacobsen, Booch, and Rumbaugh.

- ✓ Unrealistic expectations. The proponents of CASE technology—especially vendors marketing expensive tool sets—often hype expectations that the new approach will be a silver bullet that solves all problems. In reality no such technology can do that and if organizations approach CASE with unrealistic expectations they will inevitably be disappointed.
- ✓ Inadequate training. As with any new technology, CASE requires time to train people in how to use the tools and to get up to speed with them. CASE projects can fail if practitioners are not given adequate time for training or if the first project attempted with the new technology is itself highly mission critical and fraught with risk.
- ✓ Inadequate process control. CASE provides significant new capabilities to utilize new types of tools in innovative ways. Without the proper process guidance and controls these new capabilities can cause significant new problems as well.

INTRODUCTION TO MS PROJECTS

Microsoft Project is a project management software program developed and sold by Microsoft, designed to assist a project manager in developing a schedule, assigning resources to tasks, tracking progress, managing the budget, and analyzing workloads.

Project creates budgets based on assignment work and resource rates. As resources are assigned to tasks and assignment work estimated, the program calculates the cost, equal to the work times the rate, which rolls up to the task level and then to any summary task, and finally to the project level.

Each resource can have its own calendar, which defines what days and shifts a resource is available. Microsoft Project is not suitable for solving problems of available materials (resources) constrained production. Additional software is necessary to manage a complex facility that produces physical goods.

Scheduling

It is the science of using mathematical calculations and logic to generate time effective sequence of task considering any resource and cost constraints. Schedule is part of the Plan. In Project Management Methodology, schedule would only mean listing of a project's milestones,

tasks/activities, and deliverables, with start and finish dates. Of course the schedule is linked with resources, budgets and dependencies.

However, in this tutorial for MS Project (and in all available help for MS Project) the word 'Plan' is used as a 'Schedule' being created in MS Project.

This is because of two reasons. One, MS Project does more than just create a schedule it can establish dependencies among tasks, it can create constraints, it can resolve resource conflicts, and it can also help in reviewing cost and schedule performance over the duration of the project. So it does help in more than just creating a Schedule. This it makes sense for Microsoft to market MS Project as a Plan Creator rather than over-simplifying it as just a schedule creator.

Two, it is due to limitation of generally accepted form of English language, where a schedule can be both in a noun as well as verb form. As a noun, a Schedule is like a time table or a series of things to be done or of events to occur at or during a particular time or period. And in the verb form, schedule is to plan for a certain date. Therefore it is much easier to say that, "One can schedule a plan from a start date" but very awkward to say, "One can schedule a schedule from a start date". The distinction is important for you as a project manager, but as far as MS project is concerned the noun form of Schedule is a Plan. Of course, a project manager should also be able to answer other project-related questions as well. For example −

- Why this project needs to be run by the organization?
- What's the best way to communicate project details to the stakeholders?
- What is the risk management plan?
- How the vendors are going to be managed?
- How the project is tracked and monitored?
- How the quality is measured and qualified?

MS Project can help you −

Visualize your project plan in standard defined formats. Schedule tasks and resources consistently and effectively. Track information about the work, duration, and resource requirements for your project. Generate reports to share in progress meetings.

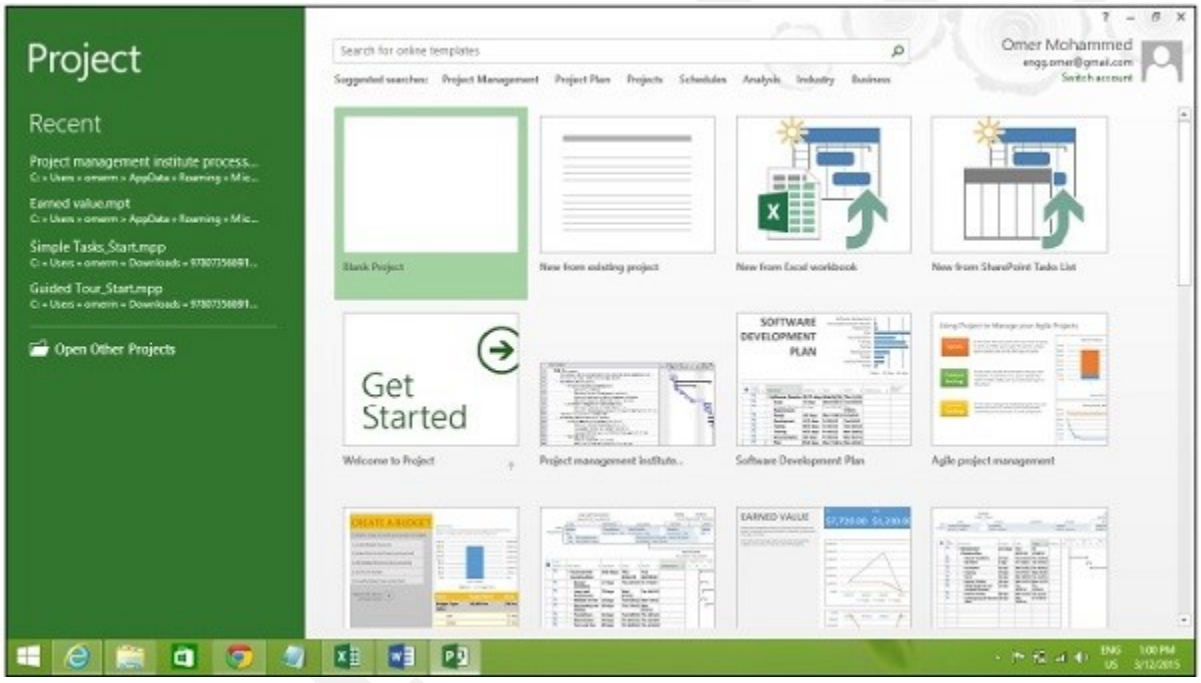
MS Project UI

Windows 7 − Click on Start menu, point to All Programs, click Microsoft Office, and then click Project 2013.

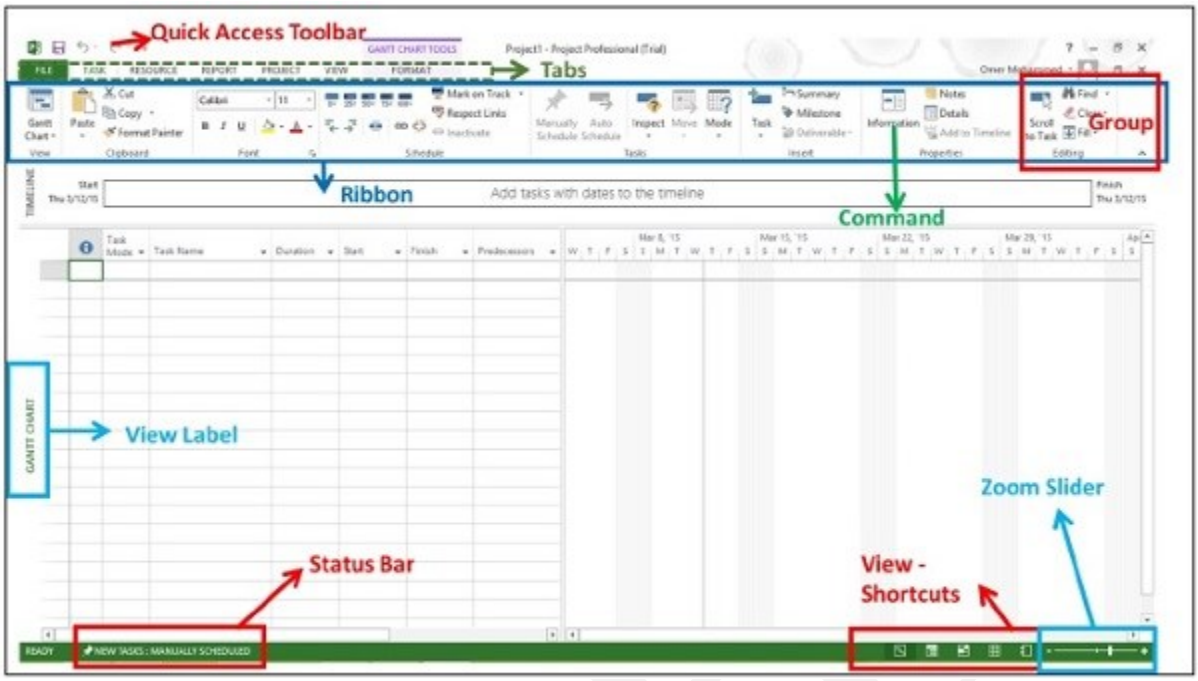
Windows 8 − On the Start screen, tap or click Project 2013.

Windows 10 − Click on Start menu → All apps → Microsoft Office → Project 2013.

The following screen is the Project's start screen. Here you have options to open a new plan, some other plans, and even a new plan template.



Click the Blank Project Tab. The following screen pops up.



The screen should have the MS Project interface displayed. The major part of this interface are −

Quick Access Toolbar − A customizable area where you can add the frequently used commands.

Tabs on the **Ribbon, Groups** − With the release of Microsoft Office 2007 came the "Fluent User Interface" or "Fluent UI", which replaced menus and customizable toolbars with a single "Office menu", a miniature toolbar known as "quick-access toolbar" and what came to be known as the ribbon having multiple tabs, each holding a toolbar bearing buttons and occasionally other controls.

Toolbar controls have heterogeneous sizes and are classified in visually distinguishable Groups. Groups are collections of related commands. Each tab is divided into multiple groups.

Commands − The specific features you use to perform actions in Project. Each tab contains several commands. If you point at a command you will see a description in a tooltip.

View Label − This appears along the left edge of the active view. **Active view** is the one you can see in the main window at a given point in time. Project includes lots of views like Gantt Chart view, Network Diagram view, Task Usage view, etc. The View label just

tells you about the view you are using currently. Project can display a single view or multiple views in separate panes.

View Shortcuts − This lets you switch between frequently used views in Project.

Zoom Slider − Simply zooms the active view in or out.

Status bar − Displays details like the scheduling mode of new tasks (manual or automatic) and details of filter applied to the active view.

In project management terminology, resources are required to carry out the project tasks. They can be people, equipment, facilities, funding, or anything (except labor) required for the completion of a project task. **Optimum Resource Scheduling** is the key to successful project management.

Resource Types

Work resources − People and equipment to complete the tasks.

Cost resources − Financial cost associated with a task. Travel expenses, food expenses, etc.

Material resources − Consumables used as project proceeds. For example, paint being used while painting a wall.

Note − Be aware of the crucial difference between People and Equipment resources. People resources will have limited work hours, say 6, 8 or 12 hours. Equipment resources have different working capacities of 2, 8 or 24 hours and could have maintenance breaks as well. Also note, that it is possible multiple people resources might be using one equipment resource, or one equipment might be accomplishing multiple tasks.

Enter Work Resource Names

You can enter resource names according to your convenience.

Resource	Example
Work resource as an identified person	John, Kevin
Work resource as a job function or group	Engineer, Coordinator, Typist
Work resource as an equipment	Lathe machine, Earth mover

Resource Max Capacity

Max Units field represents the maximum capacity of a resource to work on assigned tasks. 100% stands for 100 percent of resource's working time is available for work on task assigned. The resource is available full-time on each workday. If the resource gets allocated to task or tasks that would require more than his/its work hours, the resource is over allocated and MS Project will indicate this in red formatting.

a resource does not represent an individual person but a job function, where a group of people with the same skill set can work on the task, we can enter larger Max Units to represent the number of people in the group. So 400% would indicate, 4 individual people working full-time every workday.

Check Resource Allocations

Relationship between a resource's capacity and task assignments is called **allocation**.

This can defined by 3 states −

Under allocated − An Engineer who works for 40 hours a week, has work assigned for only 20 hours.

Fully allocated − A skilled worker who works for 40 hours a week, is assigned 40 hours of work in that week.

Over allocated − A carpenter is assigned 65 hours of work, when he only has a 40 hour work week.

Resolve Resource Over Allocation

One would need to either change the scope (reduce the amount of work), assign more resources, or accept a longer schedule to resolve overallocation.

This can be achieved by using some of the following techniques −

Adjust Schedule

By changing its lead or lag time when the resource has more tasks assigned than can be completed during a given time period. If you add delay that is less than or equal to the amount of slack on the task, you will not affect the finish date of the project.

By default when you link tasks, they are assigned a "Finish to Start" relationship. In this relationship,

Lead − Lead time causes successor task to begin before its predecessor tasks ends.

Lag − Lag time causes successor task to start after its predecessor task ends.

Click Task Tab → double-click the required Task under Task Name column → Task Information dialog box opens → Predecessors Tab.

Under Lag heading column, enter the lag in terms of hours, days, weeks, or years. You can also apply lag or lead as a percentage. If you enter 50% for the selected Task which is 6 days long, the task is delayed by 3 days after the predecessor ends.

Lag is entered as positive units and lead in negative units (example,-3d or -50%).

Check Plan's Cost

Types of cost in a project life cycle includes −

Baseline costs − All planned costs as saved in baseline plan.

Actual costs − Costs that have been incurred for tasks, resources or assignments.

Remaining costs − Difference between baseline/current costs and actual costs.

Current costs − When plans are changed due to assigning or removing resources, or adding or subtracting tasks, MS Project 2013 will recalculate all costs. This will appear under the fields labeled Cost or Total Cost. If you have started to track actual cost, it will include actual cost + remaining cost per task.

