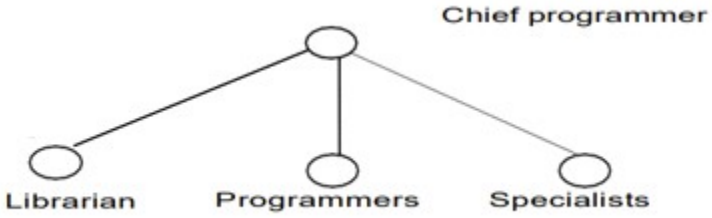
19CS7602 - SOFTWARE PROJECT MANAGEMENT

Question Bank

PART – A

Q.No.	Question
1.	**Interpret the three different approaches for identifying the activities.** Essentially there are three approaches to identifying the activities or tasks that make up a project • The activity-based approach, • The product-based approach • The hybrid approach.
2.	**How estimation is done in PERT.** Project Evaluation and Review Technique (PERT) is a project management tool used to schedule, organize, and coordinate tasks within a project. It estimation considers three values: the most optimistic estimate (O), a most likely estimate (M), and a pessimistic estimate (least likely estimate (L))
3.	**What is the use of check points in monitoring?** It is essential to set a series of checkpoints in the initial activity plan. Checkpoints may be regular, tied to specific events such as production of a report or other deliverable.
4.	**Classify the categories of reporting?** • Oral formal regular: Weekly or monthly progress meetings • Oral formal ad hoc: End-of-stage meetings • Written formal regular: job sheets, progress reports • Written formal ad hoc: Exception reports, change reports • Oral informal ad hoc: Canteen discussion, social interaction
5.	**Draw the project reporting structures**. 
6.	**How to visualize the project progress**. • Gantt chart • Slip chart • Timeline • Ball chart
7.	**Why Software Configuration Management in a project is essential?** Within a project, the purpose of configuration management is to identify, track and protect the project's deliverables or products from unauthorised change.
8.	**Discuss the time and materials contracts.** In time and material contracts the customer is charged at a fixed rate per unit of effort, for ex per staff-hour. The supplier may provide an initial estimate of the cost based on their current understanding of the customer's requirements, but this is not the basis for final payment.
9.	**How do you select the right persons for the job?** • Determining your need to hire a new employee. • Conducting a thorough job analysis. • Writing a job description and job specification for the position based on the job analysis. • Determining the salary for the position, based on internal and external equity.

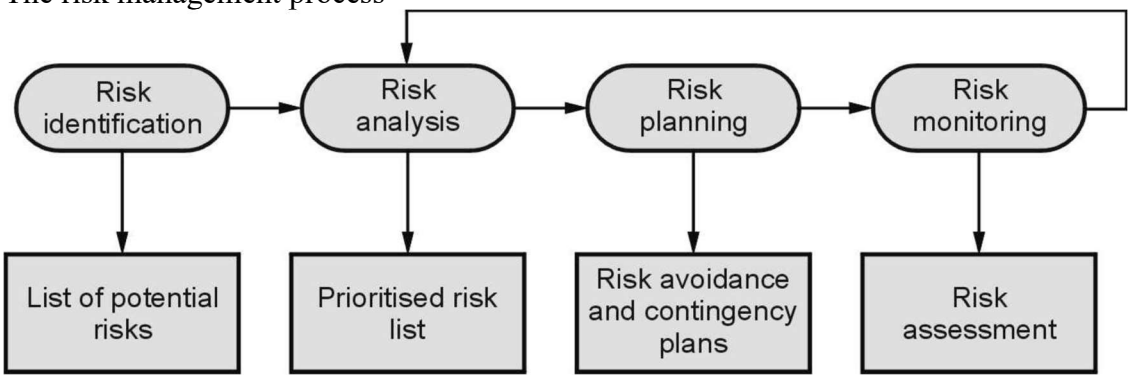
	• Deciding where and how to find qualified applicants. • Collecting and reviewing a fair amount of applications and resumes and then selecting the most qualified candidates for further consideration.
10.	**Illustrate the measures to reduce the disadvantages of group decision making?** • The cooperation of a number of experts. • The problem is presented to the experts. • The experts record their recommendations. • These recommendations are collated and reproduced. • The collect responses are recirculated
11.	**Interpret the role of ethics in project management?** Honesty, responsibility, respect and fairness are the values that drive ethical conduct for the project management profession. A code of ethics is to ensure decisions and actions are always honorable and in the best interest of stakeholders.
12.	**Draw the chief programmer team structure.** 
13.	**Distinguish between Storming and Forming**. Forming: The initial stage is the forming stage. At this point, team members are generally concerned about their role in the team and who calls the shots. As a project manager, you will have a dominant role in team building and people will be looking to you for guidance and reassurance. This stage can be relatively short in comparison to the other stages and may only last a meeting or two. Storming: At this stage, the team addresses the problems they are going to solve and how they are going to function as a unit. Team members will become more open with each other as they express their own ideas and thoughts and will often confront the project manager about certain aspects of the project
14.	**Differentiate between CPM and PERT**.

