T.Y. B.Sc. (IT) : Sem. V

Software Project Management

Time : 2½ Hrs.] Prelim Question Paper Solution [Marks : 75

Q.1 Attempt the following (any THREE) **[15]**

Q.1(a) Explain the categories of software projects. **[5]**

Ans.: The nature of software project varies from application to application. It is very much essential to identify the characteristics of a project which could affect the way in which it should be managed.

There are 4 ways in which software projects can be categorized :

1. **Outsourced Projects :** While working on medium to large commercial projects, one can outsource some modules of the project to some third -party organisation. There are many reasons for outsourcing which include lack of in-house expertise, or cost cutting etc. When project ts or sub parts of a project are outsourced, they can be completed in the stipulated time period. However there are certain challenges while projects are outsourced.

2. **Object Driven Development :** It should be very clear in the start whether the main objective of the project is to produce a product or to meet certain objectives. A lot of software projects operates in two stages. First is an object driven project resulting in recommendations. Secondly create a software product.

3. **Compulsory Versus Voluntary Users :** In many organisations, there are systems which staff have to use if they want to do something such as recording a sales transaction. However the use of a system is becoming voluntary as in the case of computer games. In such cases it is difficult to elicit precise requirements from potential users as we could with a business system. What the game will do will thus depend much on the informed ingenuity of the developers, along with techniques such as market surveys, focus groups etc.

4. **Information Systems Versus Embedded Systems :** Information systems control processes whereas embedded systems controls machines like home automation, automated warehouse.

Q.1(b) Explain the various Cost Benefit Evaluation Techniques. **[5]**

Ans.: The various cost–benefit evaluation techniques are as follows :

1. **Net Profit :** The net profit of a project is the difference between the total costs and the total income over the life of the project.

2. **Return on Investment :** The return on investment (ROI), also known as the accounting rate of return (ARR), provides a way of comparing the net profitability to the investment required. There are some variations on the formula used to calculate the return on investment but a straightforward common version is

$$\text{ROI} = \frac{\text{Average Annual Profity}}{\text{Total Investment}} \times 100$$

3. **Net Present value (NPV) :** The NPV takes into account the profitability of a project and the timing of the cash flows that are produced. P

4. **Internal rate of return (IRR) :** One disadvantage of NPV as a measure of profitability is that, although it may be used to compare projects, it might not be directly comparable with earnings from other investments or the costs of borrowing capital.

   The internal rate of return (IRR) attempts to provide a profitability measure as a percentage return that is directly comparable with interest rates. Thus, a project that showed an estimated IRR of 10% would be worthwhile if the capital could be borrowed for less than 10% or if the capital could not be invested elsewhere for a return greater than 10%.

Q.1(c) Explain stepwise project planning in detail. **[5]**

Ans.: Stepwise there are total 10 steps for planning projects which are explained as follows :

Step 1: Select Project

Step 2: Identify Project Scope and Objectives
- 2.1 Identify objectives and practical measures of the effectiveness in meeting those objectives
- 2.2 Establish Project authority
- 2.3 Identify all stakeholders
- 2.4 Modify objectives in the light of stakeholder analysis
- 2.5 Establish methods of communication with all parties

Step 3: Identify Project infrastructure
- 3.1 Establish relationship between project and strategic planning
- 3.2 Identify installation standards and procedures
- 3.3 Identify project team organization

Step 4: Analyse Project Characteristics
- 4.1 Distinguish the project as either objective or product driven
- 4.2 Analyse project characteristics
- 4.3 Identify high level project risks
- 4.4 Take into account user requirements concerning implementation
- 4.5 Select development methodology and life cycle approach
- 4.6 Review overall resource estimates

Step 5: Identify Project Products and Activities
- 5.1 Identify and describe project products
- 5.2 Document generic product flows
- 5.3 Recognize product instances
- 5.4 Produce ideal activity network
- 5.5 Modify the ideal to take into consideration need for stages and checkpoints

Step 6: Estimate Effort for each activity
- 6.1 Perform bottom–up estimates
- 6.2 Revise Plan to create controllable activities

Step 7: Identify Activity Risks
- 7.1 Identify and quantify activity - based risks
- 7.2 Plan risk reduction and contingency measures wherever applicable
- 7.3 Adjust overall plans and estimates to take account of risks

Step 8: Allocate Resources
- 8.1 Identify and allocate resources
- 8.2 Revise plans and estimates to take into account resource constraints

Step 9: Review Plan
- 9.1 Review Quality aspects of the project plan
- 9.2 Document the plans and obtain agreement

Step 10: Execute Plan

Q.1(d) Write a short note on Benefits Management. **[5]**

Ans.:
- Benefits management is a structured approach for maximising good business outcomes for an organisation as a result of change. It is fundamental to effective programme and project management and successful delivery.
- Benefits management involves identifying, planning, measuring and tracking benefits from the start of the programme or project investment until realization of the last projected benefit.

- It aims to make sure that the desired benefits are specific, measurable, agreed, realistic and time bounded. The term benefits management is often used interchangeably with the term benefits realisation.
- Benefits management is the common thread between programme and project delivery and successful change management.
- The approach to programme, project and change management needs to be benefit driven to ensure maximum value from the investment in change.
- Ultimately an organisation's approach to benefits realisation needs to be integrated within corporate planning to ensure a strong management focus beyond implementation of the programme or project.
- Benefits can be of many different types like QoS, motivated work force, reducing risks, mandatory compliance, reducing costs, increasing revenue and ROI.
- Benefits can be tangible (e.g. money saved, jobs created) or intangible (e.g. corporate reputation, capacity for change). They may, or may not, also be quantifiable in cash terms (e.g. reduced costs or greater customer satisfaction).

Q.1(e) Write a short note on Programme management. **[5]**

Ans.:
- Programme management is the process of managing several related projects, often with the intention of improving an organization's overall performance.
- In general, programme management is often closely related to systems engineering, industrial engineering, change management, and business transformation.
- The program manager has oversight of the purpose and status of the projects in a program and can use this oversight to support project-level activity to ensure the program goals are met by providing a decision-making capacity that cannot be achieved at project level or by providing the project manager with a program perspective when required, or as a sounding board for ideas and approaches to solving project issues that have program impacts.
- The program manager may be well placed to provide this insight by actively seeking out such information from the project managers although in large and/or complex projects, a specific role may be required.
- However, this insight arises, the program manager needs this in order to be comfortable that the overall program goals are achievable.
- Programmes can exist in different forms as seen below:
  1. Business Cycle Programmes : to decide which projects to be implemented within a predefined budget and schedule.
  2. Strategic Programmes : How business partnerships can strengthen organizations profits.
  3. Infrastructure Programmes : which defines how different departments in an organisation are responsible for carrying out various tasks like development, maintenance etc.
  4. Research and Development Programmes: Newer products must be developed in the market which can have high risks, high market value and lot of innovation.
  5. Innovative Partnerships: Many organisations should come together and work in collaboration on newer technologies in a pre-competitive phase.