****

KARPAGAM ACADEMY OF HIGHER EDUCATION - COIMBATORE 21

SUBJECT CODE:17MBAPS401B SUBJECT: SOFTWARE PROJECT MANAGEMENT

	Unit 2						
S.NO	**Question**	**option 1**	**option2**	**option3**	**option4**		**Answer**
1	Traffic light method is used for _____.	risk reporting	risk mitigation.	risk contingency	risk transfer		risk reporting
2	The _____ method, the activities that can proceed at the same time are ordered according to a set of simple criteria.	total float priority	total float priority	priority list.	interference priority list		ordered list priority
3	In _____activities are ordered according to their total float, those with the smallest total float having the highest priority.	priority list	resource planning	product planning	total float priority		total float priority
4	The _____is the amount of time an activity may be delayed without affecting any subsequent activity.	total float.	float	free float	interface float		free float
5	The _____ chart is useful both during the execution of a project and post implementation	ball	cumulative expenditure	slip	timeline		timeline
6	Project risk management is best described as	managing responses to threats	identifying and acknowledging threats and opportunities.	planning responses to threats.	minimising threats and maximising opportunities.		minimising threats and maximising opportunities.
7	CPM stands for _____.	critical path method	caution path method	critical process method.	caution process method		critical process method
8	The ____ model used for effort drivers concept in software estimation technique.	expert judgement	analogy	algorithmic.	top-down		expert judgement
9	In project reporting, project manger reports to _____.	team leader	project member.	steering committee	quality control section		steering committee
10	Social interaction method of reporting is an example for _____.	oral formal regular	oral formal adhoc	written formal regular	oral informal ad hoc		oral informal ad hoc
11	The overall responsibility for ensuring satisfactory progress on a project is often the role of the	project team	project manager	project analyst	project in charge		project team
12	The _____ schedule shows the planned cumulative expenditure incurred by the use of resources over time	activity	resource	cost	product		cost
13	A person with an interest or concern in something, especially a business who called_____.	Manager.	Administrator.	worker	Stakeholder		Stakeholder
14	In ball chart, if the actual start or finish date for an activity is later than the target date, the circle is colored in _____.	reduction	green	amber	blue		red
15	The use of _____ is called case based reasoning in estimation	analogy.	concept	logic	ideas		ideas
16	A _____ value analysis based in assigning a value to each task or work package	assigned	earned	general	calculated.		earned
17	A security plan begins with a(n) is _____.	security policy	risk assessment.	implementation plan.	security organization		risk assessment.
18	The only payment system that is instantly convertible without intermediation is _____.	Credit card.	Accumulating balance	Stored value.	Cash.		Cash.
19	The _____chart is a method of recording and displaying the way in which targets have changed in duration of the project.	Line	timeline	pert	bar		timeline
20	_____ is the first and foremost step in the recruitment process	Creating a job holders profile	Creating a company profile	Creating a job.	Creating a job specification		Creating a job specification
21	Higher level needs, according to Maslow, is the need for _____.	self acquisition	self-actualization	self realization	self motivation		self-actualization
22	In expectancy theory of motivation, _____ is the belief that working harder will lead to a better performance	perceived value.	expectancy	instrumentality	inspiration		expectancy
23	In expectancy theory of motivation _____is the belief that better performance will be rewarded	instrumentality.	anticipation	inspiration	incentive		instrumentality.
24	In a project team, the _____is a good team player who is willing to undertake less attractive tasks if they were needed for team success	a company worker.	team worker	monitor evaluator	complete finisher		a company worker.
25	In _____stage of a team, conflicts are largely settled and a feeling of group identity emerges.	forming	adjourning	deleting.	performing		deleting.
26	In the storming stage of a team, the _____as various members of the group try to exert leadership.	conflicts are settled.	conflicts arise.	team leader disbands.	team members disbands		conflicts arise.
27	In a project team, the _____are skilled at creating a good working environment	chair	sharpe	monitor-evaluator	team worker		team worker

28	The _____means the efforts of each participant are added to get the final result.	additive tasks.	preservative tasks	conjunctive tasks.	non-additive tasks		additive tasks.
29	In the _____the effectiveness of group depends on someone coming up with the right answer and the others recognizing it as being correct.	compensatory tasks	conjunctive tasks.	non-compensatory tasks	disjunctive tasks		disjunctive tasks
30	The group members are asked to provide the estimate of effort required for software development and an example for_____.	non-compensatory tasks	non-disjunctive tasks	non-additive tasks	compensatory tasks		compensatory tasks
31	The ____ decision making is more complex and often require a degree of creativity	structured	Unstructured	Group	Single		unstructured
32	Code reviews could be seen as an example of _____tasks	additive.	conjunctive	non-conjunctive.	compensatory.		compensatory.
33	In decision making, ____ are based on stereotypes	escalation of commitment.	faulty heuristics	information overload	participatory tasks.		faulty heuristics
34	People in groups sometimes make decisions that carry more risk than where they make the decision on their own and we called it as _____.	risky shift	risky task	unstructured decision making	structured decision making		risky shift
35	_____is based on the idea of some kind of authority or power.	decision making	Motivation.	leadership	Organizational behaviour		leadership
36	_____is a kind of position power based on having access to those who have power.	Legitimate power	Non-coercive power	Information power.	Connection power		Connection power
37	In project management the ____ number of types of contract in management	3	5	7	2		3
38	_____is based on the idea of some kind of authority or power	decision making	Motivation.	leadership	Organizational behaviour		leadership
39	The_____is one of the personal power	Legitimate power	Non-coercive power	Information power	Connection power		Information power
40	The _____makes decision alone with the close supervision of their implementation	directive autocrat	permissive autocrat	active democrat	permissive democrat		directive autocrat
41	Permissive Autocrat is a kind of leadership style where a person _____.	makes decision alone	makes decision participative.	makes decision and gives subordinates	makes decision alone with the supervision		makes decision alone
42	The _____concerns the departmentalization of organization	differentiation	separation.	demarcation	discrimination		differentiation
43	The _____ approach, the programmers and systems analysts are grouped together in project teams	functional	task oriented.	egoless.	hierarchical.		task oriented
44	_____ tasks mean the efforts of each person are added to get the final result.	specific	additive	general	initial		additive
45	In _____the assessment of the quality of delivered products will need to be thorough.	dispersed teams.	team working	virtual teams	departmentalization		dispersed teams.
46	The _____will be of concern at all stages of project planning and execution, but will be of particular interest at the some points of stepwise frame work	software coding	software testing	software quality	software verification		software quality
47	In software quality, the _____is concerned with how well the functions are to operate	resource specification	functional specification.	software specification.	quality specification		quality specification
48	The _____makes it difficult to know that a particular task in a project has been completed satisfactorily.	increasing criticality of software.	accumulating errors during software development	intangibility of software.	decreasing criticality of software		intangibility of software.
49	The measure of _____is more helpful than total faults in a program as a means of judging the quality of a program.	faults per thousand lines of code	faults per hundred line of code	faults per SLOC	faults in one programming block		faults per thousand lines of code
50	In quality specification, _____is the range of values within which it is planned the quality measurement value should lie.	now	minimally acceptable.	target range.	scale		target range.
51	The _____is the percentage of a particular time interval that a system is usable.	reliability.	maintainability.	extendibility.	availability.		availability.
52	The _____is the extent to which a program satisfies its specifications and fulfils the users objectives	correctness.	reliability.	integrity	usability		integrity
53	The ____ tasks are where the progress is governed by the rate of the slowest performer	specific	conjunctive	disjunctive	goals		conjunctive
54	The _____is the effort required to transfer one program from one h/w configuration or s/w system environment to another	flexibility	portability	reusability	portfolio		portability

55	The _____describes the fundamental features of quality management system (QMS) and the terminology used	ISO9000	ISO9001	ISO9001-2000	ISO9001-2001		ISO9000
56	The main outcome of risk identification, in a risk management process, is to:	identify and determine the relative importance of the project risks	identify and describe all risks that might occur on the project	identify and determine the responses to the project risks.	identify and describe risks that have occurred on previous projects		identify and describe all risks that might occur on the project.
57	In ISO 9126 identifies _____ software quality characteristics.	2	3	6	8		6
58	The possibility that particular assumptions are incorrect is assessed and documented in a _____.	system log	log	database log	risk log.		risk log.
59	The _____evaluates the effectiveness of the installed system after a set period of operation	pre project review	project review	post project review	peer review.		post project review
60	In students projects, _____gives the details about verifying and validating the product of the activity	pre-requisites	quality checks.	validation and verification report	product report		quality checks.

Unit – III

Testing – Overview of Test plan- Generation of Test cases, Test data - Types of Testing - Quality concepts – ISO, CMM,- Production / Implementation – User acceptance tests,- parallel runs . – Maintenance – Types - Adaptive, Corrective, Preventive version control and configuration management – documentation methods.

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.

Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- it is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

Test data generation, an important part of software testing, is the process of creating a set of data for testing the adequacy of new or revised software applications. It may be the actual data that has been taken from previous operations or artificial data created for this purpose. Test Data Generation is seen to be a complex problem and though a lot of solutions have come forth most of them are limited to toy programs. The use of dynamic memory allocation in most of the code written in industry is the most severe problem that the Test Data Generators face as the usage of the software then becomes highly unpredictable, due to this it becomes harder to anticipate the paths that the program could take making it nearly impossible for the Test Data Generators to generate exhaustive Test Data. However, in the past decade significant progress has been made in tackling this problem better by the use of genetic algorithms and other analysis algorithms.

Moreover, Software Testing is an important part of the Software Development Life Cycle and is basically labor-intensive. It also accounts for nearly one third of the cost of the system development. In this view the problem of generating quality test data quickly, efficiently and accurately is seen to be important.

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Software Validation

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

- Validation ensures the product under development is as per the user requirements.
- Validation answers the question – "Are we developing the product which attempts all that user needs from this software ?".
- Validation emphasizes on user requirements.

Software Verification

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

- Verification ensures the product being developed is according to design specifications.
- Verification answers the question– "Are we developing this product by firmly following all design specifications ?"
- Verifications concentrates on the design and system specifications.

Target of the test are -

- **Errors** - These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.
- **Fault** - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.
- **Failure** - failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

Manual Vs Automated Testing

Testing can either be done manually or using an automated testing tool:

- **Manual** - This testing is performed without taking help of automated testing tools. The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager.

  Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.

- **Automated** This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools.

A test needs to check if a webpage can be opened in Internet Explorer. This can be easily done with manual testing. But to check if the web-server can take the load of 1 million users, it is quite impossible to test manually.

There are software and hardware tools which helps tester in conducting load testing, stress testing, regression testing.

Testing Approaches

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

When functionality is being tested without taking the actual implementation in concern it is known as black-box testing. The other side is known as white-box testing where not only functionality is tested but the way it is implemented is also analyzed.

Exhaustive tests are the best-desired method for a perfect testing. Every single possible value in the range of the input and output values is tested. It is not possible to test each and every value in real world scenario if the range of values is large.

Black-box testing

It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if

the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

Black-box testing techniques:

- **Equivalence class** - The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
- **Boundary values** - The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
- **Cause-effect graphing** - In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
- **Pair-wise Testing** - The behavior of software depends on multiple parameters. In pairwise testing, the multiple parameters are tested pair-wise for their different values.
- **State-based testing** - The system changes state on provision of input. These systems are tested based on their states and input.

White-box testing

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing.

In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

The below are some White-box testing techniques:

- **Control-flow testing** - The purpose of the control-flow testing to set up test cases which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.
- **Data-flow testing** - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

Testing Levels

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified.

Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels -

Unit Testing

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

Integration Testing

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updation etc.

System Testing

The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

- **Functionality testing** - Tests all functionalities of the software against the requirement.
- **Performance testing** - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.
- **Security & Portability** - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

Acceptance Testing

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

- **Alpha testing** - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.
- **Beta testing** - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

Regression Testing

Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code. This is known as regression testing.

Testing Documentation

Testing documents are prepared at different stages -

Before Testing

Testing starts with test cases generation. Following documents are needed for reference –

- **SRS document** - Functional Requirements document
- **Test Policy document** - This describes how far testing should take place before releasing the product.
- **Test Strategy document** - This mentions detail aspects of test team, responsibility matrix and rights/responsibility of test manager and test engineer.
- **Traceability Matrix document** - This is SDLC document, which is related to requirement gathering process. As new requirements come, they are added to this matrix. These matrices help testers know the source of requirement. They can be traced forward and backward.

While Being Tested

The following documents may be required while testing is started and is being done:

- **Test Case document** - This document contains list of tests required to be conducted. It includes Unit test plan, Integration test plan, System test plan and Acceptance test plan.
- **Test description** - This document is a detailed description of all test cases and procedures to execute them.
- **Test case report** - This document contains test case report as a result of the test.
- **Test logs** - This document contains test logs for every test case report.

After Testing

The following documents may be generated after testing :

- **Test summary** - This test summary is collective analysis of all test reports and logs. It summarizes and concludes if the software is ready to be launched. The software is released under version control system if it is ready to launch.

Testing vs. Quality Control, Quality Assurance and Audit

We need to understand that software testing is different from software quality assurance, software quality control and software auditing.

- **Software quality assurance** - These are software development process monitoring means, by which it is assured that all the measures are taken as per the standards of organization. This monitoring is done to make sure that proper software development methods were followed.
- **Software quality control** - This is a system to maintain the quality of software product. It may include functional and non-functional aspects of software product, which enhance the goodwill of the organization. This system makes sure that the customer is receiving quality product for their requirement and the product certified as 'fit for use'.
- **Software audit** - This is a review of procedure used by the organization to develop the software. A team of auditors, independent of development team examines the software process, procedure, requirements and other aspects of SDLC. The purpose of software audit is to check that software and its development process, both conform standards, rules and regulations.

CMM was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University in 1987.

- It is not a software process model. It is a framework which is used to analyse the approach and techniques followed by any organization to develop a software product.
- It also provides guidelines to further enhance the maturity of those software products.
- It is based on profound feedback and development practices adopted by the most successful organizations worldwide.
- This model describes a strategy that should be followed by moving through 5 different levels.
- Each level of maturity shows a process capability level. All the levels except level-1 are further described by Key Process Areas (KPA's).

Key Process Areas (KPA's):

Each of these KPA's defines the basic requirements that should be met by a software process in order to satisfy the KPA and achieve that level of maturity.

Conceptually, key process areas form the basis for management control of the software project and establish a context in which technical methods are applied, work products like models, documents, data, reports, etc. are produced, milestones are established, quality is ensured and change is properly managed.

The 5 levels of CMM are as follows:

Level-1: Initial –

- No KPA's defined.
- Processes followed are adhoc and immature and are not well defined.
- Unstable environment for software dvelopment.
- No basis for predicting product quality, time for completion, etc.

Level-2: Repeatable –

- Focuses on establishing basic project management policies.
- Experience with earlier projects is used for managing new similar natured projects.

KPA's:

- Project Planning- It includes defining resources required, goals, constraints, etc. for the project. It presents a detailed plan to be followed systematically for successful completion of a good quality software.
- Configuration Management- The focus is on maintaining the performance of the software product, including all its components, for the entire lifecycle.
- Requirements Management- It includes the management of customer reviews and feedback which result in some changes in the requirement set. It also consists of accommodation of those modified requirements.
- Subcontract Management- It focuses on the effective management of qualified software contractors i.e. it manages the parts of the software which are developed by third parties.
- Software Quality Assurance- It guarantees a good quality software product by following certain rules and quality standard guidelines while development.