Basic	PERT	CPM
Stands for	PERT stands for **"Program Evaluation and Review Technique".**	CPM stands for **"Critical Path Method".**
Model	It is a **probabilistic** model under which the result estimated in a manner of probability.	It is a **deterministic** model under which the result is ascertained in a manner of certainty.
Time	It deals with the activities of **uncertain** time.	It deals with the activities of precise well known time.
Jobs	It is used for onetime projects that involve activities of **non-repetitive** nature.	It is used for completing of projects that involve activities of **repetitive** nature.
Orientation	It is **activity oriented** in as much as its result is calculated on the basis of the activities.	It is **even oriented,** in as much as its results are calculated on the basis of the events.
Dummy Activities	It does **not** make use of **dummy activities.**	It makes use of **dummy activities** to represent the proper sequencing of the activities.
Cost	It has nothing to do with cost of a project.	It deals with the cost of a project schedules and their minimization.

15.	**Define software reliability**. • Software reliability is the probability of failure-free software operation for a specified period of time in a specified environment. • Software reliability is also an important factor affecting system reliability. • It differs from hardware reliability in that it reflects the "design perfection ", rather than manufacturing perfection. • The high complexity of software is the major contributing factor of software reliability problems.
16.	**Mention Resource Allocation Issues.**

	• Availability • Criticality • Risk • Training • Team Building
17.	**Mention Resource Scheduling Issues** • Human resource scheduling issues, o Planned Leave, Public Holidays. o Possible sick leave (random, subjective at best and hard to predict). o General motivation and enthusiasm for the task allocated (If they dislike the task, it will flow through into the output). o Work load and stress in project. o Stress outside work.

Big Questions

1.	**Interpret the use of checklist and brainstorming in identification of risks.** Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project. A risk is a probability that some adverse circumstance will occur. o Project risks affect schedule or resources o Product risks affect the quality or performance of the software being developed o Business risks affect the organisation developing or procuring the software The Risk Management Process Risk identification o Identify project, product and business risks Risk analysis o Assess the likelihood and consequences of these risks Risk planning o Draw up plans to avoid or minimise the effects of the risk Risk monitoring o Monitor the risks throughout the project The risk management process  **Risk identification** Identify the hazards that might affect the duration or resource costs of the project **Hazard → Problem → Risk** A *hazard* is an event that might occur and will create a problem for the successful completion of the project, if it does occur Examples of hazard : a team member is ill; late delivery of a hardware component; system down **Hazard, Problem, and Risk** ***Hazard*** **:** Mary's baby may be born early ***Problem*** **:** Modules P and Q will have no coder ***Risk*** **:** Milestone 7 will be delayed, or extra budget will be needed to hire another coder **Type of Risk-1** **Type of risks** o *Generic risk* (common to all projects)

Standard checklist can be modified based on the risk analysis of previous projects
o *Specific risk* (only applies to individual projects)
 More difficult to find
 Need to involve project team members
 Need an environment that encourages risk assessment
 Generic risks are those risks relevant to all software projects.
o Examples are misunderstanding of the requirements and key staff being ill.
 Specific risks are those risks relevant to an individual project only.
o Team members of the project are the frontline of identifying these potential risks.
o Need to set up an encouraging risk-identification environment so that team members are willing to share their findings.
o "***Assuming that the problems will not occur does not prevent their occurrence.***"
 Guideline
o Use checklist that lists the potential hazards and their corresponding factors
o Maintain an updated checklist for future projects

Type of Risk-2

 Technology risks
 People risks
 Organisational risks
 Requirements risks
 Estimation risks

Risks and risk types

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. Software components which should be reused contain defects which limit their functionality.
People	It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Organisational	The organisation is restructured so that different management are responsible for the project. Organisational financial problems force reductions in the project budget.
Tools	The code generated by CASE tools is inefficient. CASE tools cannot be integrated.
Requirements	Changes to requirements which require major design rework are proposed. Customers fail to understand the impact of requirements changes.
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

2. **Describe with an example how the effect of risk on project schedule is evaluated using PERT.**

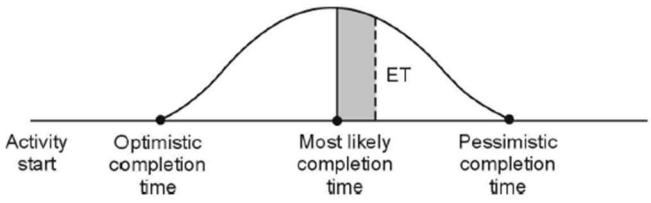
Program Evaluation and Review Technique (PERT) :
o Developed to manage the Polaris missile project.
o Many tasks pushed the boundaries of science and engineering (tasks' duration = probabilistic).
Graphically display the precedence relationships and sequence of activities.
 Estimate the project's duration.