Q.1(f) What do you understand by Business Case. Ellaborate. **[5]**

Ans.:
- A business case captures the reasoning for initiating a project or task. It is often presented in a well-structured written document but may also come in the form of a short verbal agreement or presentation.
- The business case provides justification for undertaking a project or programme.
- It evaluates the benefit, cost and risk of alternative options and provides a rationale for the preferred solution.
- All projects and programmes must have a business case that demonstrates the value of the work.

- In the concept phase of the life cycle an outline business case is prepared that is then used by senior management to assess whether to give the go-ahead for the definition phase.
- The detailed business case is prepared during the latter phase.

A business case document includes :

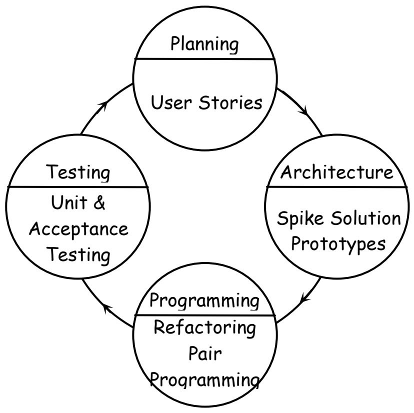
1. Introduction	2. The proposed project
3. The market	4. Organizational and operational infrastructure
5. Benefits	6. Implementation Plan
7. Costs	8. Financial Case
9. Risks	10. Management Plan

Q.2Attempt the following (any THREE) **[15]**

Q.2(a) Explain XP with the help of a neat diagram. **[5]**

Q.9 Write a short note on Extreme Programming (XP).

Ans.:



- XP is the most widely used model which follows an object-oriented approach.
- XP encompasses a set of rules and practices that occur within the context of four framework activities :
  - (i) XP Planning
  - (ii) XP Architecture
  - (iii) XP Coding
  - (iv) XP Test

(i) XP Planning :

- Begins with the creation of set of stories called as "user stories" that describe required features and functionality for s/w development.
- Each story is written by the customer and is placed on an index card. The customer assigns a value (i.e., a priority) to the story based on the overall business value of the feature or function.
- The Agile team then assesses each story and assigns it a cost.
- If the story is estimated to require more than three development weeks, the customer is asked to split the story into smaller stories and the assignment of value and cost occurs again.
- It is important to note that new stories can be written at any time.
- Stories are grouped to for a deliverable increment.
- A commitment is made on delivery date.
- After the first increment "project velocity" is used to define subsequent delivery dates for other increments.
- As development work proceeds, the customer can add stories, change the value of an existing story, split stories, or eliminate them.
- The XP team then reconsiders all remaining releases and modifies its plans accordingly.

(ii) XP Architecture :

- It follows the KIS (keep it simple) principle :
  - A simple design is always preferred over a more complex representation.
  - Encourage the use of CRC cards.
  - CRC (class-responsibility collaborator) cards identifies and organizes the object-oriented classes that are relevant to the current software increment.
- For difficult design problems, suggests the creation of "spike solutions"—a design prototype :
  - If a difficult design problem is encountered, then XP recommends the immediate creation of an operational prototype of that portion of the design called a spike solution.

(iii) XP Programming :

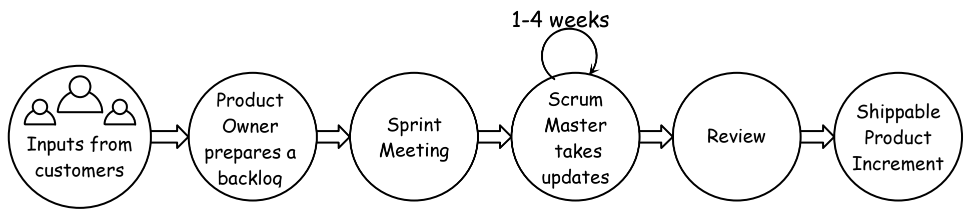
- Recommends the construction of a unit test for a story.
- Key concept during coding activity is "pair programming".
  - XP recommends that two people work together at one computer workstation to create code for a story.
  - This provides a mechanism for real-time problem solving and real-time quality assurance.
  - Encourages "refactoring" – an iterative refinement of the internal program design.
- Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves the internal structure.

(iv) XP Testing :

- All unit tests are executed daily.
- The unit tests that are created should be implemented using a framework that enables them to be automated.
- "Acceptance tests" are defined by the customer and implemented to assess visible functionality.

Q.2(b) Explain the Scrum Model in detail. **[5]**

Ans.:


1-4 weeks
Inputs from customers
Product Owner prepares a backlog
Sprint Meeting
Scrum Master takes updates
Review
Shippable Product Increment

- In SCRUM, the projects are divided into small parts of work that can be incrementally developed and delivered over time frames called sprints where each sprint takes a couple of weeks.
- Hence the product gets developed in manageable chunks.
- Scrum principles are consistent with the agile platform and are used to guide development activities within a process that incorporates the following framework activities:
  1. Requirements
  2. Analysis
  3. Design
  4. Evolution
  5. Delivery
- Within each framework activity work tasks occur within a process pattern called a sprint.
- Scrum emphasizes the use of a set of software process patterns that have proven effective for projects with tight timelines, changing requirements, and business criticality.
- Each of these process patterns defines a set of development actions :
  - **Backlog** – a prioritized list of project requirements or features that provide business value for the customer and Items can be added to the backlog at any time.
  - The product manager assesses the backlog and updates priorities as required.
  - **Sprints** – consist of work units that are required to achieve a requirement defined in the backlog that must be fit into a predefined time-box (typically 30 days).
  - Hence, the sprint allows team members to work in a short-term, but stable environment.
  - **Scrum meetings** – are typically meetings of 15 minutes held daily by the Scrum team.

- Three key questions are asked and answered by all team members:
  1. What did you do since the last team meeting?
  2. What difficulties are you facing?
  3. What is your plan to accomplish by the next team meeting?

  A team leader called a Scrum master, leads the meeting and assesses the responses from each person. The Scrum meeting helps the team to solve potential problems as early as possible.