Level-3: Defined –

- At this level, documentation of the standard guidelines and procedures takes place.
- It is a well defined integrated set of project specific software engineering and management processes.

KPA's:

- Peer Reviews- In this method, defects are removed by using a number of review methods like walkthroughs, inspections, buddy checks, etc.
- Intergroup Coordination- It consists of planned interactions between different development teams to ensure efficient and proper fulfilment of customer needs.
- Organization Process Definition- It's key focus is on the development and maintenance of the standard development processes.

- Organization Process Focus- It includes activities and practices that should be followed to improve the process capabilities of an organization.
- Training Programs- It focuses on the enhancement of knowledge and skills of the team members including the developers and ensuring an increase in work efficiency.

Level-4: Managed –

- At this stage, quantitative quality goals are set for the organization for software products as well as software processes.
- The measurements made help the organization to predict the product and process quality within some limits defined quantitatively.

KPA's:

- Software Quality Management- It includes the establishment of plans and strategies to develop a quantitative analysis and understanding of the product's quality.
- Quantitative Management- It focuses on controlling the project performance in a quantitative manner.

Level-5: Optimizing –

- This is the highest level of process maturity in CMM and focuses on continuous process improvement in the organization using quantitative feedback.
- Use of new tools, techniques and evaluation of software processes is done to prevent recurrence of known defects.

KPA's:

- Process Change Management- Its focus is on the continuous improvement of organization's software processes to improve productivity, quality and cycle time for the software product.
- Technology Change Management- It consists of identification and use of new technologies to improve product quality and decrease the product development time.
- Defect Prevention- It focuses on identification of causes of defects and to prevent them from recurring in future projects by improving project defined process.

The ISO 9000 Standards

ISO 9000 is a set of standards for quality assurance systems. The standards were developed by the International Organization for Standardization (ISO). First published in 1987, the standards were revised in 1994. They provide a foundation for organizations to develop or improve their quality assurance systems.

The following excerpt, adapted from Dr. Stovsky's white paper, What Every Executive Should Know about ISO 9000, describes the background and philosophy of the ISO system.

Background

A quality assurance system may be defined as the organizational structure, responsibilities, procedures, processes, and resources for implementing quality management. [1] Quality assurance systems are created to help organizations ensure their products and services satisfy customer expectations by meeting their specifications. These systems cover a wide variety of activities encompassing a product's entire life cycle including planning, controlling, measuring, testing and reporting, and improving quality levels throughout the development and manufacturing process.

Quality assurance is a management function consisting of auditing and reporting activities. These quality assurance functions help management determine the effectiveness of the quality system. ISO 9000 describes quality assurance elements in generic terms that can be applied to any business regardless of the products or services offered.

The Software Development Challenge

ISO 9001 was designed to be a generic standard, applicable to any business. As a result, it may be difficult to interpret the twenty requirements for a specific industry. In addition, 9001 has a "manufacturing" focus. That is, it contains an unstated assumption that the primary challenge faced by companies using the standard is to minimize variation in production processes so they may produce dozens, if not hundreds or thousands, of identical (or nearly so) items.

On the other hand, research and development organizations focus on design, where product design and development represent the major activities, as opposed to product reproduction. (Of course, designing products for manufacturability may be an important aspect of product design.) While the goal of the quality assurance systems for manufacturing organizations and research and development groups are similar–minimizing variation–the challenges faced when crafting an appropriate quality assurance system differ due to the nature of activities conducted in each type of organization. Computer software development has unique characteristics within research and development and engineering disciplines, further challenging the design of ISO-complaint quality assurance systems.

These software-specific differences include the intangible nature of the software product, potential complexity of the software, potential complexity in the interaction among software subsystems and software—hardware subsystems, and a unique product life cycle. To help address the additional challenges faced by software development organizations, ISO developed 9000-3, a guidance document designed to assist software development organizations seeking to create ISO-compliant quality assurance systems.

Computer software's ethereal nature creates challenges in project management and tracking. The inherent complexity of many software products creates design challenges as well as testing problems. It may be difficult to assess the "goodness" of a design. In addition, because of the huge number of possible execution thread permutations, testing a software product adequately may prove troublesome. Furthermore, because software products never wear out, their persistence can create configuration, maintenance, and support problems not encountered by other types of products.

The Role of ISO 9000-3

Because it is a guidance document, 9000-3 is not an auditable standard. Rather, it includes areas of concern that, when addressed by a software development organization, fulfill 9001's requirements. As a result, auditor's cannot issue non-conformances against the paragraphs in 9000-3. ISO 9001 remains the quality system standard–9000-3 simply represents one model for complying with the standard.

The quality assurance system model contained in 9000-3 is essentially a management model and, therefore, is technology-independent. As a result, it may be used as is regardless of the technology used in the development environment or the product hardware and software.

ISO 9000-3 provides insight into the scope of activities covered by the ISO 9001 model. It states that software is independent of the medium on which it is recorded. Consequently, 9000-3 may be applied to "hardware" development activities including firmware, gate arrays, programmed logic arrays, silicon compilers, and hardware description languages. In addition, software products include all documentation and data that are delivered to a user. So the creation of user manuals and other customer documentation is covered by 9000-3 as well. This broad definition of software conforms with the definition of the "software configuration" presented in [SEPA, 5/e, Chapter 9].

Unfortunately, the authors of 9000-3 did not organize the document so the presentation order and paragraph numbering mirrors that of 9001. Consequently, it may be difficult to relate directly paragraphs in 9000-3 to corresponding paragraphs in 9001. 9000-3 contains two annexes to help identify related clauses. However, reading both documents several times is essential to understanding how the guidance in 9000-3 relates to 9001's requirements.

The text of 9000-3 includes complete paragraphs from 9001. These paragraphs may be identified by the typeface used–they appear in italics, followed by a reference to the paragraph in 9001 from which the text was taken.

As you read 9000-3 you'll see references to specific documents such as quality plans, development plans, test plans, etc. These document names are used to suggest the nature of their contents–no specific documentation scheme is suggested. There is no requirement that your organization produce documents with these exact names or organization. As long as the topics contained in the suggested documents appear somewhere in your documentation scheme, your should not encounter a problem. For example, many organizations combine project plans, quality plans, and development plans into a single document.

Unfortunately, 9000-3 has not yet been updated to account for the 1994 revision of 9001. While the changes to 9001 were minimal, the 1994 standard represents the current set of quality assurance system requirements. If you're in doubt as to the precise nature of a requirement, always check with the most up-to-date version of 9001.

User acceptance testing (UAT), otherwise known as *Beta, Application*, or *End-User Testing*, is often considered the last phase in the web development process, the one before final installation of the software on the client site, or final distribution of it.

UAT is the usage of the software by people from the intended audience and recording and correcting of any defects which are discovered. It's the closest thing to a "_real world_" test available. It gives users the chance to interact with the software and find out if everything works as it should if features have been overlooked, miscommunicated, not communicated, and so on.

Whilst UAT – User Acceptance Testing – is essential, typically, it's not able to be undertaken until the application is largely feature-complete. Guru99 lists 10 prerequisites, which must be met before UAT can begin. These are:

1. Business Requirements must be available
2. Application Code should be fully developed
3. Unit Testing, Integration Testing & System Testing should be completed
4. No Show stoppers, or High or Medium defects in the System Integration Test Phase
5. Only Cosmetic errors are acceptable before UAT
6. Regression Testing should be completed with no major defects
7. All the reported defects should be fixed and tested
8. Traceability matrix for all testing should be completed
9. UAT Environment must be ready
10. Sign off mail or communication from System Testing Team that the system is ready for UAT execution

Types of Software Maintenance

There are four types of maintenance, namely, corrective, adaptive, perfective, and preventive. Corrective maintenance is concerned with fixing errors that are observed when the software is in use. Adaptive maintenance is concerned with the change in the software that takes place to make the software adaptable to new environment such as to run the software on a new operating system. Perfective maintenance is concerned with the change in the software that occurs while adding new functionalities in the software. Preventive maintenance involves implementing changes to prevent the occurrence of errors. The distribution of types of maintenance by type and by percentage of time consumed.

Corrective maintenance deals with the repair of faults or defects found in day-today system functions. A defect can result due to errors in software design, logic and coding. Design errors occur when changes made to the software are incorrect, incomplete, wrongly communicated, or the change request is misunderstood. Logical errors result from invalid tests and conclusions, incorrect implementation of design specifications, faulty logic flow, or incomplete test of data. All these errors, referred to as residual errors, prevent the software from conforming to its agreed specifications. Note that the need for corrective maintenance is usually initiated by bug reports drawn by the users.

Types of Software Maintenance

In the event of a system failure due to an error, actions are taken to restore the operation of the software system. The approach in corrective maintenance is to locate the original specifications in order to determine what the system was originally designed to do. However, due to pressure from management, the maintenance team sometimes resorts to emergency fixes known as patching. Corrective maintenance accounts for 20% of all the maintenance activities.

Adaptive Maintenance

Adaptive maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns, and government policies have a significant impact on the software system.

For instance, a government policy to use a single 'European currency' will have a significant effect on the software system. An acceptance of this change will require banks in

various member countries to make significant changes in their software systems to accommodate this currency. Adaptive maintenance accounts for 25% of all the maintenance activities.

Perfective Maintenance

Perfective maintenance mainly deals with implementing new or changed user requirements. Perfective maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users' changing needs.

Examples of perfective maintenance include modifying the payroll program to incorporate a new union settlement and adding a new report in the sales analysis system. Perfective maintenance accounts for 50%, that is, the largest of all the maintenance activities.

Preventive Maintenance

Preventive maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understandability and increasing software maintainability. It comprises documentation updating, code optimization, and code restructuring. Documentation updating involves modifying the documents affected by the changes in order to correspond to the present state of the system. Code optimization involves modifying the programs for faster execution or efficient use of storage space. Code restructuring involves transforming the program structure for reducing the complexity in source code and making it easier to understand.

Preventive maintenance is limited to the maintenance organization only and no external requests are acquired for this type of maintenance. Preventive maintenance accounts for only 5% of all the maintenance activities.

Parallel running is a strategy for system implementation where a new system slowly assumes the roles of the older system while both systems operate simultaneously.This conversion takes place as the technology of the old system is outdated so a new system is needed to be installed to replace the old one. After a period of time, when the system is proved to be working correctly, the old system will be removed completely and users will depend solely on the new system. The

phrase parallel running can refer to the process of changing a fragment of business information technology operation to a new system or to the technique applied by the human resources departments in which the existing staff stay on board during the transition to a new staff

Configuration management (**CM**) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. The CM process is widely used by military engineering organizations to manage changes throughout the system lifecycle of complex systems, such as weapon systems, military vehicles, and information systems. Outside the military, the CM process is also used with IT service management as defined by ITIL, and with other domain models in the civil engineering and other industrial engineeringsegments such as roads, bridges, canals, dams, and buildings.

Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. The documentation either explains how the software operates or how to use it, and may mean different things to people in different roles.

Documentation is an important part of software engineering. Types of documentation include:

- Requirements – Statements that identify attributes, capabilities, characteristics, or qualities of a system. This is the foundation for what will be or has been implemented.
- Architecture/Design – Overview of software. Includes relations to an environment and construction principles to be used in design of software components.
- Technical – Documentation of code, algorithms, interfaces, and APIs.
- End user – Manuals for the end-user, system administrators and support staff.
- Marketing – How to market the product and analysis of the market demand.

KARPAGAM ACADEMY OF HIGHER EDUCATION- COIMBATORE 21

SUBJECT CODE:17MBAPS401B SUBJECT:SOFTWARE PROJECT MANAGEMENT

	Unit 3						
S.NO	**Question**	**option 1**	**option2**	**option3**	**option4**		**Answer**
1	The inspection is lead by a _____who has had specific training in the technique	supervisor	team leader	project leader	team member		supervisor
2	_____carries out testing in clean room software development.	certification team	testing team	development team	team member.		certification team
3	In software quality, _____is an attempt made to quantify the presence of the quality	binary measure	relative quantity measure	binary quantity measure	relative measure		relative quantity measure
4	_____are the persons who assess the quality of a software product, not for themselves but for a community of users	acquirers.	developers.	independent evaluators	testers.		independent evaluators
5	The _____is carried out to calculate the earliest date on which each event may be achieved and the earliest dates on which each activity may be started and completed.	top down estimation	bottom up estimation	forward pass	backward pass.		forward pass
6	The rule of _____states that the latest date for an event is the latest start date for all the activities that may commence from that event	backward pass.	forward pass	CPM network.	network analysis.		backward pass.
7	PERT stand for _____.	program evaluation and review technique	process evaluation and review technique	product evaluation and review technique	program evaluation and reporting technique		program evaluation and review technique
8	The first stage in setting up an earned value analysis is to create the _____.	budgeted workdays	schedule completion.	baseline budget	calculate cumulative workdays		baseline budget
9	The _____is the process of making the new system operational	qualification testing	integration	installation	architecture design		installation
10	A record on how much the organization is willing to spend on the system is called as ____.	functional requirements	quality requirements	resource requirements	functional specification		resource requirements
11	The _____ phase of the project investigates whether starting the project is worth	planning	feasibility study.	project execution.	project analysis.		feasibility study
12	The _____method of project planning covers only the planning stages of a project and not monitoring and control	PRINCE	PRINCE2	ISO 12207.	step wise project planning		step wise project planning
13	The _____is the time taken to break even or pay back the initial investment	pay back period	return on investment	annual rate of return	net present value		pay back period
14	The _____attempts to provide a profitability measure as a percentage of return that is directly comparable with interest rates.	internal Rate of return	positive NPV.	negative NPV	present value		internal Rate of return
15	A _____ is a working model of one or more aspects of the projected system which is constructed and tested quickly and inexpensively in order to test the assumptions	prototype.	software	trial product	stereotype		trial product
16	The _____ is developed and modified until it is finally in a state where it becomes the operational system	software prototype.	throw away prototype	software working model	evolutionary prototype		evolutionary prototype