	□ Identify critical activities that cannot be delayed without delaying the project. □ Estimate the amount of slack associated with non-critical activities.

Activity	Description	Optimistic time	Most likely time	Pessimistic time
A	Develop product specifications	2	4	6
B	Design manufacturing process	3	7	10
C	Source and purchase materials	2	3	5
D	Source and purchase tooling and equipment	4	7	9
E	Receive and install tooling and equipment	12	16	20
F	Receive materials	2	5	8
G	Pilot production run	2	2	2
H	Evaluate product design	2	3	4
I	Evaluate process performance	2	3	5
J	Write documentation report	2	4	6
K	Transition to manufacturing	2	2	2

Using beta probability distribution to calculate expected time durations

- A typical beta distribution is shown below, note that it has definite end points.
- The expected time for finishing each activity is a weighted average.



$$\text{Exp. time} = \frac{\text{Optimistic} + 4(\text{most likely}) + \text{pessimistic}}{6}$$

Calculating Expected Task Times

$$\text{Exp. time} = \frac{\text{Optimistic} + 4(\text{mos likely}) + \text{pessimistic}}{6}$$

Activity	Optimistic time	Most likely time	Pessimistic time	Expected time
A	2	4	6	4
B	3	7	10	6.83
C	2	3	5	3.17
D	4	7	9	6.83
E	12	16	20	16
F	2	5	8	5
G	2	2	2	2
H	2	3	4	3
I	2	3	5	3.17
J	2	4	6	4
K	2	2	2	2

3.	**Derive critical path for completion of the project. Based on task dependency diagram allocate resource and derive Activity Chart.**

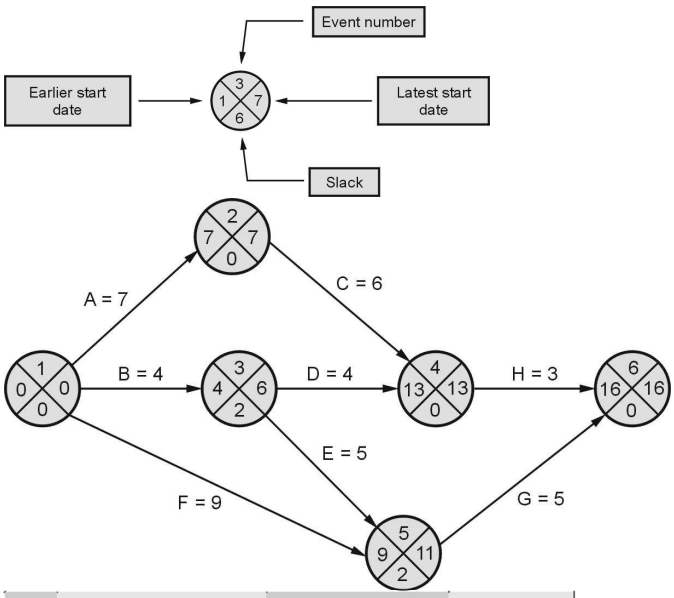
- Primary objectives :
  - Planning the project so that it can be completed as quickly as possible.
  - Identifying those activities where their delays is likely to affect the overall project completion date
- Developed by Du Pont Chemical Company and published in 1958.
- Capture the activities and their inter-relationships using a graph.
  - Lines are used to represent the activities.
  - Nodes are used to represent the start and stop of activities.
- Dummy activities (dotted lines) can be used to :
  - Avoid logical errors to paths.
  - Document related activities that can be done in parallel and have a time lag
- In CPM, each activity has a time estimate.
- Adding time dimension
  - The forward pass

1. calculate the earliest start dates of the activities
2. to calculate the project completion date