Q.2(c) What are the problems with over and under estimates **[5]**

Ans.: A project leader like Amanda will need to be aware that an over estimate may cause the project to take longer time than usual. This can be explained by the application of two laws:

- **Parkinson's Law :** 'Work expands to fill the time available', which implies that given an easy target, staff will work less hard.
- **Brooks' Law :** The effort required to implement a project will go up disproportionately with the number of staff assigned to the project. As the project team grows in size so will the effort that has to go into management, co-ordination and communication. This has given rise, in extreme cases, to the notion of Brooks'.

Law: 'putting more personnel on a late job makes it later'. If there is an over-estimate of the effort required then this might lead to more staff being allocated than are needed and managerial overheads will be increased. This is more likely to be of significance with large projects.

Some have suggested that while the under-estimated project might not be completed in deadline or to cost, it might still be implemented in a shorter time than a project with a more generous estimate. There must, however, be limits to this phenomenon where all the slack in the project is taken up.

The danger with the under-estimate is the effect on quality. Staff, particularly those with less experience, might respond to pressing deadlines by producing work which is sub-standard.

Because of the possible effects on the behaviour of development staff caused by the size of estimates, they might be artificially reduced by their managers to increase pressure on staff. This will work only where staff are unaware that this has been done. Research has found that motivation and morale are enhanced where targets are achievable. If, over a period of time, staff become aware that the targets set are unattainable and that projects are routinely not meeting their published targets, then this will help to destroy motivation. Furthermore, people like to think of themselves as winners and there is a general tendency to put success down to our own efforts, while failure is blamed on the organization.

In the end, an estimate is not really a prediction, it is a management goal. Barry Boehm has suggested that if a software development cost is within 20% of the estimated cost estimate for the job then a good manager can turn it into a self-fulfilling prophecy. A project leader like Amanda will work hard to make the actual performance conform to the estimate.

Q.2(d) Explain Capers Jones estimating rules of thumb. **[5]**

Ans.:
- Capers Jones published a set of empirical rules based on his experience in estimating various parameters of a large number of software projects.
- Jones wanted his rules to be as simple as possible and that the project manager should make a fair decision about the various dimensions of the project.
- Most often, the following rules are used.

Rule 1: SLOC Function Point Equivalence One Function Point = 125 SLOC for C programs.

Rule 2: Project Duration Estimation Function Points raised to the power 0.4 predicts the approximate development time in calendar months.

Rule 3: Rate of requirements creep user requirements creep in at an average rate of 2% per month from the design through coding phases.

Rule 4: Defect Removal Efficiency each software review, inspection, or test step will find and remove 30% of bugs that are present.

Rule 5: Project manpower estimation. The size of the software divided by 150 predicts the approximate number of the people required for developing the application.

Rule 6: Software Development Effort Estimation. The approximate number of staff months of effort required to develop a software is given by the software development time multiplied with the number of personnel required.

Rule 7: Function Points divided by 500 predicts the approximate number of personnel required for regular maintenance activities.

Q.2(e) Explain Mark II Function points. [5]

Ans.:
- The Mark II method has been recommended by the CCTA (Central Computer and Telecommunications Agency), which lays down standards for UK government projects. At one time this Mark II approach seemed to be a good method to use with SSADM but some difficulties are now apparent. The 'Mark II' label implies an improvement and replacement of the Albrecht method. The Albrecht method, however, has had many refinements made to it and FPA Mark II remains a minority method used mainly in the UK.
- As with Albrecht the information processing size is initially measured in unadjusted function points (UFPs) to which a technical complexity adjustment can then be applied (TCA).
- It is a size estimation technique of a software product. It belongs to the class of functional point group of measurements. Traditionally, software size was measured in terms of the number of source code lines (SLOC or KLOC). The amount of SLOC had a direct association with the relative size of software.
- MK II is also known as MK II Function Point Analysis. Developers are responsible to generate an accurate estimate of the effort and cost required to deliver a software product, and also to compare different solutions, technology, and tools to assess effort and cost estimate.
- In order to carry out such activities, they must be able to determine the 'what', 'how much' aspects of the software product.

Q.2(f) Explain Cosmic Full Function Points. [5]

Ans.: The COSMIC method defines the principles, rules and a process for measuring a standard functional size of a piece of software. 'Functional size' is a measure of the amount of functionality provided by the software, completely independent of any technical or quality considerations.

Applicability of the method :

The COSMIC method may be used to size software such as business applications; real-time software; infrastructure software such as in operating systems; and hybrids of these. The common characteristic of all these types of software is that they are dominated by functions that input data, store and retrieve data, and output data. Experience has shown that the method can also often be successfully applied to size 'data manipulation-rich' software, e.g. some scientific/engineering software.

Subject to the above, the method may be applied to measure the Functional User Requirements ('FUR') of software:

- At any level of decomposition, e.g. a 'whole' piece of software or any of its components, sub-components, etc;
- In any layer of a multi-layer architecture;
- At any point in the life-cycle of the piece of software;

The principles for measuring the COSMIC functional size of a piece of software
The method uses a model of software, known as the 'COSMIC Generic Software Model', which is based on fundamental software engineering principles, namely:

- Functional user requirements of a piece of software can be analyzed into unique functional processes, which consist of sub-processes. A sub-process may be either a data movement or a data manipulation;
- Each functional process is triggered by an 'Entry' data movement from a functional user which informs the functional process that the functional user has identified an event that the software must respond to by sending data related to the event;
- A data movement moves a single data group of attributes describing a single 'object of interest', where the latter is a 'thing' of interest to a functional user;

There are four types of data movement sub-processes. An **'Entry'** moves a data group into the software from a functional user and an **'Exit'** moves a data group out. **'Writes'** and **'Reads'** move a data group to and from persistent storage, respectively.

As an approximation for measurement purposes (and in light of the applicability of the method, described above), data manipulation sub-processes are not separately measured.

The size of a piece of software is then defined as the total number of data movements (Entries, Exits, Reads and Writes) summed over all functional processes of the piece of software. Each data movement is counted as one 'COSMIC Function Point' ('CFP'). The smallest size of a functional process, and hence the size of a piece of software, is 2 CFP (COSMIC Function Points). Single functional processes have been measured in banking of over 60 CFP and in avionics of over 100 CFP.