17	The ____ changes affect more than one part of the processing and must have to be done with the subject of a design review before they can be implemented.	local	global	cosmetic	partial		global
18	The _____can be used to establish the order in which increments are to be developed.	cost to value ratio	value to value ratio	cost to cost ratio	value to cost ratio		value to cost ratio
19	Work expands to fill the time available is _____.	Weinbergs Zeroth Law of reliability	Brooks law	Parkinsons Law	Law of estimation		Parkinsons Law
20	SLOC stands for _____.	source lines of coding	simple lines of coding	software lines of coding	starting lines of code.		source lines of coding
21	The _____are based on an estimate of the functionality of the delivered software	size related measures	function related measures	software related measures	estimation related measures		function related measures
22	Which one of the following statements about the project risk register is false?	It facilitates the review and monitoring of risks	It facilitates the risk appetite	It facilitates the recording of risk responses.	It facilitates the recording of risks		It facilitates the risk appetite
23	_____ is the process of setting goals, developing strategies, and outlining tasks and schedules to accomplish the goals	panning	activity	playing	reading		planning
24	The _____is a personality in project team who adept at finding resources in the terms of both physical resources and information.	complete investigator	resource investigator.	complete finisher	company worker		resource investigator.
25	The _____refers to the way that once you have made a decision it is difficult to alter, even it is wrong.	information overload.	faulty heuristics	escalation of commitment	decision fault		escalation of commitment
26	PRINCE2 is a _____for project management providing an easily tailored and scalable method for the management of all types of projects	process based approach	product based approach.	hybrid approach	work based approach		process based approach
27	COCOMO stands for	Constructive Cost Model	Communication Connection Method	Collective Cost Model	Credit Card Mode.		Constructive Cost Model
28	_____of the following is negative possibility if quality requirements are not met.	Overworking team members and poor product quality.	Customer complaints and late product	Poor quality	Termination.		Overworking team members and poor product quality.
29	Expand SSAD	Structured System Analysis and design	Software system analysis and design	System Security Adhoc design.	System Security analysis and design.		Structured System Analysis and design
30	____ is a term that encompasses all forms of technology used to create, store, exchange, and use information in its various forms.	Computer Technology.	Network Technology.	Information Technology	Client Server Technology		Information Technology
31	A communication management plan identifies the relevant information that should be communicated to	the project team.	the project stakeholders	the project board	the project sponsor.		the project stakeholders
32	The most creative and challenging phase of the system life cycle is	system design	system analysis	system cycle	all the above		system design
33	User training was	poor	best	fast	medium		poor
34	Which one of the following best describes a project issue?	A major problem that requires formal escalation.	A problem that the project manager has to deal with on a day-to-day basis	An uncertain event that may or may not occur.	An opportunity that occurs through change control.		A major problem that requires formal escalation.

35	Scheduling can best be defined as the process used to determine	overall project duration.	project cost estimating	the project management plan	sub-contractor's responsibilities		overall project duration.
36	Which one of the following statements is true?	An increase in project scope is likely to increase project cost	A decrease in the project time is likely to increase project quality	An increase in the project quality requirements is likely to decrease project cost.	A decrease in the project cost is likely to decrease project time		An increase in project scope is likely to increase project cost
37	Which one of the following statements best defines the purpose of a Product Breakdown Structure (PBS)?	To define the hierarchy of deliverables that are required to be produced on the project	To define how the products are produced by identifying derivations and dependencies.	To establish the extent of work required prior to project commissioning and the handover	To identify the health and safety strategies and procedures to be used on the project		To define the hierarchy of deliverables that are required to be produced on the project
38	Which one of the following is least likely to be a success criteria?	A target for the project to receive zero change requests	The date by which the project is to be completed	Delivery of products that meet required specifications.	The awarding of bonuses to senior management		The awarding of bonuses to senior management
39	Which one of the following is a valid project Key Performance Indicator (KPI)?	Staff appraisals.	Management buy in.	Milestone achievement	Master schedule		Milestone achievement
40	A word of caution regarding activities?	prototype cycle	lifecycle	candidate cycle	ta		lifecycle
41	Who owns the Project Management Plan (PMP)?	The project team.	The chief executive	The project manager.	The project support office		The project manager.
42	Which one of the following best describes users?	Providers of both strategic and tactical direction to the project	Those intended to receive benefits or operate outputs	Facilitators of an appropriate issue resolution procedure.	Those providing full-time commitment to the project.		Those intended to receive benefits or operate outputs.
43	Which statement best describes a responsibility of the project manager	to be the sole source of expertise for estimating techniques on cost and time	to deliver the project objectives to enable benefits to be realised	to take ultimate accountability for the delivery of the business benefits	to delegate all accountability for managing time, cost and quality to team leaders.		to deliver the project objectives to enable benefits to be realised.
44	A project is typically defined in terms of scope, time, cost and which other parameter?	BENEFITS	QUALITY	Tolerance.	CONTROLS		QUALITY
45	Which one of the following statements is true?	Business-as-usual activities cannot be improved.	Business-as-usual activities are more difficult to manage than projects	Projects are transient endeavours that bring about change to business-as-usual	A project is always the starting point for operation refinement		Projects are transient endeavours that bring about change to business-as-usual.
46	What is defined as "the ability to influence and align others towards a common purpose"?	teamwork	Motivation.	management	leadership		leadership
47	Which one is a true statement relating to project communications?	A project sponsor is responsible for all communication methods and media.	Different stakeholders typically have different communication needs	It is best to have a standard set of project reports used for every project	Email is the only way to communicate with large numbers of people.		Different stakeholders typically have different communication needs.
48	In project management, the term quality is best defined as:	inspection, testing and measurement.	reviews and audits	fitness for purpose of deliverables	professionally-bound project reports.		fitness for purpose of deliverables

49	The main aim of a project risk management process should be to:	identify project risks and then manage them appropriately	identify all project risks and transfer them immediately	identify all the things that are threats or opportunities on a project.	satisfy the organisation's project management process		identify project risks and then manage them appropriately.
50	Which one of the following is not considered in resource management?	Identifying resources.	Influencing resources	Assigning resources to activities	Matching resources to the schedule		Influencing resources.
51	Which one of the following does project change control primarily seek to ensure?	All variance to the project scope is evaluated.	No reduction in the perceived quality of the project outcome	Management costs of the project do not increase.	Any decrease in the scoped deliverable of the project is rejected.		All variance to the project scope is evaluated
52	Which one of the following is captured in the Work Breakdown Structure (WBS)?	The life cycle phases	The logical order of tasks	The scope of the project	Project costs.		The scope of the project.
53	Project reporting can best be defined as:	informing stakeholders about the project	storing and archiving of project information	gathering stakeholder feedback	collecting project information.		informing stakeholders about the project
54	Which one of the following statements best defines an estimate?	An approximation of project time and cost targets, refined throughout the project life cycle.	A prediction of a future condition or event based on information or knowledge available now.	The value of useful work done at any given point in a project to give a measure of progress.	A situation that affects or influences the outcome of the project expressed in time or cost terms.		An approximation of project time and cost targets, refined throughout the project life cycle.
55	The justification for the investment to be made in a project is documented in the:	Cost Breakdown Structure	procurement strategy.	business case.	Project Management Plan.		business case.
56	Which one of the following is a responsibility of the project steering group/board?	To identify potential problems for the project team to solve.	To provide strategic direction and guidance to the sponsor.	To manage the project team in all daily activities.	To receive and consider daily reports from team members.		To provide strategic direction and guidance to the sponsor.
57	One of the reasons a project life cycle is split into phases is to:	facilitate formal go/no-go decision making during the project.	balance the costs of work in each phase of project development.	mirror the major deployments of resources throughout the project.	chunk work into time periods of similar durations.		facilitate formal go/no-go decision making during the project.
58	Which of the following best describes a project environment?	The type of organisation concerned with implementation	The structured method used to control the project.	The context within which a project is undertaken.	An understanding of the risks involved in the project		The context within which a project is undertaken.
59	Which one of the following statements best describes a project?	A project is a set of tools and techniques often used when delivering organisational change	A project is the sum of activities needed to remove uncertainty from a unique piece of work.	A unique transient endeavour undertaken to achieve a desired outcome	A project is a method of planning work.		A unique transient endeavour undertaken to achieve a desired outcome.
60	The document that identifies what information needs to be shared, to whom, why, when and how is called the:	communication management plan	stakeholder mapping grid.	document distribution schedule	responsibility assignment matrix.		communication management plan.

Unit – IV

Acquisition process - Hardware, software, Network, Infrastructure - Requirement planning- sizing - selection Methodology including Benchmarking - Documents involved IT HRM – Selection – Retention – Training – Career path planning – IT operations – Scheduling – Performance Evaluation.

Selecting hardware and software for implementing information system in an organization is a serious and time-consuming process that passes through several phases. The main steps of the selection process are listed below:

1. Requirement analysis: - System configuration requirements are clearly identified and a decision to acquire the system is taken in this step.
2. Preparation of tender specifications: - After studying the feasibility and deciding upon the configuration, tender documents are prepared for the benefit of vendors to clarify the details of various specifications, as listed below.

I) Purchase procedure and schedule: it includes

a) Date of tender submission

b) Evaluation criteria

c) Scope for negotiations, if any and

d) Expected usage environment and load pattern

ii) Equipment specification

Detailed technical specifications of each item required for both mandatory and optional items.

II) Quotation format:

a) Format for stating technical details and quoting prices
b) Whether deviations from specifications should be specifically listed
c) Prices and levies (duties, taxes etc.) could be quoted as lumpsum or required separately.
d) Required validity of the quotation.
e) Earnest money deposit required, if any.

III) Proposed terms of contract
a) Expected delivery schedule.
b) Uptime warranties required
c) Penalty clause, if any
d) Payment terms (Whether advance payment acceptable)
e) Arbitrary clauses
f) Training needs.
g) Post warranty maintenance terms expected.
v) Any additional information required.

3. Inviting tenders: - After the preparation of tender specifications, tenders are invited. Invitation of tenders may depend upon the magnitude of purchase (estimate equipment cost). It may be through

i) Open tender (through newspaper advertisement)
ii) Limited tender (queries sent to a few selected vendors)
iii) Propriety purchase (applies mostly to upgrade requirements)
iv) Direct purchase from market. (applies mostly to consumables)

4. Technical scrutiny and short listing: - This step involves the following activities.

i) All tendered bids are opened on a pre-defined date and time.
ii) Deviations from the specifications, if any, in each bid are noted.

iii) A comparative summery is prepared against the list of tendered technical features.

Additional factors to considered are:

i) Financial health of the vendor
(from balance sheets)
ii) Nature and extent of support
(from information provided on number of support staff per installed site an cross-check with selected customers)
iii) Engineering quality pf products
(factory inspection of product facilities, QA procedures and R&D)

5. Detailed evaluation of short listed vendors: - This step primarily involves getting any finer technical clarifications. Visits to customer sites and factory inspections may be planned. If any specific performance requirement is stipulated, the offered product is to be examined at this stage through suitable benchmark tests. For benchmark tests, standard benchmarks may be used as adequate performance indicators.

6. Negotiation and procurement decision: - Because of the extensive competition, computer system vendors may offer significant concessions. Negotiations are held to maximize these concessions. However, price negotiations are often not permitted by some organizations.

When price negotiations are permitted, the committee members should have a good knowledge of the prevailing market prices, current trends, and also the duty/tax structure.

i) Computer magazines
ii) Vendor directories.
iii) Contact with other users

iv) Past personal experience.

7. Delivery and installation: - In this step, the vendor delivers the hardware/software to the buyer's organization, where it is matched with the specifications mentioned in the purchase order. If conforms to these specifications, the vendor installs the system in the premises of the organization.

8. Post-installation review: - After the system is installed, a system evaluation is made to determine how closely the new system conforms to the plan. A post-installation review, in which system specifications and user requirements are audited, is made. The feedback obtained in this step helps in taking corrective decision.

- **Feasibility Study**
- **Requirement Gathering**
- **Software Requirement Specification**
- **Software Requirement Validation**

Let us see the process briefly -

Feasibility study

When the client approaches the organization for getting the desired product developed, it comes up with rough idea about what all functions the software must perform and which all features are expected from the software.

Referencing to this information, the analysts does a detailed study about whether the desired system and its functionality are feasible to develop.

This feasibility study is focused towards goal of the organization. This study analyzes whether the software product can be practically materialized in terms of implementation, contribution of project to organization, cost constraints and as per values and objectives of the organization. It explores technical aspects of the project and product such as usability, maintainability, productivity and integration ability.

The output of this phase should be a feasibility study report that should contain adequate comments and recommendations for management about whether or not the project should be undertaken.

Requirement Gathering

If the feasibility report is positive towards undertaking the project, next phase starts with gathering requirements from the user. Analysts and engineers communicate with the client and end-users to know their ideas on what the software should provide and which features they want the software to include.

Software Requirement Specification

SRS is a document created by system analyst after the requirements are collected from various stakeholders.

SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

SRS should come up with following features:

- User Requirements are expressed in natural language.
- Technical requirements are expressed in structured language, which is used inside the organization.
- Design description should be written in Pseudo code.
- Format of Forms and GUI screen prints.
- Conditional and mathematical notations for DFDs etc.

Software Requirement Validation

After requirement specifications are developed, the requirements mentioned in this document are validated. User might ask for illegal, impractical solution or experts may interpret the

requirements incorrectly. This results in huge increase in cost if not nipped in the bud. Requirements can be checked against following conditions -

- If they can be practically implemented
- If they are valid and as per functionality and domain of software
- If there are any ambiguities
- If they are complete
- If they can be demonstrated

Requirement Elicitation Process

Requirement elicitation process can be depicted using the folloiwng diagram:

- **Requirements gathering -** The developers discuss with the client and end users and know their expectations from the software.
- **Organizing Requirements -** The developers prioritize and arrange the requirements in order of importance, urgency and convenience.
- **Negotiation & discussion -** If requirements are ambiguous or there are some conflicts in requirements of various stakeholders, if they are, it is then negotiated and discussed with stakeholders. Requirements may then be prioritized and reasonably compromised.

  The requirements come from various stakeholders. To remove the ambiguity and conflicts, they are discussed for clarity and correctness. Unrealistic requirements are compromised reasonably.

- **Documentation -** All formal & informal, functional and non-functional requirements are documented and made available for next phase processing.

Requirement Elicitation Techniques

Requirements Elicitation is the process to find out the requirements for an intended software system by communicating with client, end users, system users and others who have a stake in the software system development.

There are various ways to discover requirements

Interviews

Interviews are strong medium to collect requirements. Organization may conduct several types of interviews such as:

- Structured (closed) interviews, where every single information to gather is decided in advance, they follow pattern and matter of discussion firmly.
- Non-structured (open) interviews, where information to gather is not decided in advance, more flexible and less biased.
- Oral interviews
- Written interviews
- One-to-one interviews which are held between two persons across the table.
- Group interviews which are held between groups of participants. They help to uncover any missing requirement as numerous people are involved.