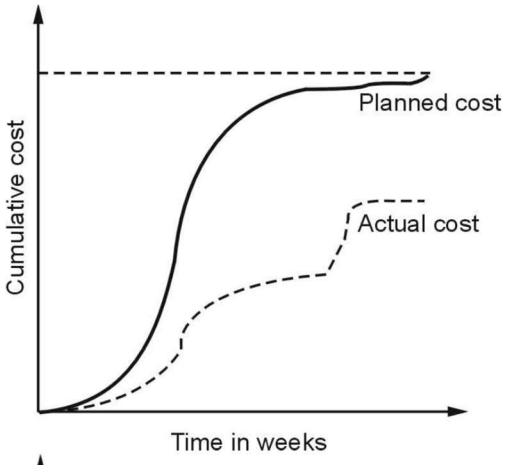
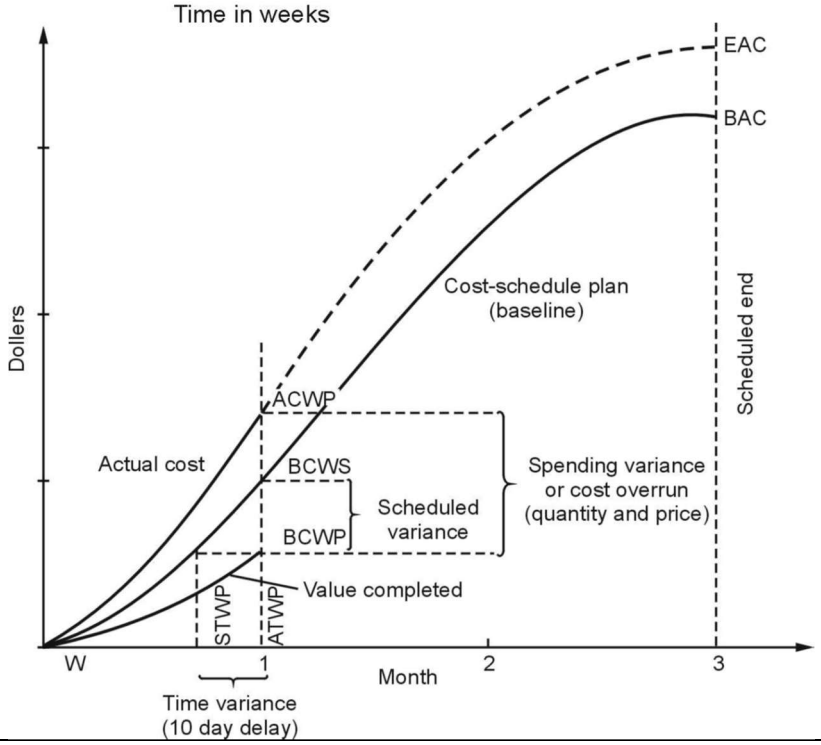
  - The backward pass

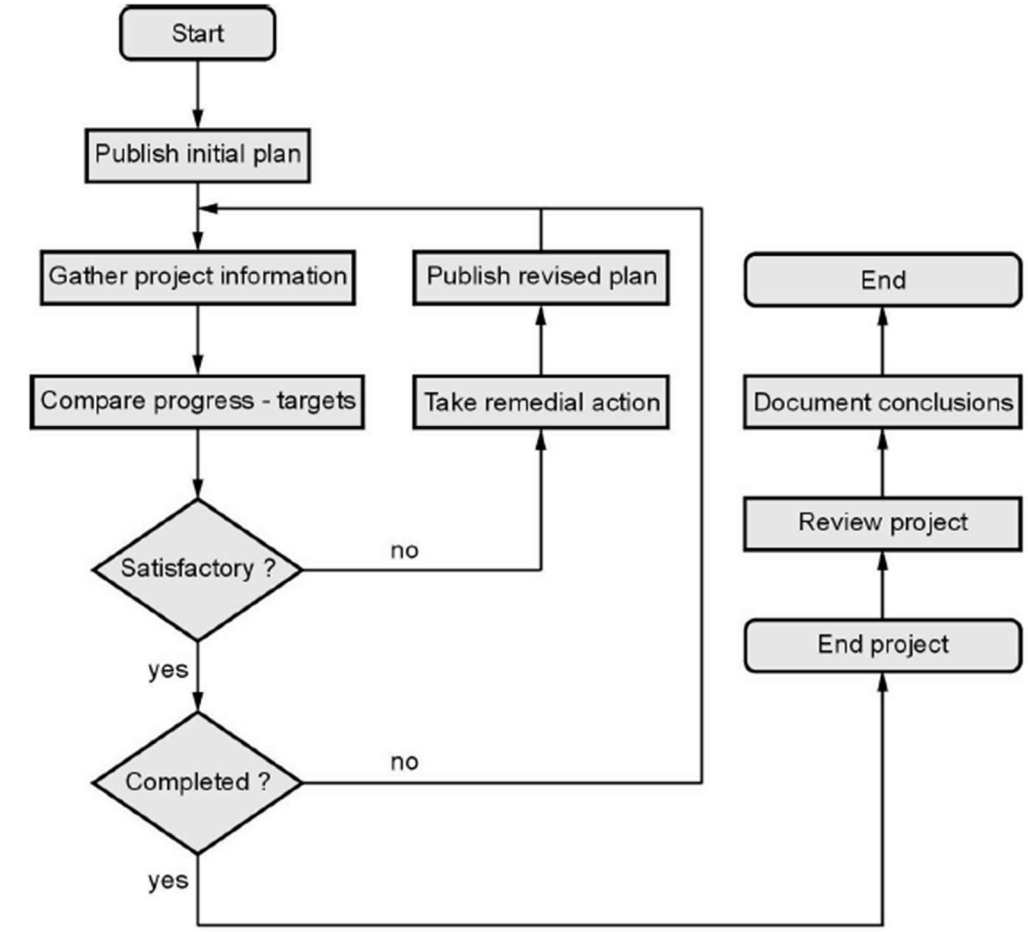
1. calculate the latest start dates for activities
2. identify the critical path from the graph

- Identifying critical path and critical event
  - **Critical event :** an event that has zero *slack*
  - **Critical path :** a path joining those critical events
- Slack : measures how late an event may be without affecting the overall target date of the project
  - slack = latest start date – earliest start date (for an event)
- Any delay of the critical events will delay the project
- There is always at least one critical path in the CPM network.