Q.3Attempt the following (any THREE) [15]
Q.3(a) Explain Boehm's Top 10 risks and its counter measures. [5]
Ans.: Top-Ten List of Software Risk Items :

1. Personnel shortfalls
2. Unrealistic schedules and budgets
3. Developing the wrong software functions
4. Developing the wrong user interface
5. Gold plating
6. Continuing stream of requirement changes
7. Shortfalls in externally furnished components
8. Shortfalls in externally performed tasks
9. Real-time performance shortfalls
10. Straining computer-science capabilities

Counter Measures :

1. Personnel shortfalls :
- Staffing with top talent
- Job matching

- Teambuilding
- Morale building
- Cross-training
- Pre-scheduling key people

2. **Unrealistic schedules and budgets :**
   - Detailed, multisource cost and schedule estimation
   - Design to cost
   - Incremental development
   - Software reuse
   - Requirements scrubbing

3. **Developing the wrong software functions :**
   - Organizational analysis
   - Mission analysis
   - User surveys
   - Prototyping
   - Early user manuals

4. **Developing the wrong user interface :**
   - Task analysis
   - Prototyping
   - Scenarios
   - User characterization (functionality, style, workload)

5. **Gold plating :** It means adding features to a software which are marginally useful.
   - Requirements scrubbing
   - Prototyping
   - Cost-benefit analysis
   - Design to cost

6. **Continuing stream of requirement changes :**
   - High change threshold
   - Information hiding
   - Incremental development

7. **Shortfalls in externally furnished components :**
   - Benchmarking
   - Inspections
   - Reference checking
   - Compatibility analysis

8. **Shortfalls in externally performed tasks :**
   - Reference checking
   - Pre-award audits
   - Award-fee contracts
   - Competitive design or prototyping
   - Teambuilding

9. **Real-time performance shortfalls :**
   - Simulation
   - Benchmarking
   - Modeling
   - Prototyping

- Instrumentation
- Tuning

10. **Straining computer-science capabilities :**
    - Technical analysis
    - Cost-benefit analysis
    - Prototyping
    - Reference checking

Q.3(b) Explain PERT with an example. **[5]**

Ans.:
- PERT (Program Evaluation and Review Technique) was developed to take account of the uncertainty surrounding estimates of task durations. It was developed in an environment of expensive, high-risk and
- PERT was published in the same year as CPM. Developed for the Fleet Ballistic Missiles Program it is said to have saved considerable time in development of the Polaris missile.
- State-of-the-art projects - not that dissimilar to many of today's large software projects.
- The method is very similar to the CPM technique (indeed many practitioners use the terms PERT and CPM interchangeably) but. instead of using a single estimate for the duration of each task.
- PERT requires three estimates :

1. **Most likely time** - the time we would expect the task to take under normal circumstances. We shall denote this by the letter m.
2. **Optimistic time** - the shortest time in which we could expect to complete the activity, barring outright miracles. We shall use the letter a to denote this.
3. **Pessimistic time** - the worst possible time allowing for all reasonable eventualities but excluding 'acts of (iod and warfare' (as they say in most insurance exclusion clauses). We shall denote this by b.

PERT then combines these three estimates to form a single expected duration. t_e, using the formula $t_e = \frac{a + 4m + b}{6}$.

Example :

Activity	Activity durations (weeks)				
	Optimistic (a)	Most likely (m)	Pessimistic (b)	Expected (t_e)	Standard deviation (s)
A	5	6	3	6.17	0.50
B	3	4	5	4.00	0.33
C	2	3	3	2.83	0.17
D	3.5	4	5	4.08	0.25
E	1	3	4	2.83	0.50
F	8	10	15	10.50	1.17
G	2	3	4	3.00	0.33
H	2	2	2.5	2.08	0.08

Q.3(c) Explain Monte Carlo simulation. **[5]**

Ans.:
- Monte Carlo simulation, or probability simulation, is a technique used to understand the impact of risk and uncertainty in financial, project management, cost, and other forecasting models.
- The Monte Carlo Analysis is important in project management as it allows a project manager to calculate a probable total cost of a project as well as to find a range or a potential date of completion for the project.

- Since a Monte Carlo Analysis uses quantified data, this allows project managers to better communicate with senior management, especially when the latter is pushing for impractical project completion dates or unrealistic project costs.
- Also, this type of an analysis allows the project managers to quantify perils and ambiguities in project schedules.

Example of the Monte Carlo Analysis :
A project manager creates three estimates for the duration of the project: one being the most likely duration, one the worst case scenario and the other being the best case scenario. For each estimate, the project manager consigns the probability of occurrence.

The project is one that involves three tasks :
- The first task is likely to take three days (70% probability), but it can also be completed in two days or even four days. The probability of it taking two days to complete is 10% and the probability of it taking four days to finish is 20%.
- The second task has a 60% probability of taking six days to finish, a 20% probability each of being completed in five days or eight days.
- The final task has an 80% probability of being completed in four days, 5% probability of being completed in three days and a 15% probability of being completed in five days.
- Using the Monte Carlo Analysis, a series of simulations are done on the project probabilities. The simulation is to run for a thousand odd times, and for each simulation, an end date is noted.
- Once the Monte Carlo Analysis is completed, there would be no single project completion date. Instead the project manager has a probability curve depicting the likely dates of completion and the probability of attaining each.
- Using this probability curve, the project manager informs the senior management of the expected date of completion. The project manager would choose the date with a 90% chance of attaining it.
- Therefore, it could be said that using the Monte Carlo Analysis, the project has a 90% chance of being completed in X number of days.
- Similarly, a project manager can adjudge the estimated budget for a project using probabilities to simulate different end results and in turn use the findings in a probability curve.
- The above example was one that contained a mere three tasks. In reality, such projects contain hundreds if not thousands of tasks.
- Using the Monte Carlo Analysis, a project manager is able to derive a probability curve to show the ambiguity surrounding the duration and the costs surrounding these hundreds or thousands of tasks.
- Conducting simulations involving hundreds or thousands of tasks is a tedious job to be done manually.
- Today there is project management scheduling software that can conduct thousands of simulations and offer the project manager different end results in a probability curve.

Q.3(d) Write a short note on Risk Management. [5]

Ans.: Project risk is an uncertain event or condition that, if it occurs, has an effect on one or many project objectives. A risk is anything that could potentially impact project's timeline, performance or budget. Risks are potentialities, and in a project management context, if they become realities, they then become classified as "issues" that must be immediately addressed.

Categories of Risk : The most common project risks are :
- **Cost risk**, typically escalation of project costs due to poor cost estimating accuracy and scope creep.

- **Schedule risk**, the risk that activities will take longer than expected. Slippages in schedule typically increase costs and, also, delay the receipt of project benefits, with a possible loss of competitive advantage.
- **Performance risk**, the risk that the project will fail to produce results consistent with project specifications.