Surveys

Organization may conduct surveys among various stakeholders by querying about their expectation and requirements from the upcoming system.

Questionnaires

A document with pre-defined set of objective questions and respective options is handed over to all stakeholders to answer, which are collected and compiled.

A shortcoming of this technique is, if an option for some issue is not mentioned in the questionnaire, the issue might be left unattended.

Task analysis

Team of engineers and developers may analyze the operation for which the new system is required. If the client already has some software to perform certain operation, it is studied and requirements of proposed system are collected.

Domain Analysis

Every software falls into some domain category. The expert people in the domain can be a great help to analyze general and specific requirements.

Brainstorming

An informal debate is held among various stakeholders and all their inputs are recorded for further requirements analysis.

Prototyping

Prototyping is building user interface without adding detail functionality for user to interpret the features of intended software product. It helps giving better idea of requirements. If there is no software installed at client's end for developer's reference and the client is not aware of its own requirements, the developer creates a prototype based on initially mentioned requirements. The prototype is shown to the client and the feedback is noted. The client feedback serves as an input for requirement gathering.

Observation

Team of experts visit the client's organization or workplace. They observe the actual working of the existing installed systems. They observe the workflow at client's end and how execution problems are dealt. The team itself draws some conclusions which aid to form requirements expected from the software.

Software Requirements Characteristics

Gathering software requirements is the foundation of the entire software development project. Hence they must be clear, correct and well-defined.

A complete Software Requirement Specifications must be:

- Clear
- Correct
- Consistent
- Coherent
- Comprehensible
- Modifiable
- Verifiable
- Prioritized
- Unambiguous

- Traceable
- Credible source

Software Requirements

We should try to understand what sort of requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system.

Broadly software requirements should be categorized in two categories:

Functional Requirements

Requirements, which are related to functional aspect of software fall into this category.

They define functions and functionality within and from the software system.

EXAMPLES -

- Search option given to user to search from various invoices.
- User should be able to mail any report to management.
- Users can be divided into groups and groups can be given separate rights.
- Should comply business rules and administrative functions.
- Software is developed keeping downward compatibility intact.

Non-Functional Requirements

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

Non-functional requirements include -

- Security
- Logging
- Storage
- Configuration
- Performance
- Cost
- Interoperability

- Flexibility
- Disaster recovery
- Accessibility

Requirements are categorized logically as

- **Must Have** : Software cannot be said operational without them.
- **Should have** : Enhancing the functionality of software.
- **Could have** : Software can still properly function with these requirements.
- **Wish list** : These requirements do not map to any objectives of software.

While developing software, 'Must have' must be implemented, 'Should have' is a matter of debate with stakeholders and negation, whereas 'could have' and 'wish list' can be kept for software updates.

User Interface requirements

UI is an important part of any software or hardware or hybrid system. A software is widely accepted if it is -

- easy to operate
- quick in response
- effectively handling operational errors
- providing simple yet consistent user interface

User acceptance majorly depends upon how user can use the software. UI is the only way for users to perceive the system. A well performing software system must also be equipped with attractive, clear, consistent and responsive user interface. Otherwise the functionalities of software system can not be used in convenient way. A system is said be good if it provides means to use it efficiently. User interface requirements are briefly mentioned below -

- Content presentation
- Easy Navigation
- Simple interface
- Responsive

- Consistent UI elements
- Feedback mechanism
- Default settings
- Purposeful layout
- Strategical use of color and texture.
- Provide help information
- User centric approach
- Group based view settings.

Software System Analyst

System analyst in an IT organization is a person, who analyzes the requirement of proposed system and ensures that requirements are conceived and documented properly & correctly. Role of an analyst starts during Software Analysis Phase of SDLC. It is the responsibility of analyst to make sure that the developed software meets the requirements of the client.

System Analysts have the following responsibilities:

- Analyzing and understanding requirements of intended software
- Understanding how the project will contribute in the organization objectives
- Identify sources of requirement
- Validation of requirement
- Develop and implement requirement management plan
- Documentation of business, technical, process and product requirements
- Coordination with clients to prioritize requirements and remove and ambiguity
- Finalizing acceptance criteria with client and other stakeholders

Software Metrics and Measures

Software Measures can be understood as a process of quantifying and symbolizing various attributes and aspects of software.

Software Metrics provide measures for various aspects of software process and software product.

Software measures are fundamental requirement of software engineering. They not only help to control the software development process but also aid to keep quality of ultimate product excellent.

According to Tom DeMarco, a (Software Engineer), "You cannot control what you cannot measure." By his saying, it is very clear how important software measures are.

Let us see some software metrics:

- **Size Metrics -** LOC (Lines of Code), mostly calculated in thousands of delivered source code lines, denoted as KLOC.

  Function Point Count is measure of the functionality provided by the software. Function Point count defines the size of functional aspect of software.

- **Complexity Metrics -** McCabe's Cyclomatic complexity quantifies the upper bound of the number of independent paths in a program, which is perceived as complexity of the program or its modules. It is represented in terms of graph theory concepts by using control flow graph.
- **Quality Metrics -** Defects, their types and causes, consequence, intensity of severity and their implications define the quality of product.

  The number of defects found in development process and number of defects reported by the client after the product is installed or delivered at client-end, define quality of product.

- **Process Metrics -** In various phases of SDLC, the methods and tools used, the company standards and the performance of development are software process metrics.
- **Resource Metrics -** Effort, time and various resources used, represents metrics for resource measurement

What is software benchmarking?

Software benchmarking is the collection of and comparison of data from multiple sources. This process doesn't necessarily have to do with software, and recognizing that helps you build a

formal plan for software benchmarking. On the other hand, you might be benchmarking the software development process itself to determine if teams are working as efficiently as possible.

Consider the situation of comparing insurance rates. You might make a few phone calls, visit a few websites, and gather up the data but before you can compare the data, you need to normalize it to make sure you're not comparing apples to oranges. Are you comparing monthly rates to bi-annual rates? Are the rates for the same services? Similarly, with benchmarking software development processes, you have to make sure your data is normalized. To get the data normalized, you need to use a formal process, just as you do with software quality in general. In a sense, designing a software benchmark is similar to architecting software itself.

software benchmarking process as:

1. Needs assessment: What are you trying to accomplish? Create the questions you need answered, and determine if those questions cover everything, you need to know. Are you factoring in cost? Often software developers think of software benchmarking strictly in terms of time, i.e. which program runs faster. But benchmarking is much broader and includes costs, quality, and so on.
2. Industry/Domain Classification. You need to determine what type of benchmarking you need based on the industry and domain. From there you can determine whether you can use standard benchmarking tests developed by experts, or if you need to develop your own, specific benchmarking tests.
3. Data Collection. This step is, of course, vital, yet it's easy to take it too lightly. In the previous steps, you determined what data you need, in this step, we cover not just the data collection itself, but how you're going to collect it. Some data might be collected automatically by software; other data might be collected through surveys from users. The data collection process might take only minutes; or, it might take months, depending on your needs. In either case, you'll want to survey the data you're collecting to make sure it's consistent and is providing the information you need.

4. Data Normalization and Purification. As the data comes in, you need to make sure it's normalized. A trivial example is that if some data is in US dollars and other data is in Euros, you need to normalize it to a single currency. You also need to remove data that might be inaccurate or could improperly skew the results. This, of course, will likely involve standard statistical models and methods.

5. Data Comparison and Benchmark Preparation. After the data is collected, you have your benchmarks. More analysis takes place here, however. If the data seems incorrect, then perhaps it is. Do some more digging and find out why, and if the tests need to be performed again.

6. Benchmark Reporting. The process of reporting the data is also not trivial. The results need to be presented in an accurate but usable fashion target at a specific audience. Business managers would need different information than technical managers.

7. Improvement Recommendations. As with any process, your benchmarking process will likely have room for improvement. Are you gathering the necessary data? Are you packaging the results into reports that are useful?

IT in HRM

1. Human Resource Planning: With the help of innovation construct databases, voluminous information about the employees can be stored, which not just aides in distinguishing the involved and vacant positions, additionally it also helps determining if the individual is the best fit or not.

2. Administration: All the basic data identified with the workforce, like their name, address, email, contact no., capability, compensation benefits, encounter, date of passage in organizations, employment status (contract, perpetual, full-time, low maintenance, and so on), are incorporated in a database that can be recovered at any time.

3. Recruitment: The web has brought on the biggest change to the enrollment procedure in the previous decade, as it connects the companies and the job seekers.

4. Training and Development: E-learning is a progressive approach to enable the workforce to

keep pace with a quickly evolving market. By connecting the evaluation process to the HR database, the e-learning framework can be used effectively.

5. Compensation and Benefits: The e-pay bundles offer straightforward, simple, precise and assessable data on the compensation structure of the employees.

Opportunities for implementing IT with HR function

1. Competitive Advantage: Giving customized applications through HRM portals implies that e-HRM can be a key technique in innovation.

2. Accessibility: Data is accessible to everyone, through web or intranet. Any employee can get any information effortlessly HR entryways permit the representatives to get to all the required data at a transgression click.

3. Rapid and Mistake-free exchanges: Technological innovations have expanded the pace of administration in organizations. Mechanical frameworks eliminate human errors.

4. Interactive Atmosphere: Technology enhances interactions among the representatives through the electronic gateways. Bigger organizations have more data needs, and they can take more points of interest from these data. With mid-size organizations, it enables data spread over various structures and locations.

Challenges associated with HR technology:

1. Fetched: "Technology pulls cost". An innovation-based HR framework is expensive, but once executed, it decreases the operational expenses. Substantial organizations may introduce HR gateways/bundles, while small- to mid-size organizations find it difficult to bear the cost.

2. Acknowledgment: Because of IT usage, different issues like skills/knowledge for its utilization, job dangers and so on dependably ascend in its direction. Acknowledgment from the workforce is required for using it up to its fullest.

3. Back-ups and Security: Maintenance cost is high if we need to prevent hacking/open to all arrangement/illegal acts. A lot of thought is required on these lines.

4. Increasing Isolation: Due to the arrangement of virtual networks through intranet or eHR gateways, the individual collaboration among the representatives has reduced. In the traditional frameworks, they collaborate with the representatives, and were integral to the organization. They are disengaged from each other now, and are connected for all intents and purposes through such entryways only.

Types of Performance evaluations:

Annual Evaluations — The Annual evaluation covers the calendar year (January 1 through December 31) and is the evaluation period used for all staff who are not serving on a probationary period. The annual evaluations are due on March 1

Probationary Evaluations — A probationary evaluation covers the probationary period when an employee begins a new position. The probationary period is typically 6 months long, but can be extended by Human Resource Management (HRM). Please refer to the University's Probationary Period Policy. Probationary evaluations are due prior to the employee's probationary end date.

Special Evaluations — A Special Evaluation covers a defined period of time as established by the supervisor in conjunction with HRM. An employee can be placed on a special evaluation at any time. An employee who is placed on a special evaluation will be notified of the duration and due date at the time a Special Evaluation period begins.

Objectives

The objectives of the performance evaluation process for Staff Performance Evaluations are to:

- Provide employees with feedback to improve or maintain job performance
- Identify areas for employee development

- Set performance standards and goals for the next review period
- Recognize job-related accomplishments
- Enhance communication and working relationships

KARPAGAM ACADEMY OF HIGHER ECUATION - COIMBATORE 21

SUBJECT CODE: 17MBAPS401B SUBJECT: SOFTWARE PROJECT MANAGEMENT

	Unit 4						
S.NO	**question**	**option1**	**option2**	**option3**	**option4**		**Answer**
1	An important aim of a post-project review is to:	validate overall progress to date against the budget and schedule.	capture learning and document it for future usage	ensure acceptance of all permanent documentation, signed by the sponsor	establish that project benefits have been identified.		capture learning and document it for future usage.
2	The process that evaluates overall project performance to provide confidence is called	quality assurance	quality planning	quality control	quality audit.		quality assurance
3	Need identification and analysis are concerned	user	output	inputs	all the above		user
4	Which one of the following statements best defines procurement?	A technique to establish the best approach for obtaining the resources for the project	A group of interrelated resources and activities that transform inputs into outputs	The description of the purpose, form and components to support delivery of a product	The process by which products and services required for the project are acquired		The process by which products and services required for the project are acquired
5	Once a change has been requested what is the next step in the change control process?	Evaluate the change.	Advise the sponsor.	Update the change log.	Update the project plan		Evaluate the change
6	A Responsibility Assignment Matrix (RAM) can be used to:	define the terms of reference of the project manager	define the limits of the project sponsor's responsibilities.	allocate risk management response activities to project personnel	allocate work packages to those responsible for project work		allocate work packages to those responsible for project work
7	An Organisational Breakdown Structure (OBS) is used to identify:	the reporting structure and current availability of all individuals in the project.	technical ability and line of communication for all individuals in the project	lines of communication and responsibility for all the individual managers in the project	the reporting structure and lines of communication for all individuals in the projects		the reporting structure and lines of communication for all individuals in the projects
8	Which one of the following best describes project success criteria?	Actively seeking some senior management support.	Measures by which the success of the project is judged.	Achievement of milestones	A motivated project team		Measures by which the success of the project is judged.
9	Comparative estimating uses	current data from similar projects	historic data from all projects.	historic data from similar projects	current data from all projects		historic data from similar projects.
10	Which one of the following best describes a project stakeholder?	A party who is concerned about the project going ahead.	A party with an interest or role in the project or is impacted by the project.	A party who has a vested interest in the outcome of the project.	A party who has a financial stake in the organisation managing the project		A party with an interest or role in the project or is impacted by the project.
11	The main purpose of the Project Management Plan is to:	provide justification for undertaking the project in terms of evaluating the	ensure the project sponsor has tight control of the project manager's	document the outcomes of the planning process and provide the reference	document the outcome of the risk, change and configuration		document the outcomes of the planning process and provide the
12	Who has ultimate responsibility for project risk?	Steering group.	Risk owner.	Project sponsor.	Project manager		Project sponsor
13	When a project has completed the handover and closure phase	the project deliverables are ready for commissioning	the project deliverables are ready for handing over to the users	the project documentation must be disposed of.	the capability is now in place for the benefits to be realised.		the capability is now in place for the benefits to be realised.
14	Which one of the following illustrates why effective project management is beneficial to an organisation?	It utilises resources as and when required under direction of a project manager	It advocates employing a consultancy firm which specialises in managing change	It recommends using only highly skilled people in the project team.	It ensures that the chief executive is accountable for the achievement of the defined benefits		It utilises resources as and when required under direction of a project manager.
15	A key aspect of managing a project involves	defining which operational systems to put in place	identifying routine tasks	ensuring ongoing operations are maintained	planning to achieve defined objectives		planning to achieve defined objectives
16	Which one of the following statements best defines teamwork?	People working collaboratively towards a common goal.	Developing skills that will enhance project performance.	Gathering the right people together to work on a project.	Establishing vision and direction towards a common purpose		People working collaboratively towards a common goal.
17	A review undertaken to decide whether a project should proceed into its next phase is known as a:	gate review.	feasibility study.	milestone review	`evaluation review.		gate review.
18	Which one of the following statements best describes the use of an issue log?	A summary of all possible alternative resolutions of an issue.	A summary of all the project issues, their analysis and status.	A tool to ensure that a process is in place for capturing all issues.	A tool to ensure that the issue management process is adhered to.		A summary of all the project issues, their analysis and status.
19	……is a data-flow based methodology	structured design	structured system	structured tools	structured systems		structured tools
20	What is a visual representation of a project's planned activities against a calendar called?	A Gantt chart.	A critical path network	A product flow diagram	A Pareto chart.		A Gantt chart.
21	Configuration management is best described as:	control in the implementation of changes to project schedules	an organisation to review proposed changes to the project deliverables	quality control of project deliverables and documentation	creation, maintenance and controlled change of the project deliverables		creation, maintenance and controlled change of the project deliverables
22	A Cost Breakdown Structure (CBS) shows costs assigned to:	individual work packages using the Work Breakdown Structure (WBS).	individual resources using the Work Breakdown Structure (WBS).	individual resources using the Responsibility Assignment Matrix (RAM).	individual deliverables using the Responsibility Assignment Matrix (RAM		individual work packages using the Work Breakdown Structure (WBS).