Id.	Activity Name	Duration (weeks)	Precedents
A	Hardware selection	7	
B	Software design	4	
C	Hardware Installation	6	A
D	Coding	4	B
E	Data Preparation	5	B
F	User Documentation	9	
G	User Training	5	E,F
H	System Installation	3	C,D

4.	**Give the importance of cost monitoring in detail.** □ It is an important component of project control. Not only in itself, but also because it provides an indication of the effort that has gone into (or at least been charged to) a project. □ It provides a simple method of comparing actual and planned expenditure. □ The more cost is incurred to complete the activities to keep the project on schedule. □ The chart does a comparison between the actual and the planned expenditure. □ Cost charts become much more useful if we add projected future costs calculated by adding the estimated costs of uncompleted work to the costs already incurred. □ Where a computer-based planning tool is used, revision of cost schedules is generally provided automatically once actual expenditure has been recorded.  
5.	**Analyze the methods used for project tracking.** **Project Tracking** □ Finding out what is happening □ Need a plan and schedule □ To collect data □ Ideally, the plan and schedule should be absolutely clear and highly visible to every personnel involved in the project. **The project control cycle**



The overall responsibility for ensuring satisfactory progress on a project is often the role of the project steering committee or project board.

 Categories of reporting are classified as, Formal and Informal.

 Formal regular types can be oral or written.

 Standard oral communication of minutes are kept where as written type gets the reporting issues in a separate written format.

 Formal ad hoc are mostly received information of different levels towards the end of the project and generate written reports.

 Informal, oral and ad hoc provides early warning to the system and must be backed up by formal reporting procedures.

Tracking the Performance

 Tracking performance means assessing the progress

o Need to trust your personnel to give an objective assessment of their work and their estimation on completion of their work

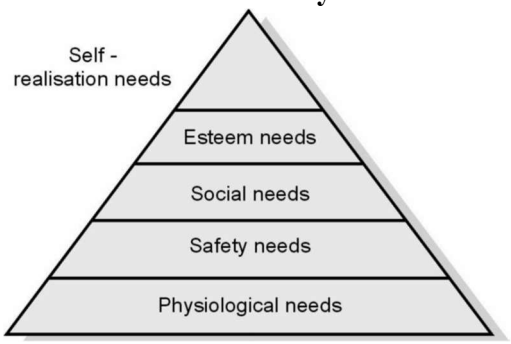
 Setting check points

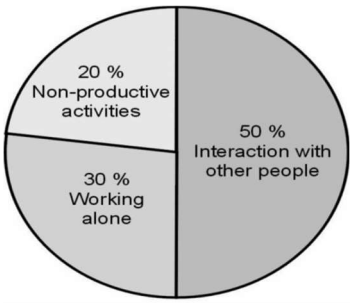
 Collecting data

Check Point

 Based on regular time intervals

	o Can be weekly or monthly or quarterly o Depend on what to check and how to Based on a particular event o At the end of each activity o In the middle of a critical activity **Time interval:** Duration actually depends on who to check, what to check, the degree of risk of the project; how familiar the new employee is to the process of the organization. **Critical activity:** The number of check points depends on how critical the event is. Tied to specific events such as the production of a report. Should be set before the plan was published.
6.	**Explain the types of contracts with example.** **1. Fixed Price Contracts :** These are also known as Lump Sum contracts. The seller and the buyer agree on a fixed price for the project. The seller is bound to accept high risk in this type of contract. The buyer is in the least risk category as the price is already fixed and the seller has agreed to this. There must be fully detailed specifications, checklists, project scope statements from the seller side which the buyer will use. Often, sellers may try to cut the scope to deliver the projects on time and within budget. If the project is finished on time with the desired quality, the project is over for that contract. However, if the project is delayed and there are cost overruns, then the seller will absorb all the extra costs. Fixed price contracts are typically used in government based projects. **Advantages** of fixed price contracts include : o Minimizing risk for buyers. o Known customer expenditure o Supplier motivation The major **disadvantage** of Fixed Price Contracts is that o The seller starts cutting scope in order to finish on time and within budget. o Higher prices to allow for contingency o Difficulties in modifying requirements o Upward pressure on the cost of changes o Threat to system quality **Below are a few types of fixed contracts :** ***Fixed Price Incentive Fee (FPIF)*** *– If project ends sooner, an additional amount is paid to the seller.* ***Fixed Price Award Fee (FPAF)*** *– If the performance of the seller exceeds expectations, an additional amount (say 10% of the total price) will be paid to the seller.* ***Fixed Price Economic Price Adjustment (FPEPA)*** *– The fixed price can be redetermined depending on the market pricing rate.* **2. Cost Reimbursable Contracts:** **What you will do when the scope of the work is not clear?** Fixed price contracts would be out of the question since you are not sure what you need out of the project. In such cases, ideally you would need to opt for cost reimbursable contracts. Under a cost reimbursable contract, the seller will work for a fixed time period, and will raise the bill after finishing work. A major drawback of this type of contract is that the seller can raise an unlimited or unknown amount which the buyer is compelled to pay. This is why cost reimbursable contracts are rarely used. **Below are a few types of cost reimbursable contracts :**