There are many other types of risks of concern to projects. These risks can result in cost, schedule, or performance problems and create other types of adverse consequences for the organization. For example:

- **Governance risk** relates to board and management performance with regard to ethics, community stewardship, and company reputation.
- **Strategic risks** result from errors in strategy, such as choosing a technology that can't be made to work.
- **Operational risk** includes risks from poor implementation and process problems such as procurement, production, and distribution.
- **Market risks** include competition, foreign exchange, commodity markets, and interest rate risk, as well as liquidity and credit risks.
- **Legal risks** arise from legal and regulatory obligations, including contract risks and litigation brought against the organization.
- **Risks associated with external hazards**, including storms, floods, and earthquakes; vandalism, sabotage, and terrorism; labor strikes; and civil unrest.

Q.3(e) Explain Forward pass with an example. [5]

Ans.: The first step in the calculation process is known as the Forward Pass. In the forward pass, the Early Start and Early Finish values for each activity, along with the overall Project Duration, are calculated. To facilitate schedule calculations, an "end of day" notation is used for both the Early Start and the Late Start values. By doing this, the start of the network diagram is the "end of Day Zero". In other words, the calculation process begins with placing a zero in the Early Start (ES) position of the first activity. The rest of the calculation continues with the use of the following formulas :

- Early Start = Maximum (or Highest) EF value from immediate Predecessor(s)
- Early Finish = ES + Duration

An example of a Forward Pass calculation is shown in Figure 3. Generic activities are used in the examples in this paper to help illustrate the different calculation processes.

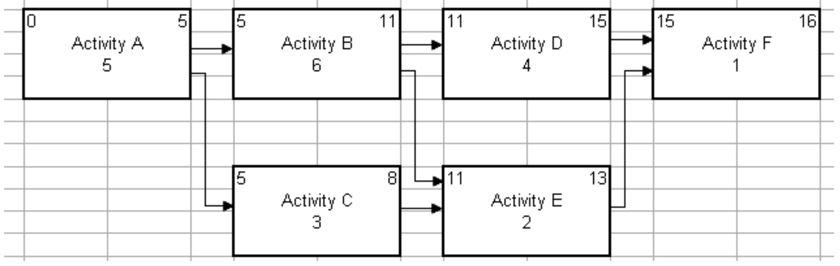


Fig. 3 : AON Forward Pass.

In the diagram shown in Figure 3, note that the early start value for Activity E is based on the maximum early finish from its two predecessors (Activity B and Activity C). Based on the Finish-to-Start relationships shown, both Activity B and Activity C must both be finished before Activity E can start. In other words, Activity E is waiting on whichever predecessor finishes the latest (in this case, Activity B). The same situation exists for Activity F. The duration of the schedule is shown in the EF of the latest activity in the network diagram. Figure 3 shows an EF value of "16" for Activity F. Therefore, assuming that the time-unit of the example schedule is based on days, the schedule duration is sixteen days.

Q.3(f) Draw a Gantt chart and explain it. [5]

Ans.: One of the simplest and oldest techniques for tracking project progress is the Gantt chart. This is essentially an activity bar chart indicating scheduled activity dates and durations, frequently augmented with activity.

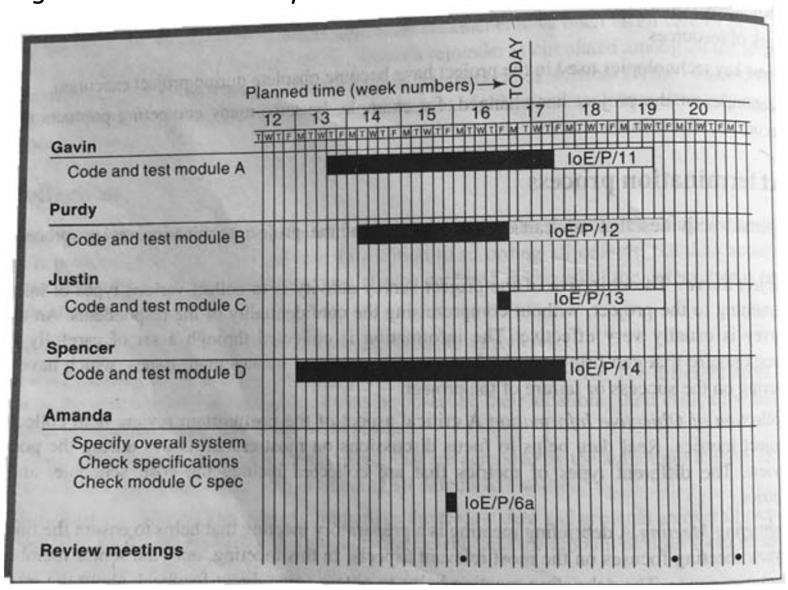


Fig.: Part of Amanda's Gantt chart with the 'today cursor' in week 17

Q.4Attempt the following (any THREE) [15]

Q.4(a) Write a short note on SCM. [5]

Ans.: Software configuration management (SCM) 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 acronym "SCM" is also expanded as source configuration management process and software change and configuration management. However, "configuration" is generally understood to cover changes typically made by a system administrator. The purpose of SCM is :

- **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.

Q.4(b) Explain the different types of contracts. [5]

Ans.: The external resources required could be in the form of services. A simple example of this could be using temporary staff on short term contracts to carry out some project tasks. At Brightmouth College, Brigette could use temporary staff to type into the computer system the personnel details needed to set up the payroll standing data for the new system, while at IOE a decision might be made to carry out the required system building in-house but to augment the permanent staff with contract programmers for the duration of the project. A more far-reaching use of external services would be for the contractor not only to supply the new system but to also operate it on the customer's behalf. For example, it might well be worth Brightmouth College abandoning the idea of buying a package and instead getting a payroll services agency to carry out all the payroll work on their behalf.
On the other hand, the contract could be placed for the supply of a completed software application.

This could be :
- a bespoke system, that is, a system that is created from scratch specifically for one customer,
- off-the-shelf which you buy *as is* - this is sometimes referred to as shrink-wrapped software;
- **customized off-the-shelf (COTS) software** - this is a basic core system, which is modified to meet the needs of a particular customer.

Another way of classifying contracts is by the way that the payment to suppliers is calculated. We will look at :

1. **Fixed Price Contracts :** As the name implies, in this situation a price is fixed when the contract is signed. The customer knows that, if there are no changes in the contract terms, this is the price to be paid on the completion of the work. In order for this to be effective, the customer's requirement has to be known and fixed at the outset. In other words, when the contract is to construct a software system, the detailed requirements analysis must already have been carried out. Once the development is under way, the customer will not be able to change their requirements without renegotiating the price of the contract. The advantages of this method are the following :
   - Known customer expenditure If there are few subsequent changes to the original requirements, then the customer will have a known outlay.
   - Supplier motivation like supplier has a motivation to manage the delivery of the system in a cost-effective manner.