23	The accuracy of an estimate should:	decrease as a project progresses through its life cycle.	increase as a project progresses through its life cycle.	stay constant throughout the project life cycle.	vary independently of where the project is in its life cycle		increase as a project progresses through its life cycle.
24	Which one of the following best defines a benefit?	A positive result of stakeholder management	The successful management of a project.	An improvement resulting from project deliverables	The successful delivery of project reports and updates.		An improvement resulting from project deliverables.
25	The objective is to build a new document ,called	system planning	system control	system development	system specifications		system development
26	Who are project team members primarily accountable to?	External stakeholders	The end users	The finance director	The project manager		The project manager
27	The phases of a project life cycle are:	starting, planning, control and closing.	concept, definition, development, handover and closure	initiation, definition, planning, monitoring and operations	initiation, definition, planning, monitoring and operations		concept, definition, development, handover and closure
28	A portfolio can best be defined as:	a group of projects and programmes carried out within an organisation	a group of programmes carried out under the sponsorship of an organisation	a group of projects carried out under the sponsorship of an organisation.	a range of products and services offered by an organisation.		a group of projects and programmes carried out within an organisation
29	Which one of the following best describes project management?	Using APM's Body of Knowledge 6th edition as a guide to all projects	Employing a project manager who has undertaken similar projects	Utilising team members who can work on a project full time.	Application of processes and methods throughout the project life cycle.		Application of processes and methods throughout the project life cycle.
30	Which structure shows the reporting relationships and communications channels for a project?	Work Breakdown Structure.	Organisational Breakdown Structure.	Product Breakdown Structure.	Responsibility assignment structure.		Organisational Breakdown Structure
31	1.Identify, from among the following, thecorrectstatement.	One of the main challenges Software Engineering facing today is the requirement of most software systems to work with a multitude of homogenous systems	'Legacy systems' are custom developed software systems for the legal domain	Software does not wear-out in the traditional sense of the term, but software does tend to deteriorate as it evolves	With the advent of component based software assembly, we find that only less than 20% of today's software is still custom built.		Software does not wear-out in the traditional sense of the term, but software does tend to deteriorate as it evolves
32	Software Engineering	Is a set of rules about developing software products	Has been around as a discipline since the early 50's	Started as a response to the so-called 'Software Crisis' of the late 90's	Is an engineering discipline concerned with all the aspects of software production		Is now a mature discipline on par with other established engineering fields.
33	Read the following paragraph and identify the correct statement."Imagine that you were recently hired as a software engineer to a company that specializes in aircraft navigation control software. While orientating yourselves to the company's work practices, you observe that they in fact do not conduct a few tests that they should in order to comply with the relevant safety standard. When you inquire about this from the project manager, he dismisses it saying that those tests are really unnecessary (and takes an unreasonably long time to conduct, as well as being superfluous) and that they have managed with the other tests for so long, without any problems."(	You should immediately resign from the company and file a complaint with the relevant standard institution	You should do nothing and let the matter slide	Although you are new to the company, and you hardly know anything about the internal processes and politics, you should insist on the company changing its work practices immediately; failing which you threaten to report the matter	Since you are new to the company, and you are unfamiliar with the internal processes and politics, you should first find-out more about the issue and its background		Since you are new to the company, and you are unfamiliar with the internal processes and politics, you should first find-out more about the issue and its background
34	With regard to Evolutionary development, identify the correct statement	Evolutionary development usually comes in two flavours; exploratory development, and throw-away prototyping	Very large projects are natural candidates for an evolutionary development based approach	Exploratory development is used in situations where most of the requirements are well understood in advance	One of the strong points of evolutionary development is that it facilitates easy project management, through the high volume of documentation it generates		Evolutionary development usually comes in two flavours; exploratory development, and throw-away prototyping

35	What is the fundamental reason that software cannot be considered to be engineered?	It is designed by humans and therefore flawed	Software engineering (as opposed to other forms of engineering, such as Civil) is an art - not a science	The discipline is relatively new, say in comparison to bridge building that is an activity that has millennia of practice	The complexity of systems and their interaction continues faster than we can understand it.		The complexity of systems and their interaction continues faster than we can understand it.
36	The software life cycle can be said to consist of a series of phases. The classical model is referred to as the waterfall model. Which phase may be defined as "The concept is explored and refined, and the client's requirements are elicited?"(	Requirements	Specification	Design	Implementation		Requirements
37	The individual or organisation who wants a product to be developed is known as the	Developer	User	Contractor	Client.		Client.
38	Which of the following items should not be included in the software project management plan?(	The techniques and case tools to be used	Detailed schedules, budgets and resource allocations	The life cycle model to be used	None of the above		None of the above
39	The final form of testing COTS software is _____ testing	Unit	Integration	Alpha	Beta		Beta.
40	In the maintenance phase the product must be tested against previous test cases. This is known as _____ testing	Unit	Integration	Regression	Module		Regression
41	Which property of the rapid prototype is not important?	The speed with which it can be developed	The speed with which it can be modified	Its ability to determine the client's real needs	Its internal structure		Its ability to determine the client's real needs
42	An example of the risk involved in software development is	Key personnel may resign before the product is complete	The manufacturer of critical components	Technology changes may render the product obsolete	All of these are risks involved in software development.		All of these are risks involved in software development.
43	A simple way of looking at the spiral software life-cycle model is as a waterfall model with each phase proceeded by	Build-and-fix	Freezing	Synchronization	Risk analysis		Risk analysis
44	.The degree of interaction between two modules is known as	Cohesion	Strength	Inheritance	Coupling		Coupling
45	The relationship between a derived class (or subclass) and base class is referred to as	Association	Inheritance	Polymorphism	Instantiation		Inheritance
46	Myers (1978) identifies seven levels of cohesion. Which level of cohesion maybe defined as followed; "the output from one element in the component serves as input for some other element"?(	Communicational cohesion	Functional cohesion	Communicational cohesion	Temporal cohesion		Communicational cohesion
47	A design is said to be a good design if the components are	Strongly coupled	Weakly cohesive	Strongly coupled and Weakly cohesive	Strongly cohesive and weakly coupled		Strongly cohesive and weakly coupled
48	If a control switch is passed as an argument this is an example of _____ coupling	Content	Common	Control	Data		Control
49	Which of the following is a type of abstraction?	Data	Procedural	Iteration	All of the above		All of the above
50	.In the classical chief programmer team approach, the team member responsible for maintaining the detailed design and coding is	The chief programmer	The programming secretary	A specialized function that exists outside 'the team'	The individual coder		The individual coder

51	Internal costs include	Developers salaries	Managers and support personnel salaries	The cost of overheads such as utilities, rent and senior managers	Materials (such as manuals) and services such as travel		Developers salaries
52	Problems with using Lines of Code to measure the size of a product include	The creation of source code is only part of the development effort	The Lines of Code (LOC) will differ between languages and cannot be measured for some languages	Should comments, data definitions etc (i.e. non-executable LOC) be included as well	All of the above		All of the above
53	Software Science bases its estimation of the size of a product on	and Processes	Lines of Code (kLOC)	Function Points (FP	operands and operators		operands and operators
54	In Intermediate COCOMO the mode that represents complex products is referred to as	Embedded	Semidetached	Organic	Multiplicative		Embedded
55	Work that continues throughout the project and does not relate to any specific phase of software development is termed a(n)	Milestone	Project function	Activity	Task		Project function
56	The advantage of following the IEEE Standard for drawing up a Software Project Management Plan (SPMP) - seeIEEE Standard 1059.1 1987 - is	It is drawn up by representatives from major software development organisations	It is designed for all types of software products	It is a framework that can be used irrespective of process model or specific techniques	All of the above		All of the above
57	The best way to test the Software Project Management Plan (SPMP) is by	Prototyping	Inspection	Simulation	Compilation		Inspection
58	Algorithmic cost estimation in different organisations may be different for the same application development, because	Different organisations consider complexity factors differently	Different organisations may use different programming languages	Developers' skills may vary	All of the above may be true		All of the above may be true
59	The aim of software engineering is to produce software that is	Fault-free	Delivered on time	Delivered within budget	All of these are the aims of software engineering		All of these are the aims of software engineering
60	Object-oriented concepts are not new. The first OO language was considered to be	ALGOL-68(b)FORTRAN 77	MODULA	FORTRAN	SIMULA		SIMULA

;

Unit – V

Risk Management – Nature and types of risk – managing risk – evaluating risk – A software management process frame works- Life cycle phases – software maintenance and configuration management – Maintenance characteristics – Maintenance task – Maintenance side effects.

What Is Risk In Software?

Very simply, a risk is a *potential* problem. It's an activity or event that may compromise the success of a software development project. Risk is the possibility of suffering loss, and total risk exposure to a specific project will account for both the *probability* and the *size* of the potential loss.

Guesswork and crisis-management are never effective. Identifying and aggregating risks is the only predictive method for capturing the probability that a software development project will experience unplanned or inadmissible events. These include terminations, discontinuities, schedule delays, cost underestimation, and overrun of project resources

Risk management includes the following tasks:

- ***Identify*** risks and their triggers
- ***Classify*** and prioritize all risks
- Craft a ***plan*** that links each risk to a mitigation
- ***Monitor*** for risk triggers during the project
- Implement the ***mitigating action*** if any risk materializes
- ***Communicate*** risk status throughout project

Identify and Classify Risks

Most software engineering projects are inherently risky because of the variety potential problems that might arise. Experience from other software engineering projects can help managers classify risk. The importance here is not the elegance or range of classification, but rather to

precisely *identify and describe all of the real threats to project success*. A simple but effective classification scheme is to arrange risks according to the areas of impact.

Five Types of Risk In Software Project Management

For most software development projects, we can define five main risk impact areas:

- New, unproven technologies
- User and functional requirements
- Application and system architecture
- Performance
- Organizational

New, unproven technologies. The majority of software projects entail the use of new technologies. Ever-changing tools, techniques, protocols, standards, and development systems increase the probability that technology risks will arise in virtually any substantial software engineering effort. Training and knowledge are of critical importance, and the improper use of new technology most often leads directly to project failure.

User and functional requirements. Software requirements capture all user needs with respect to the software system features, functions, and quality of service. Too often, the process of requirements definition is lengthy, tedious, and complex. Moreover, requirements usually change with discovery, prototyping, and integration activities. Change in elemental requirements will likely propagate throughout the entire project, and modifications to user requirements might not translate to functional requirements. These disruptions often lead to one or more critical failures of a poorly-planned software development project.

Application and system architecture. Taking the wrong direction with a platform, component, or architecture can have disastrous consequences. As with the technological risks, it is vital that the team includes experts who understand the architecture and have the capability to make sound design choices.

Performance. It's important to ensure that any risk management plan encompasses user and partner expectations on performance. Consideration must be given to benchmarks and threshold testing throughout the project to ensure that the work products are moving in the right direction.

Organizational. Organizational problems may have adverse effects on project outcomes. Project management must plan for efficient execution of the project, and find a balance between the needs of the development team and the expectations of the customers. Of course, adequate staffing includes choosing team members with skill sets that are a good match with the project.

Risk Management Plan

After cataloging all of the risks according to type, the software development project manager should craft a risk management plan. As part of a larger, comprehensive project plan, the risk management plan outlines the response that will be taken for each risk—if it materializes.

Monitor and Mitigate

To be effective, software risk monitoring has to be integral with most project activities. Essentially, this means frequent checking during project meetings and critical events.

Monitoring includes:

- Publish project status reports and include risk management issues
- Revise risk plans according to any major changes in project schedule
- Review and reprioritize risks, eliminating those with lowest probability
- Brainstorm on potentially new risks after changes to project schedule or scope

When a risk occurs, the corresponding mitigation response should be taken from the risk management plan.

Mitigating options include:

- **Accept:** Acknowledge that a risk is impacting the project. Make an explicit decision to accept the risk without any changes to the project. Project management approval is mandatory here.
- **Avoid:** Adjust project scope, schedule, or constraints to minimize the effects of the risk.
- **Control:** Take action to minimize the impact or reduce the intensification of the risk.
- **Transfer:** Implement an organizational shift in accountability, responsibility, or authority to other stakeholders that will accept the risk.
- **Continue Monitoring:** Often suitable for low-impact risks, monitor the project environment for potentially increasing impact of the risk.

Communicate

Throughout the project, it's vital to ensure effective communication among all stakeholders, managers, developers, QA—especially marketing and customer representatives. Sharing information and getting feedback about risks will greatly increase the probability of project success.