	□ ***Cost Plus Fee (CPF) or Cost Plus Percentage of Costs (CPPC)*** *– The seller will get the total cost they incurred on the projects plus a percentage of fee over cost. Always beneficial for the seller.* □ ***Cost Plus Fixed Fee (CPFF) –*** *A fixed amount (for seller) is agreed upon before work commences. Cost incurred on the project is reimbursed on top of this.* □ ***Cost Plus Incentive Fee (CPIF)*** *– A performance-based extra amount will be paid to the seller over and above the actual cost they have incurred on the projects.* □ ***Cost Plus Award Fee (CPAF)*** *– The seller will get a bonus amount plus the actual cost incurred on the projects. Very similar to a CPIF contract.* **3. Time and Material Contracts or Unit Price Contracts :** □ Unit price contracts are what we call an hourly rate. □ For example, if the seller spends 1,200 hours on a project, and his or her charges are $100 an hour, the seller will be paid for $120,000 by the buyer. □ This type of contract is typical in freelance work. □ The main **advantage** of this type of contract is that the seller will make money for every hour he spends on the project.
7.	**Discuss about the different models of Motivation.** □ An important role of a manager is to motivate the people working on a project. □ Motivation is a complex issue but it appears that there are different types of motivation based on : o Basic needs (e.g. food, sleep, etc.); o Personal needs (e.g. respect, self-esteem); o Social needs (e.g. to be accepted as part of a group). **Human needs hierarchy**  **Motivating people** □ Motivations depend on satisfying needs □ It can be assumed that physiological and safety needs are satisfied □ Social, esteem and self-realization needs are most significant from a managerial viewpoint **Need satisfaction** □ Social o Provide communal facilities; o Allow informal communications. □ Esteem o Recognition of achievements; o Appropriate rewards. □ Self-realization o Training - people want to learn more; o Responsibility. **Individual motivation** Alice's assistive technology project starts well. Good working relationships develop within the team and creative new ideas are developed. However, some months into the project, Alice notices that Dorothy, the hardware design expert starts coming into work late, the quality of her work deteriorates and, increasingly, she does not appear to be communicating

	with other members of the team. Alice talks about the problem with other team members to try to find out if Dorothy's personal circumstances have changed and if this might be affecting her work. They don't know of anything so Alice decides to talk with Dorothy to try to understand the problem. After denying that there is a problem, Dorothy admits that she seems to have lost interest in the job. She expected a job where she would develop and use her hardware interfacing skills. However, she is basically working as a C programmer with other team members and she is concerned that she is not developing her interfacing skills. She is worried that she will find it difficult to find a job after this project that involves hardware interfacing. Because she does not want to upset the team by revealing that she is thinking about the next project, she has decided that it is best to minimise conversation with them. **Personality types** The needs hierarchy is almost certainly an over-simplification of motivation in practice. Motivation should also take into account different personality types : o Task-oriented; o Self-oriented; o Interaction-oriented. Task-oriented. o The motivation for doing the work is the work itself; Self-oriented. o The work is a means to an end which is the achievement of individual goals - e.g. to get rich, to play tennis, to travel etc.; Interaction-oriented o The principal motivation is the presence and actions of co-workers. People go to work because they like to go to work. **Motivation balance** Individual motivations are made up of elements of each class. The balance can change depending on personal circumstances and external events. However, people are not just motivated by personal factors but also by being part of a group and culture. People go to work because they are motivated by the people that they work with. **Managing Groups** Most software engineering is a group activity o The development schedule for most non-trivial software projects is such that they cannot be completed by one person working alone. Group interaction is a key determinant of group performance. Flexibility in group composition is limited o Managers must do the best they can with available people. **Time distribution** 
8.	**How to deal with ethical and programming concerns in software project management.** The process of developing a new software application takes time and effort. It takes time to design, develop and release the final product. Unfortunately for many software companies and developers, they are given a small window of time and a small budget to release a software package. Software companies mainly its developers are under pressure to release a virtually bug-free product on time at the lowest possible cost. However, they face a lot of obstacles that hinders this goal.