2. **Time and materials contracts :** With this type of contract, the customer is charged at a fixed rate per unit of effort, for example, per staff-hour. At the start of the project, the supplier normally provides an estimate of the overall cost based on their current understanding of the customer's requirements, but this is not the basis for the final payment. The advantages of this approach are the following :
   - Changes to requirements are dealt with easily. Where a project has a research orientation and the direction of the project changes as options are explored, then this can be an appropriate method of calculating payment.
   - Tack of price pressure : The lack of price pressure can allow better quality software to be produced.

3. **Fixed price per delivered unit contracts :** The size of the system to be delivered might be estimated in lines of code, but FPs can be more easily and reliably derived from requirements documents. A price per unit is also quoted. The final price is then the unit price multiplied by the number of units. The advantages of this approach are as follows :

- **Customer understanding :** The customer can see how the price is calculated and how it will vary with changed requirements.
- Comparability Pricing schedules can be compared.
- **Emerging functionality :** The supplier does not bear the risk of increasing functionality.
- **Supplier efficiency :** The supplier still has an incentive to deliver the required functionality in a cost-effective manner (unlike with time and materials contracts).
- **Life cycle range :** The requirements do not have to be definitively specified at the outset. Thus the development contract can cover both the analysis and design stages of the project.

Q.4(c) Explain the expectancy theory of motivation. [5]

Ans.: The three influences on motivation are :

- **expectancy**, the belief that working harder will lead to a better performance;
- **instrumentality**, the belief that better performance will be rewarded;
- **perceived value**, of the resulting reward.

Motivation will be high when all three factors are high. A zero level for any one of the factors can lead to a lack of motivation.

Imagine that you are trying to get a software package supplied by a third party to work. If you realize that you will never get it to work because of a bug in it, you will give up. No matter how hard you work you will not be able to do any better (zero expectancy).

If you are working on a package for a user and, although you think you can get it to work, you discover that the user has started employing an alternative package and no longer needs this one, then you will probably feel you are wasting your time and give up (zero instrumentality).
Given that the users really do want the package, your reward in this set of circumstances might simply be a warm feeling that you have helped your colleagues and that they are grateful to you. If in fact, when the users employ the package all they do is complain and hold you responsible for any shortcomings, then you will probably avoid getting involved if they later ask for help implementing a different package (low perceived value of reward).

Q.4(d) Explain the Oldham-Hackman job characteristics model. [5]

Ans.: Managers should try to group together the elements of the tasks that need to be carried out so that they form meaningful and satisfying assignments. Oldham and Hackman suggest that the satisfaction that a job gives is based on five factors. The first three factors make the job 'meaningful' to the person who is doing it :

- skill variety, the number of different skills that the job holder has the opportunity to exercise
- task identity, the degree to which your work and its results are identifiable as belonging to you;
- task significance, the degree to which your job has an influence on others. The other two factors are:
  - autonomy, the discretion you have about the way that you do the job.
  - feedback, the information you get back about the results of your work.

Couger and Zawacki found that programmers in general rated their jobs lower on these factors than other professions, while systems analysts and analyst-programmers rated them higher. Computer development people experienced about the same level of meaningfulness in their work as other, non-IT, professionals, but had lower perceptions of the degree of responsibility and knowledge of results of their work.

Cheney found that in the programming environment, the degree to which programmers got feedback on their work and the degree to which they could contribute to decision making had positive influences on both productivity and job satisfaction, although 'consideration', which was 'the degree to which the leader develops a work climate of psychological support, mutual trust and respect, helpfulness and friendliness', rated as less important.

In practical terms, activities should be designed so that, where possible, staff follow the progress of a particular product and feel personally associated with it.

Q.4(e) What do you understand by Organizational Behaviour. [5]

Ans.: Taylor attempted to analyse the most productive way of doing manual tasks. The workers were then trained to do the work in the way.
Taylor had three basic objectives.
- To select the best people for the job;
- To instruct them in the best methods;
- To give incentives in the form of higher wages to the best works.

'Taylorism' is often represented as crude and mechanistic. However, a concern for identifying best practice is valid. In the more mundane world of software development, the growth of both structured and agile methods is an example of an emphasis on best practice. Both Amanda and Brigette will be concerned that tasks are carried out in the proper way. More contentious is Taylor's emphasis on the exclusively financial basis of staff motivation, although Amanda and Brigette will find many colleagues who hold Taylor's view on the importance of 'performance-related pay'. Unfortunately Amanda and Brigette are likely to have very little control over the financial rewards of their staff. However, they should be encouraged by findings that motivation rests not just on such rewards.

During the 1920s, OB researchers discovered, while carrying out a now famous set of tests on the conditions under which staff worked best, that not only did a group of workers for whom conditions were improved increase their work rates, but also a control group for whom conditions were unchanged. Simply showing a concern for what workers did increased productivity. This illustrated how the state of mind of workers influenced their productivity.

The cash-oriented, or instrumental, view of work of some managers can thus be contrasted with a more rounded vision of people in their place of work. The two attitudes were labeled Theory X and Theory Y by Donald McGregor.

Theory X holds that:
- The average human has an innate dislike of work.
- There is a need therefore for coercion, direction control;
- People tend to avoid responsibility.

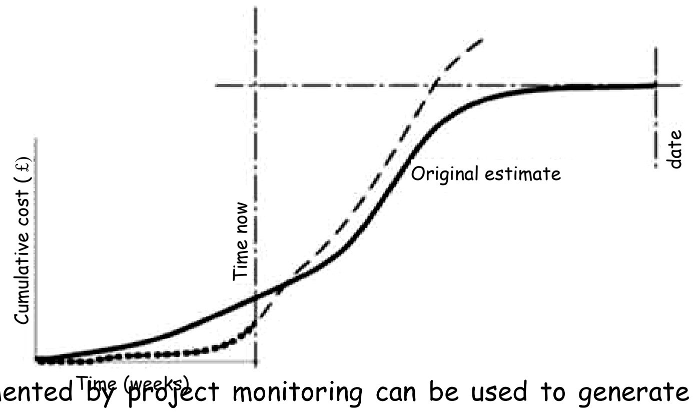
Theory Y, on the other hand, holds that:
- Work is as natural as rest or play;
- External control and coercion are not the only ways of bringing about effort directed towards an organization's ends;
- Commitment to objectives is a function of the rewards associated with their achievement;
- The average human can learn to accept and further seek responsibility;
- The capacity to exercise imagination and other creative qualities is widely distributed.

Q.4(f) Explain Earned Value Analysis. **[5]**

Ans.: Earned Value Analysis, also known as Budgeted Cost of Work Performed, is recommended by a number of agencies including the US and Australian departments of defence.