Software risk management is all about risk quantification of risk. This includes:

1. Giving a precise description of risk event that can occur in the project
2. Defining risk probability that would explain what are the chances for that risk to occur
3. Defining How much loss a particular risk can cause
4. Defining the liability potential of risk

Risk Management comprises of following processes:

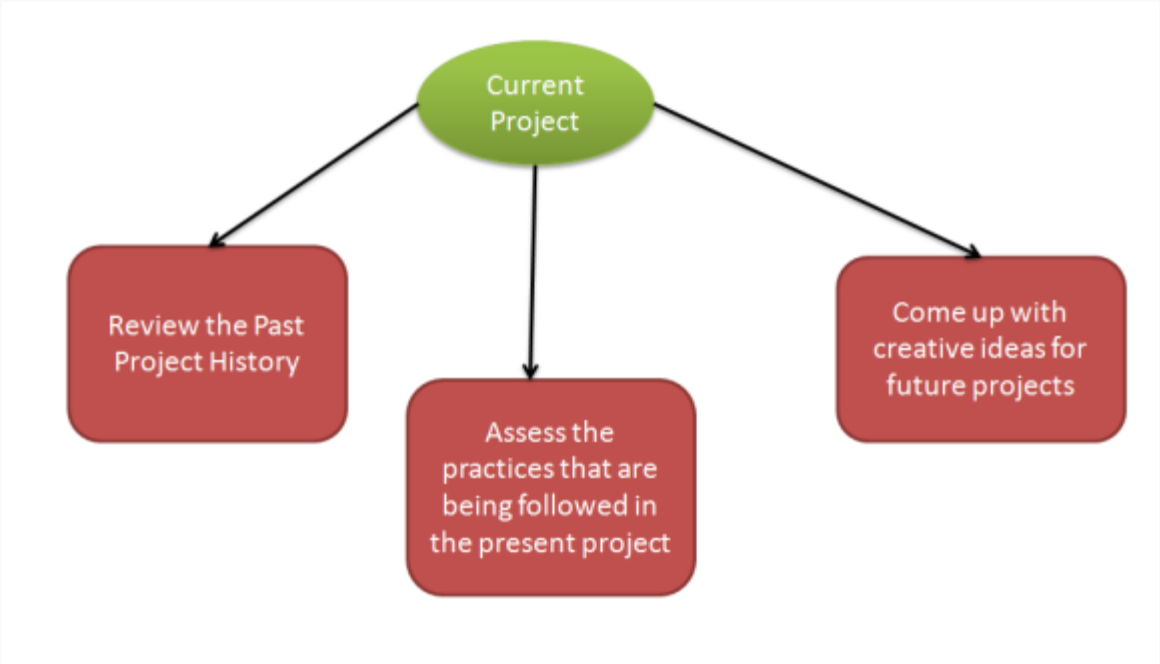
1. Software Risk Identification
2. Software Risk Analysis
3. Software Risk Planning
4. Software Risk Monitoring

These Processes are defined below.

Software Risk Identification

In order to identify the risks that your project may be subjected to, it is important to first study the problems faced by previous projects. Study the project plan properly and check for all the possible areas that are vulnerable to some or the other type of risks. The best ways of analyzing a project plan is by converting it to a flowchart and examine all essentialareas. It is important to

conduct few brainstorming sessions to identify the known unknowns that can affect the project. Any decision taken related to technical, operational, political, legal, social, internal or external factors should be evaluated properly.



Software Risk Identification

In this phase of Risk management you have to define processes that are important for risk identification. All the details of the risk such as unique Id, date on which it was identified, description and so on should be clearly mentioned.

Software Risk Analysis

Software Risk analysisis a very important aspect of risk management. In this phase the risk is identified and then categorized. After the categorization of risk, the level, likelihood (percentage) and impact of the risk is analyzed. Likelihood is defined in percentage after examining what are the chances of risk to occur due to various technical conditions. These technical conditions can be:

1. Complexity of the technology
2. Technical knowledge possessed by the testing team
3. Conflicts within the team
4. Teams being distributed over a large geographical area
5. Usage of poor quality testing tools

With impact we mean the consequence of a risk in case it happens. It is important to know about the impact because it is necessary to know how a business can get affected:

1. What will be the loss to the customer
2. How would the business suffer
3. Loss of reputation or harm to society
4. Monetary losses
5. Legal actions against the company
6. Cancellation of business license

Level of risk is identified with the help of:

Qualitative Risk Analysis: Here you define risk as:

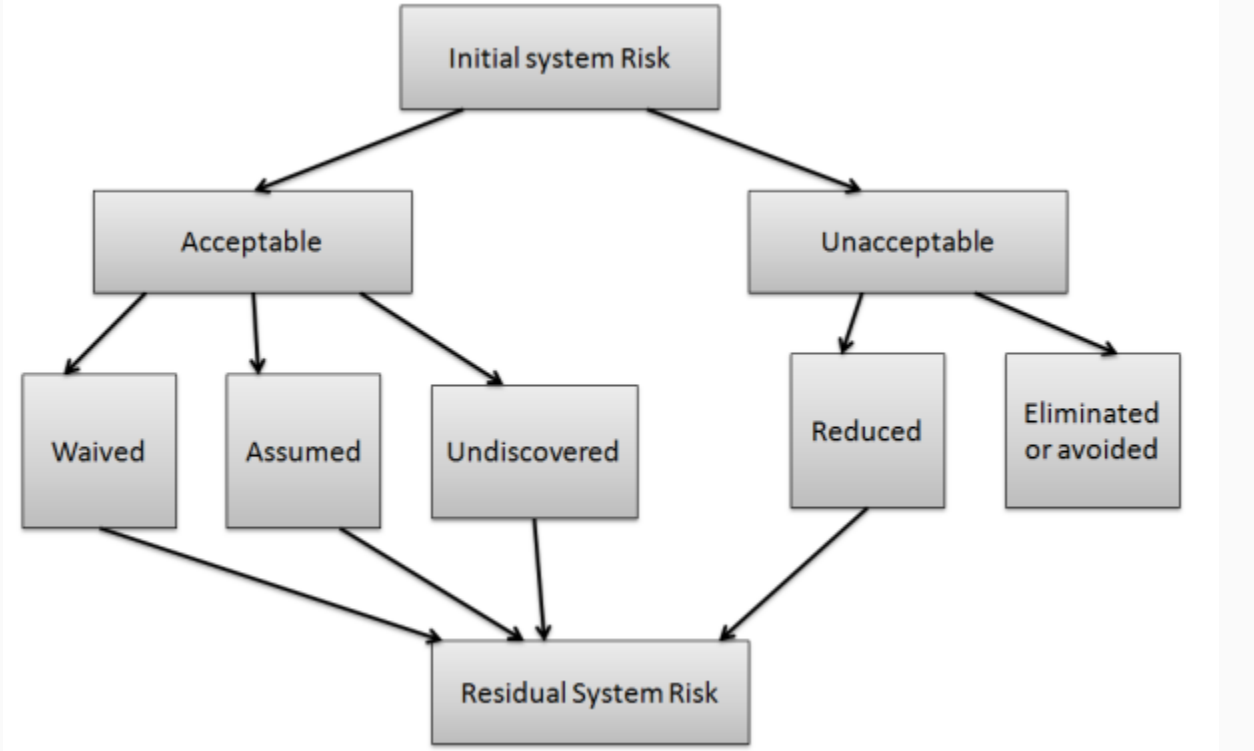
- High
- Low
- Medium

Quantitative Risk Analysis: can be used for software risk analysis but is considered inappropriate because risk level is defined in % which does not give a very clear picture.

Software Risk Planning

Software risk planning is all about:

1. Defining preventive measure that would lower down the likelihood or probability of various risks.
2. Define measures that would reduce the impact in case a risk happens.
3. Constant monitoring of processes to identify risks as early as possible.



Software Risk Planning

Software Risk Monitoring

Software risk monitoring is integrated into project activities and regular checks are conducted on top risks. Software risk monitoring comprises of:

- Tracking of risk plans for any major changes in actual plan, attribute, etc.
- Preparation of status reports for project management.
- Review risks and risks whose impact or likelihood has reached the lowest possible level should be closed.
- Regularly search for new risks

Software Process Framework

- The process of framework defines a small set of activities that are applicable to all types of projects.
- The software process framework is a collection of task sets.
- Task sets consist of a collection of small work tasks, project milestones, work productivity and software quality assurance points.

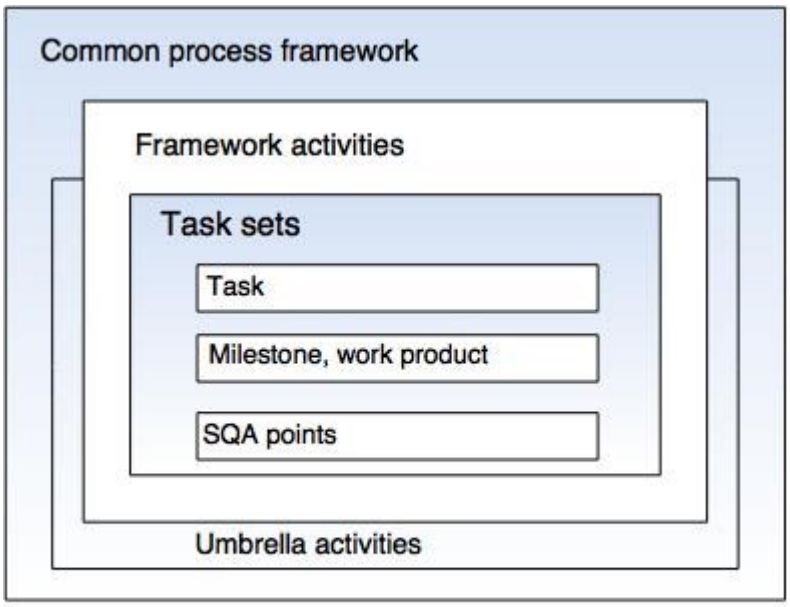


Fig.- A software process framework

Umbrella activities

Typical umbrella activities are:

1. Software project tracking and control

- In this activity, the developing team accesses project plan and compares it with the predefined schedule.
- If these project plans do not match with the predefined schedule, then the required actions are taken to maintain the schedule.

2. Risk management

- Risk is an event that may or may not occur.
- If the event occurs, then it causes some unwanted outcome. Hence, proper risk management is required.

3. Software Quality Assurance (SQA)

- SQA is the planned and systematic pattern of activities which are required to give a guarantee of software quality.

For example, during the software development meetings are conducted at every stage of development to find out the defects and suggest improvements to produce good quality software.

4. Formal Technical Reviews (FTR)

- FTR is a meeting conducted by the technical staff.
- The motive of the meeting is to detect quality problems and suggest improvements.
- The technical person focuses on the quality of the software from the customer point of view.

5. Measurement

- Measurement consists of the effort required to measure the software.
- The software cannot be measured directly. It is measured by direct and indirect measures.
- Direct measures like cost, lines of code, size of software etc.
- Indirect measures such as quality of software which is measured by some other factor. Hence, it is an indirect measure of software.

6. Software Configuration Management (SCM)

- It manages the effect of change throughout the software process.

7. Reusability management

- It defines the criteria for reuse the product.
- The quality of software is good when the components of the software are developed for certain application and are useful for developing other applications.

8. Work product preparation and production

- It consists of the activities that are needed to create the documents, forms, lists, logs and user manuals for developing a software.

software configuration management (**SCM** or **S/W CM**) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of configuration management SCM practices include revision control and the establishment of baselines. If something goes wrong, SCM can determine what was changed and who changed it. If a configuration is working well, SCM can determine how to replicate it across many hosts.

The goals of SCM are generally

- Configuration identification - Identifying configurations, configuration items and baselines.
- Configuration control - Implementing a controlled change process. This is usually achieved by setting up a change control board whose primary function is to approve or reject all change requests that are sent against any baseline.
- Configuration status accounting - Recording and reporting all the necessary information on the status of the development process.
- Configuration auditing - Ensuring that configurations contain all their intended parts and are sound with respect to their specifying documents, including requirements, architectural specifications and user manuals.
- Build management - Managing the process and tools used for builds.
- Process management - Ensuring adherence to the organization's development process.
- Environment management - Managing the software and hardware that host the system.
- Teamwork - Facilitate team interactions related to the process.
- Defect tracking - Making sure every defect has traceability back to the source.

Software Maintenance

Software Maintenance is the process of modifying a software product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software application after delivery to correct faults and to improve performance.

Need for Maintenance

Software Maintenance must be performed in order to:

- Correct faults.
- Improve the design.
- Implement enhancements.
- Interface with other systems.
- Accommodate programs so that different hardware, software, system features, and telecommunications facilities can be used.
- Migrate legacy software.

- Retire software.

Categories of Software Maintenance

Maintenance can be divided into the following:

1. **Corrective maintenance:**
   Corrective maintenance of a software product may be essential either to rectify some bugs observed while the system is in use, or to enhance the performance of the system.
2. **Adaptive maintenance:**
   This includes modifications and updations when the customers need the product to run on new platforms, on new operating systems, or when they need the product to interface with new hardware and software.
3. **Perfective maintenance:**
   A software product needs maintenance to support the new features that the users want or to change different types of functionalities of the system according to the customer demands.
4. **Preventive maintenance:**
   This type of maintenance includes modifications and updations to prevent future problems of the software. It goals to attend problems, which are not significant at this moment but may cause serious issues in future.

Reverse Engineering

Reverse Engineering is processes of extracting knowledge or design information from anything man-made and reproducing it based on extracted information. It is also called back Engineering.

Software Reverse Engineering

Software Reverse Engineering is the process of recovering the design and the requirements specification of a product from an analysis of it's code. Reverse Engineering is becoming important, since several existing software products, lack proper documentation, are highly unstructured, or their structure has degraded through a series of maintenance efforts.

Why Reverse Engineering?

- Providing proper system documentatiuon.
- Recovery of lost information.
- Assisting with maintenance.
- Facility of software reuse.
- Discovering unexpected flaws or faults.

Used of Software Reverse Engineering –

- Software Reverse Engineering is used in software design, reverse engineering enables the developer or programmer to add new features to the existing software with or without knowing the source code.
- Reverse engineering is also useful in software testing, it helps the testers to study the virus and other malware code .