Mostly, the top reasons for software project failure were :

- Project objectives not fully specified
- Bad planning and estimating
- Technology new to the organization
- Inadequate or no project management methodology
- Insufficient senior staff on the team
- Poor performance by suppliers of hardware and/or software

Because of the time and money constraints, as well as the obstacles that they face to make a quality product, software companies and developers are often tempted to perform unethical and illegal acts to make their goal.

There are five ethical issues that software companies and developers face. They are :

- Using open-source code in their own code without properly crediting the source
- Using illegal software to perform their tasks
- Reverse engineering code to find out how a process works
- Not addressing known bugs
- Taking talent from the competition

Ethical Issues

1. Using Open-Source Code: According to the definition on the Open Source Initiative's web site, open source is source code that is readily available to the user. In other words, the application contains the source code that was used to create the product.

There are three particular types of open source code :

- **Licensed Source Code:** The source code may contain a GPL (General Public License) or an LGPL (Library General Public License) that details how the software and the source code is to be distributed, copied and modified.
- **Copyrighted or Credited Source Code:** The source code may be freely published on a web site with the author's consent for the programmer to use the source code as long as the author is credited in the code.
- **Public Domain:** The source code may be in public domain, which means that the author explicitly relinquishes all rights to the software.

2. Using Illegal Software : Due to time and money crunches, it is tempting for a company to use a pirated copy of software or violate the software license. There are some published ethics guides on how employees are supposed to use software. These guides cover points such as :

- The definitions of software licenses
- Penalties those companies and employees will face if they violate copyright laws
- Answers to frequently asked questions about software use

3. Reverse Engineering Code : Reverse engineering is a controversial and a confusing subject in the software development world. Out of all the issues mentioned, this issue frequently creates dilemmas for software engineers and companies. Reverse engineering is the process of decompiling an application in order to reveal the source code. In the early days of software development, many software engineers engaged in the practice of reverse engineering to find out how a particular program performed an action. With the passing of the DMCA, reverse engineering has legal implications. There are issues with reverse engineering that could cause confusion with how to use it.

For example:

- If the software is considered public domain, then the programmer is allowed to reverse-engineer it.
- The DMCA prohibits the act of circumventing a technological measure used by copyright owners to control access to their works. Acts of circumventing include: copying media, decrypting encryption tools, and reverse-engineering software.

4. Non-compete Agreement: A document signed by an employee that promises that the said employee will not work for a direct competitor for a specific amount of time after s/he leaves the company try to prevent talent from going to competitive firms by having its employees sign non-compete agreements.

Ethical Problems

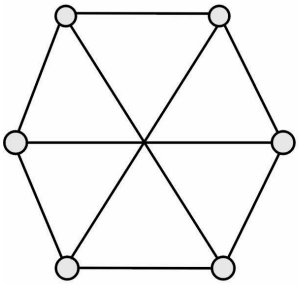
	□ Assign task to a compliance officer to make sure that the licenses are being used properly. □ Perfect quality assurance □ Consult with legal department about non-compete agreements and fair use with reverse engineering □ Let public know about flaws or delay the software release. □ Publish ethical guidelines on software development and use.
9.	**How to make decision while managing the people and organizing teams** Decision can be categorized as **Structured** o Simple o Routine o Straightforward rules **Unstructured** o More complex o Requires degree of creativity **Mental obstacles to good decision making** □ Many decisions are made under pressure □ With incomplete information o Faulty Heuristics (Rule of thumb) – dangers are there o Information in hand-misleading o Stereotypes (well-known fact) o Escalation of commitment(difficult to alter once made a commitment) - Information Overload **Group decision making** □ Decisions made by the team as a whole are more likely to be accepted that those that are imposed □ Complementary skills and expertise □ Communicate freely/get ideas □ Brainstorming techniques □ Aim is to have involvement of end users? - Prototyping and participatory approaches - JAD (Joint Application Development) **Barriers to good team decisions** □ Inter-personal conflicts –team formation □ Conflicts tend to be a dampened by emergence of group norms – shared group opinions and attitudes Risky shift – people in groups are more likely to make risky decisions than they would as individuals **Approaches to Make Decision** **(a) Delphi approach** To avoid dominant personalities intruding the following approach is adopted 1. Enlist co-operation of experts 2. Moderator presents experts with problem 3. Experts send in their recommendations to the moderator 4. Recommendations are collated and circulated to all experts 5. Experts comment on ideas of others and modify their own recommendation if so moved 6. If moderator detects a consensus, stop; else back to 4 **(b) Team 'heedfulness'** □ As a football team. □ Where group members are aware of the activities of other members that contribute to overall group success □ Impression of a 'collective mind'