Earned Value Analysis has gained in popularity in recent years and may be seen as a refinement of the cost monitoring. Earned Value Analysis is based on assigning a 'value' to each task or work package based on the original expenditure forecasts. The assigned value is the original budgeted cost for the item and is known as the baseline budget or budgeted cost of work scheduled (BCWS). A task that has not started is assigned the value zero and when it has been completed, it. and hence the project, is credited with the value of the task. The total value credited to a project at any point is known as the earned value or budgeted cost of work performed (BCWP) and this can be represented as a value or as a percentage of the BCWS.

Original Total Cost



Revised Total Cost :

- Project costs augmented by project monitoring can be used to generate forecasts of future costs.
- The cumulative expenditure chart can also show revised estimates of cost and completion date.
- Where tasks have been started but are not yet complete, some consistent method of assigning an earned value must be applied. Common methods in software projects are :
  - the 0/100 technique Where a task is assigned a value of zero until such time that it is completed when it is given a value of 100*£ of the budgeted value;
  - the 50/50 technique Where a task is assigned a value of 50% of its value as soon as it is started and then given a value of 100# once it is complete;
  - the milestone technique Where a task is given a value based on the achievement of milestones that have been assigned values as part of the original budget plan.
- Of these, we prefer the 0/100 technique. The 50/50 technique can give a false sense of security by over-valuing the reporting of activity starts. The milestone technique might be appropriate for activities with a long duration estimate but in such cases, it is better to break that activity into a number of smaller ones.

The Baseline Budget

The first stage in setting up an earned value analysis is to create the baseline budget. The baseline budget is based on the project plan and shows the forecast grow.th in earned value through time. Earned value may be measured in monetary values but, in the case of staff-intensive projects such as software development, it is common to measure earned value in person-hours or workdays.

- **Budget variance :** This can be calculated as ACWP - BCWS and indicates the degree to which actual costs differ from those planned.
- **Schedule variance :** The schedule variance is measured in cost terms as BCWP-BCWS and indicates the degree to which the value of completed work differs from that planned. Figure below also indicates the schedule variance in time, which indicates the degree to which the project is behind schedule.

- **Cost variance :** This is calculated as BCWP - ACWP and indicates the difference between the budgeted cost and the actual cost of completed work. It is also an indicator of the accuracy of the original cost estimates.

Performance ratios Two ratios are commonly tracked: the cost performance index (CPI = BCWP/ACWP) and the schedule performance index (SPI = BCWP/ BCWS). They can be thought of as a 'value-for-money* indices. A value greater than one indicates that work is being completed better than planned whereas a value of less than one means that work is costing more than and/or proceeding more slowly than planned.

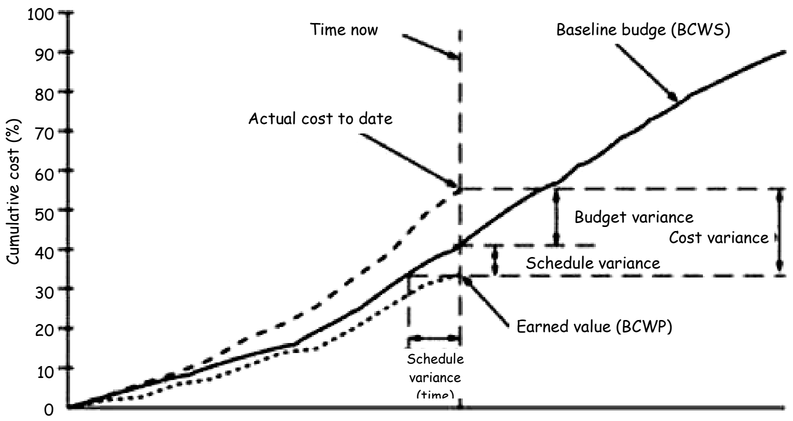


Fig.: An earned value tracking chart.

Q.5Attempt the following (any THREE) **[15]**

Q.5(a) Explain ISO 9126 in detail. **[5]**

Ans.: **ISO 9126 identifies six software quality characteristics :**

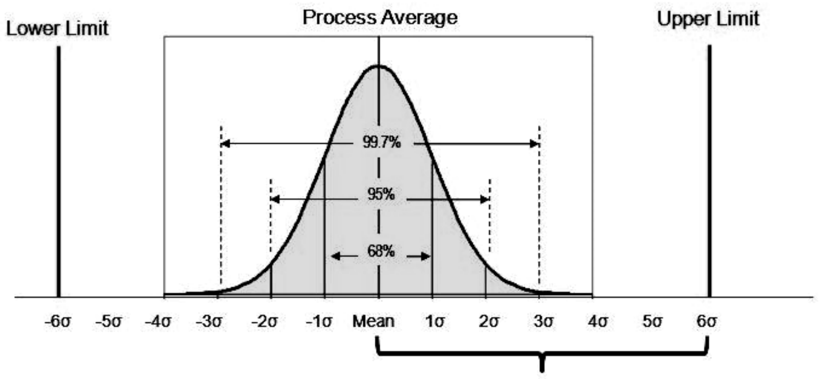
- **functionality**, which covers the functions that a software product provides to satisfy user needs
- **reliability**, which relates to the capability of the software to maintain its level of performance
- **usability**, which relates to the effort needed to use the software
- **efficiency**, which relates to the physical resources used when the software is executed
- **maintainability**, which relates to the effort needed to the make changes to the software
- **portability**, which relates to the ability of the software to be transferred to a different environment

ISO 9126 suggests sub-characteristics for each of the primary characteristics. It is perhaps indicative of the difficulties of gaining widespread agreement that these sub-characteristics are outside the main standard and are given in the document for information only. They are useful as they clarify what is meant by the main characteristics.

Q.5(b) Write a short note on Six Sigma. **[5]**

Ans.: SixSigma is a disciplined, statistical-based, data-driven approach and continuous improvement methodology for eliminating defects in a product, process or service. It was developed by Motorola in early to middle 1980’s based on quality management fundamentals, and then became a popular management approach at General Electric (GE) in the early 1990’s. Hundreds of companies around the world have adopted Six Sigma as a way of doing business.

Sigma represents the population standard deviation, which is a measure of the variation in a data set collected about the process. If a defect is defined by specification limits separating good from bad outcomes of a process, then a six sigma process has a process mean (average) that is six standard deviations from the nearest specification limit. This provides enough buffer between the process natural variation and the specification limits.