KARPAGAM ACADEMY OF HIGHER EDUCATION - COIMBATORE 21

SUBJECT CODE:17MBAPS401B SUBJECT: SOFTWARE PROJECT MANAGEMENT

	Unit 5						
S.NO	**Question**	**option1**	**option2**	**option3**	**option4**		**Answer**
1	Software engineering is the systematic approach to the	Development of software	Operation of software	Maintenance of software	All of the above		All of the above
2	Brooks' view of the essence of software included	People, Quality, Process and Productivity	Performance, Robustness, Maintainability and Reusability	Complexity, Conformity, Changeability and Invisibility	Efficiency, Reliability, Usability and Robustness		Complexity, Conformity, Changeability and Invisibility
3	What is the essence of software engineering?	Requirements Definition, Design Representation, Knowledge Capture and Quality Factors	Maintaining Configurations, Organizing Teams, Channeling Creativity and Planning ResourceUse	Time/Space Tradeoffs, Optimizing Process, Minimizing Communication and Problem Decomposition	Managing Complexity, ManagingPersonnel Resources, Managing Time and Money and Producing Useful Products		Managing Complexity, ManagingPersonnel Resources, Managing Time and Money and Producing Useful Products
4	Which of the following is a life-cycle concern?	Testing	Portability	Programming	Planning		Planning
5	Which best captures the nature of the quality paradigm?	The Nature of Quality, A Process Perspective, Defect Elimination	Process, Product, People, Problem	Measurement, Quality Control, Validation	Feasibility, Requirements, Economics, Customer's Needs(e)Analysis, Testing, Design.		The Nature of Quality, A Process Perspective, Defect Elimination
6	Prototyping is appropriate for	Data-oriented applications	Applications with emphasis on the user interface	Applications which are highly interactive	All of the above		All of the above
7	What are the major activities of the spiral model of software engineering?	Planning, Risk Analysis, Engineering, Customer Evaluation	Defining, Prototyping, Testing, Delivery	Requirements	Quick Design, Build Prototype, Evaluate Prototype, Refine Prototype		Planning, Risk Analysis, Engineering, Customer Evaluation
8	In choosing a development life-cycle model, one would consider the	Development Group Expertise, Problem Characteristics, User Expectations	Languages, Development Schedule, Competition	System Context, User Population, Platforms	Organizational Structure, User Tasks, Performance Criteria		Development Group Expertise, Problem Characteristics, User Expectations
9	What are the factors to be considered when planning a software development effort?	Performance, Problem, Product, Planning	People, Problem, Product, Process	People, Problem, Productivity, Performance	People, Problem, Product, Portability		People, Problem, Product, Process
10	Which of the following could be a deliverable for a software system?	Source Code	Reference Manual	Requirements Document	All of the above		All of the above
11	.Which of the following is not viewed as a primary mover in improving the software process?	Increased Effectiveness	Improved Staff Satisfaction	Better Product Quality	Tighter managerial control		Tighter managerial control
12	Symptoms of the software crisis would include	Software delivered behind schedule	Software exceeding cost estimate	Unreliable	Difficult to maintain		Difficult to maintain
13	Which of the following projects would be a good one for adopting the prototyping paradigm for software development?	Accounting System	Spread sheet	Automobile Cruise Control	Algebra Tutor		Algebra Tutor
14	Views of quality software would not include	Optimizing price and performance	Minimizing the execution errors	Conformance to specification	Establishing valid requirements		Minimizing the execution errors(c)Conformance to specification
15	Software configuration activities would not include	Identify change	Control change	Ensure improper implementation of change	Report change to interested parties		Ensure improper implementation of change
16	In planning a software project one would	Find ways to produce results using limited resources	Pad the schedule to accommodate errors	Overestimate the budget	Structure the team to prevent administrative interference		Find ways to produce results using limited resources
17	A systematic approach to software development, as epitomized by the various life-cycle models, is useful in	Helping us understand the nature of the software product	)Convincing the customer that we know what we are doing	Filling texts on software engineering	Managing the various activities necessary to get the job done		Managing the various activities necessary to get the job done
18	A process view in software engineering would consider which of the following	Product performance	Staffing	Functionality	Reliability		Staffing
19	Software measurement is useful to	Indicate quality of the product	Track progress	Assess productivity	All of the above		All of the above

20	Which of the following is not a 'concern' during the management of a software	Money	Time	Product quality	Product quantity		Product quantity
21	What would be investigated during Requirements analysis?	System performance , Test Scheduling, Organizational Structure	Languages , Platforms, Competition	System Context , User Populations, User Tasks	Verification, Formal Methods, Accuracy		System Context , User Populations, User Tasks
22	Which of the following is not a description of planning?	Planning is used to find credible ways to produce results with limited resources and limited schedule flexibility	Planning is finding new personnel resources to support labour intensive development	Planning is identifying and accommodating the unforeseen	Planning is blending the efforts of many people to produce a product that satisfies the customer's need		Planning is finding new personnel resources to support labour intensive development
23	The information we need to capture during requirements analysis not include	Hiring Authority	Communication Paths	Synchronization	Temporal Dependencies		Hiring Authority
24	What do you call when two modules are coupled, when they communicate via a composite data item?	Content coupling	Common coupling	Control coupling	Stamp coupling		Stamp coupling
25	Which among the following measures how strongly the elements within a module are related?	Coupling	Cohesion	Aggregation	inheritance		Cohesion
26	What do you call, when the elements of a module, all operation the same data?	Functional cohesion	Temporal cohesion	Procedural cohesion	Communicational cohesion		Communicational cohesion
27	Which tests are designed to confront the program with abnormal situations?	Recovery testing	Security testing	Stress testing	Performance testing		Stress testing
28	To which software category does Knowledge based system belongs?	System software	Real time software	Embedded software	Artificial Intelligent software		Artificial Intelligent software
29	Which is not involved in software development process?	People	Process	Problem	Practice		Practice
30	Which of the following are direct measures? I. Size. I. Effort.III. Schedule.IV. Quality	Both (I) and (II) above	Both (I) and (III) above	Both (I) and (IV) above(d)Both (II) and (III) above	All (I), (II), (III) and (IV) above		All (I), (II), (III) and (IV) above
31	How does a software project manager need to act to minimize the risk of software failure?	Double the project team size	Request a large budget	Form a small software team	Track progress		Track progress
32	To be an effective aid in process improvement the baseline metrics used must be	Based on reasonable estimates from failed projects	Measured consistently across projects	Drawn from large projects only	Drawn from failed projects		Measured consistently across projects
33	Empirical estimation models are typically based on	Expert judgment based on past project experiences	)Refinement of current project estimation	Regression models derived from historical project data	Trial and error determination of the parameters and coefficients		Regression models derived from historical project data
34	Which of the following is not the guiding principle of software project scheduling?	Compartmentalization	Market assessment	Time allocation	Effort validation		Market assessment
35	The tools for computing critical path and project completion times from activity networks is/are I. CPM.II. DRE.III. FP.IV. PERT	Both (I) and (III) above	Both (I) and (IV) above	Both (II) and (IV) above	Both (II) and (III) above		Both (I) and (IV) above
36	The purpose of earned value analysis is to	Determine how to compensate developers based on their productivity	Provide a quantitative means of assessing software project progress	Provide a qualitative means of assessing software project progress	Set the price point for a software product based on development effort		Provide a quantitative means of assessing software project progress
37	The rapid application development model is	Same as component-based development	A useful approach when a customer cannot define requirements clearly	A high-speed adaptation of the linear sequential model	Same as incremental model		A high-speed adaptation of the linear sequential model
38	Which of the following is not an objective for building analysis models?	Define set of software requirements that can be validated	Describe customer requirements	Develop a solution for the problem	Establish basis for software design		Develop a solution for the problem
39	The entity relationship diagram	Depicts relationships between data objects	Depicts functions that transform the data flow	Indicates how data are transformed by the system	Indicates system reactions to external events		Depicts relationships between data objects
40	Which of the following is not an area of concern in the design model?	Architecture	Data design	Interfaces design	Project scope		Project scope
41	Coupling is a qualitative indication of the degree to which a module	Can be written more compactly	Focuses on just one thing	Is able to complete its function in a timely manner	Is connected to other modules		Is connected to other modules
42	Which of the following interface design principles reduces the user's memory load? Define intuitive shortcuts II. Disclose information in a progressive fashion III. Establish meaningful defaults IV. Provide an on-line tutorial	Only (I) above	Only (II) above	Only (III) above	)(I), (II) and (III) above		)(I), (II) and (III) above
43	Black box testing is also called	Specification-based testing	Structural testing	Verification	Unit testing		Specification-based testing
44	Which configuration objects would not typically be found in the project database?	Design specification	Marketing data	Executable code	Test plans		Marketing data

45	Which of the following task(s) is/are not part of software configuration management? Change control.II. Reporting.III. Statistical quality control	Only (I) above	Only (II) above	Only (III) above	Both (I) and (II) above		Only (III) above
46	Which box specification isnotassociated with cleanroom approach?	Black box	Clear box	State box	White box		White box
47	Which of the following is not a logical layer of the application in client server system?	Presentation layer	Application layer	Data Management layer	Programming layer		Programming layer
48	Traditionally, the phase of software development where a formal approach used is	Processes application logic	Performs data management task	Places heavy processing load on the server	Makes use of processing power of the client		Places heavy processing load on the server
49	Domain Engineering in CBSE is to	Identification of components	Catalogue reusable components	Domain modelling	All the above		All the above
50	which of the following is not a technology driver for an system	enterprise applications	object technologies	knowledge asset management	collaborative technologies		knowledge asset management
51	Software risk always involves two characteristics	fire fighting and crisis management	known and unknown risks	uncertainty and loss	staffing and budget		uncertainty and loss
52	Three major categories of risks are	business risks, personnel risks, budget risks	project risks, technical risks, business risks	planning risks, technical risks, personnel risks	management risks, technical risks, design risks		project risks, technical risks, business risks
53	A risk item checklist would contain known and predictable risks from which of these categories?	product size	development environment	staff size	process definition		product size
54	Questions that should be asked to assess the overall project risk include:	Have top managers formally committed to support the project?	Are end-users committed to the project and proposed system being built?	Are requirements fully understood by development team and customers?	Does the proposed budget have time allocated for marketing?		Does the proposed budget have time allocated for marketing?
55	Software risk impact assessment should focus on consequences affecting	planning, resources, cost, schedule	marketability, cost, personnel	business, technology, process	performance, support, cost, schedule		marketability, cost, personnel
56	Risk projection attempts to rate each risk in two ways	likelihood and size	likelihood and probability	likelihood and impact	likelihood and mitigation		likelihood and probability
57	Risk tables are sorted by	probability and cost	probability and impact	probability and size	probability and exposure		probability and size
58	Which factors affect the probable consequences likely if a risk does occur?	risk cost	Risk timing	Risk scope	Risk resources		Risk cost
59	An effective risk management plan will need to address which of the following issues?	risk avoidance	risk monitoring	contingency planning	all of the above		risk monitoring
60	Hazard analysis focuses on the identification and assessment of potential hazards that can cause	project termination	schedule slippage	external problems	entire system to fail		schedule slippage

Reg…….
[17MBAPS401B]

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act, 1956)
Coimbatore-641021
(For candidates admitted from 2017 onwards)
CONTINUOUS INTERNAL ASSESSMENT I- FEBRUARY 2019.
MASTER OF BUSINESS ADMINISTRATION
FOURTH SEMESTER
SOFTWARE PROJECT MANAGEMENT

Time: 2 Hours **Maximum: 50 Marks**
Date: 04.02.2019 **Session : FN**

PART – A (15 X 1 = 15 Marks)
ANSWER ALL THE QUESTIONS

1. Software is
   a. superset of programs b. subset of programs
   c. set of programs d. set of problems
2. Effort is measured in terms of
   a. person months b. rupees c. persons d. months
3. Milestone are used to
   a. know the cost of the project c. know the status of the project
   b. know user expectations d. know process
4. CASE tools is
   a. computer aided software engineering
   b. component aided software engineering
   c. constructive aided software engineering
   d. computer analysis software engineering
5. If requirement are easily understandable and defined, which model is best suited?
   a. waterfall model b. spiral model c. quick and fix model
   d. capability maturity model
6. The term module used during design phase refers to_____
   a. function b. system c. process d. project
7. Software mistakes during coding are known as _____
   a. Failures b. defects c. bugs d. errors
8. Regression testing is primarily related to____ testing
   a. Functional b. data flow c. development d. maintenance
9. Integration testing techniques are
   a. top down b. bottom up c. sandwich d. vertical
10. A COCOMO model is
    a. Common Cost Estimation Model
    b. Constructive cost estimation model
    c. Complete cost estimation model
    d. Comprehensivve cost estimation model

11. Total number of units in Function point analysis are
    a. 2 b. 5 c. 4 d. 1
12. CMM stands for
    a. Capacity maturity model c. capability maturity model
    b. cost management model d. comprehensive maintenance model
13. The model to measure the software process improvement is called
    a. ISO9000 b. ISO9126 c. CMM d. Spiral model
14. which is not a product metric?
    a. size b. reliability c. productivity d. functionality
15. SRS stands for
    a. software requirement specification c. software requirement solution
    b. system requirement specification d. system random software

PART B (3 X 8 =24 Marks)
ANSWER ALL THE QUESTIONS

16. a. From your learning, give the different steps in software development planning.
    (OR)
    b. Write in brief how COCOMO is used as a cost estimating model?

17. a. "Quality is an inherent tool for a software process" - Discuss.
    (OR)
    b. What is function point analysis? Explain.
18. a. Describe in detail about CASE tools and its uses in project management.
    (OR)
    b. Is testing necessary for software? If Yes what are the ways to test a software?

PART C (1 X 11 = 11 Marks)
ANSWER ALL THE QUESTIONS

19. **Case study (Compulsory)**

Awara IT solutions is one of the leading providers of IT services and solution for efficient business in Russia. In addition, Awara provides service for customers from Scandinavia and Western Europe. They implemented a new process 'Easy project' to connect support with project management

Before they implemented the Easy Project, the client used three systems at the same time Redmine and MS project for project management and KAYAKO for automating customer support services. The advantage of this solution has been its disability to unify support and project management into one system.

That is the main requirement for Easy Project was its complexity- aggregation of various business processes such as planning, project management, automation of customer support, budget monitoring, back office processes: scheduling, attendance, payments all in one system.

Question: From your learning analyse the case and give a project plan and requirement plan for Awara IT solutions.