For example, if a product must have a thickness between 10.32 and 10.38 inches to meet customer requirements, then the process mean should be around 10.35, with a standard deviation less than 0.005 (10.38 would be 6 standard deviations away from 10.35).

Six Sigma can also be thought of as a measure of process performance, with Six Sigma being the goal, based on the defects per million. Once the current performance of the process is measured, the goal is to continually improve the sigma level striving towards 6 sigma. Even if the improvements do not reach 6 sigma, the improvements made from 3 sigma to 4 sigma to 5 sigma will still reduce costs and increase customer satisfaction.

Sigma Level	Defects per Million	Yield
6	3.4	99.99966%
5	230	99.977%
4	6,210	99.38%
3	66,800	93.32%
2	308,000	69.15%
1	690,000	30.85%

Q.5(c) Write a short note on CMM. **[5]**

Ans.: The capability maturity model (CMM) is an assessment model developed by the Software Engineering Institute at Carnegie Mellon University in 1990, to ascertain the process maturity levels in the software.

The model describes five levels of best engineering and management practices based on data collected from various industries. Organizations looking for solutions to manage projects aim to attain CMM certification by compliance with the CMM level closest to their level of process maturity.

CMM became popular as it allowed software companies attain process consistency, predictability, and reliability. However, the implementation of CMM led to many hurdles.

1. **Lack of Integration:** CMM has separate models for each function. Such models often overlap, contradict, and display different levels of maturity. This lack of standardization leads to confusion and conflict during the implementation phase and increase training and appraisal costs.
2. **Limitations of KPA: The** "Key Performance Areas (KPA)," that define CMM levels focus on "policing" activities such as specifications, documentation, audits, and inspections, and do not reveal architecturally significant flaws.
3. **Activity-based Approach:** CMM is an activity-based approach that considers only the completion of a specific activity, and not whether the completed activity achieved the desired results.

4. **Paperwork:** CMM places great importance on paperwork and meetings that take management's time and effort away from actual work processes. CMM traps the organization in recording and complying with processes, often at the cost of strategic goals.

CMMI :

The Software Engineering Institute at Carnegie Mellon University developed Capability Maturity Model Integration (CMMI) in 2006 to integrate and standardize the separate models of CMM, and to eradicate other drawbacks of CMM.

CMMI documents industry best practices categorized on separate areas of interests rather than separate functions. Organizations choose from any of the 22 available models depending on the business objectives, and each model covers all the functional areas.

- **Level 1 (Initial):** The first level of both CMM and CMMI describes an immature organization without any defined processes, run in an ad hoc, uncontrolled, and reactive manner.
- **Level 2 (Repeat):** Organizations that repeat some processes attain Level 2 CMM. Level 2 of CMMI however requires management of organizational requirements through planned, performed, measured, and controlled processes.
- **Level 3 (Defined):** CMM Level 3 mandates a set of documented standard processes to establish consistency across the organization. CMMI Level 3 is an improvement of CMMI Level 2 and describes the organizational processes in standards, procedures, tools, and methods.
- **Level 4 (Manage):** CMM Level 4 requires organizations to attain control over processes by using quantitative statistical techniques. CMMI Level 4 demands likewise, but also identifies sub processes that significantly contribute to overall process efficiency.
- **Level 5 (Optimized):** CMM Level 5 mandates use of quantitative tools and objectives to manage process improvement. CMMI Level 5 on the other hand focuses on continuously improving process performance through incremental and innovative technological improvements.

While CMM is a certification tool, CMMI is not. An organization is appraised and awarded a CMMI Rating from 1 to 5 depending on the extent to which the organization adopts the selected CMMI model.

Q.5(d) What are the reasons for Project Closure. [5]

Ans.: 1. **Expensive or does not meet company's goal**
Make an estimate of the total cost of the project in the planning stage itself. A few thousand dollars here and there are manageable, but when you see the figure going way over your approximate value, it is better to put an end to the project right in the initiation stage. Also, if the project does not go well with the strategic plan of the company, it should not be given the green signal.

2. **Your competitors are doing a better job**
As a project manager, you may be motivated to prove your mettle and take your company ahead in the market, but think logically and determine if it is possible. Many a times, you may be motivated at the start of the project but once you begin with it and have to face grave challenges one after another, the positive drive may fizzle out and you may be left with a project that is going nowhere. Even if you realize it midway on the project, do not hesitate to pull the plug.

3. **Project gets out of control**
When operations get way beyond control or when damages cannot be repaired anymore, you know it is time to terminate the project.

4. **Important or priority project comes up**
   Businesses take up several projects simultaneously. However, there are some projects which need more time, energy and resources. If a certain project is stopping you from allocating the required resources in a bigger, important project, it is better to let go of the smaller project.

5. **Failure in testing process**
   It is sad to see a project fail during testing. However, if the team members gave it all that they could and the project still could not succeed, putting an end to the project is a sensible choice rather than spending twice the energy and resources on it again.

Q.5(e) Write a short note on Quality Plans. [5]

Ans.: Some organizations produce quality plans for each project. These show how the standard quality procedures and standards laid down in an organization's quality manual will actually be applied to the project. If an approach to planning such as Step Wise has been followed, quality-related activities and requirements will have been identified by the main planning process with no need for a separate quality plan. However, where software is being produced for an external client, the client's quality assurance staff might require that a quality plan to produced to ensure the quality of the delivered products. A quality plan can be seen as a checklist that all quality issues have been dealt with by the planning process. Thus, most of the content will be references to other documents.

A quality plan might have entries for:

- purpose –scope of plan;
- List of references to other documents;
- Management arrangements, including organization, tasks and responsibilities;
- Documentation to be produced;
- Standards, practices and conventions;
- Reviews and audits;
- Testing;
- Problem reporting and corrective action;
- Tools, techniques and methodologies;
- Code, media and supplier control;
- Records collection, maintenance and retention;
- Training;
- Risk management-the methods of risk management that are to be used.

Q.5(f) Explain the different kinds of people needed in a team suggested by Belbin. [5]

Ans.:

- The chair: not necessarily brilliant leaders but they must be good at running meetings, being calm, strong but tolerant.
- The plant: someone who is essentially very good at generating ideas and potential solutions to problems.
- The monitor-evaluator: good at evaluating ideas and potential solutions and helping to select the best one.
- The shaper: rather a worrier, who helps to direct the team's attention to the important issues.
- The team worker: skilled at creating a good working environment, for example, by 'jollying people along'.
- The resource investigator: adept at finding resources in terms of both physical resources and information.
- The completer-finisher: concerned with completing tasks.
- The company worker: a good team player who is willing to undertake less attractive tasks if they are needed for team success.

❑ ❑ ❑ ❑ ❑