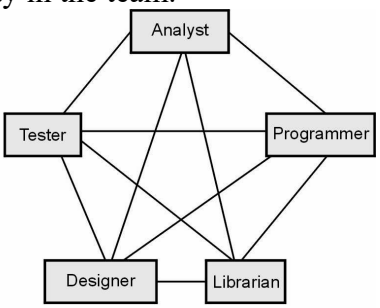
	□ Some attempts to promote this: **- Egoless programming** **- Chief programmer teams** **- XP** **- Scrum** **Egoless programming** □ Gerry Weinberg noted a tendency for programmers to be protective of their code and to resist perceived criticisms by others of the code □ Encouraged programmers to read each other's code □ Argued that software should become communal, not personal – hence 'egoless programming' **Chief programmer teams** □ Fred Brooks was concerned about the need to maintain 'design consistency' in large software systems □ Appointment of key programmers, chief programmers, with responsibilities for defining requirements, designing, writing and test software code Assisted by a support team : co-pilot – shared coding, editor who typed in new or changed code, program clerk who write and maintain documentation and tester. □ Problem – finding staff capable of the chief programmer role **Extreme programming (XP)** XP can be seen as an attempt to improve team heedfulness and reduce the length of communication paths (the time between something being recorded and it being used) □ Software code enhanced to be self-documenting □ Software regularly refactored to clarify its structure □ Test cases/expected results created before coding – acts as a supplementary specification □ Pair programming – a development of the co-pilot concept **Scrum** □ Named as an analogy to a rugby scrum – all pushing together □ Originally designed for new product development where 'time-to-market' is important □ 'Sprints' increments of typically one to four weeks □ Daily 'scrums' – daily stand-up meetings of about 15 minutes □ Unlike XP, requirements are frozen during a sprint □ At the beginning of the sprint there is a sprint planning meeting where requirements are prioritized, At end of sprint, review meeting where work is reviewed and requirements may be changed or added to.
10.	**Analyze the Team structure that addresses the issue of organization of the individual project teams.** **Software Project Teams** There are three main types of software project teams: **1. The Egoless Programming Team (the democratic team or Open structured teams)** o A "grass roots" anti-elitist style of team organization o Egoless : group owns the documents and code (not individuals) o Consists of 10 to 12 members o All decisions are based on team consensus o Depends on total cooperation of its members o Requires clear structure for the way the team interacts o Functional roles (e.g. moderator, recorder) rotate among team members o A technical leader has external responsibility and resolves issues when team doesn't reach consensus o No permanent central authority o Rarely found today, however, sometimes used in research organizations. o Provides - Higher morale and job satisfaction to the engineers - Therefore leads to less employee turnover.

o Suitable for less understood problems,
- A group of engineers can invent better solutions than a single individual.
o A manager provides administrative leadership :
- At different times different members of the group provide technical leadership.

Disadvantage :

o Team members may waste a lot time arguing about trivial points :
o Absence of any authority in the team.



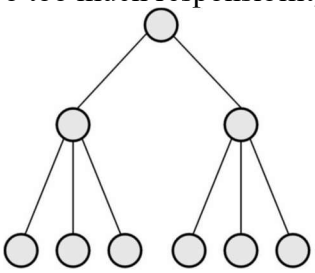


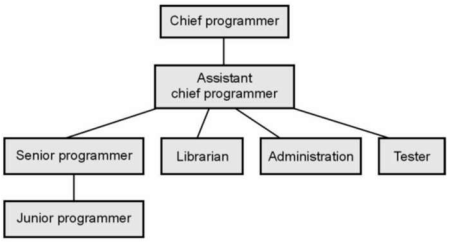
2. The Chief Programming Team

o Consists of 3 or 4 permanent team members : chief programmer, backup programmer, and librarian.
o Other programmers or analysts are assigned as needed.
o Chief programmer makes all technical and managerial decisions.
o Rarely used today, because of difficulty in recruiting and training chief programmers.
o A senior engineer provides technical leadership :
- Partitions the task among the team members.
- Verifies and integrates the products developed by the members.
o Works well when
- The task is well understood
- Also within the intellectual grasp of a single individual,
o importance of early completion outweighs other factors
- Team morale, personal development, etc.

Disadvantage

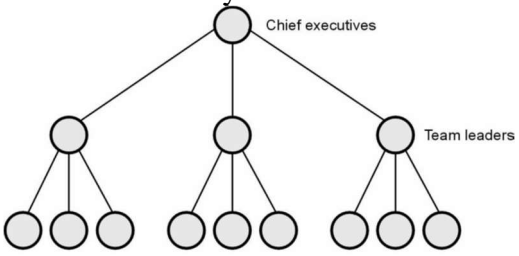
o Chief programmer team is subject to single point failure.
o too much responsibility and authority is assigned to the chief programmer.





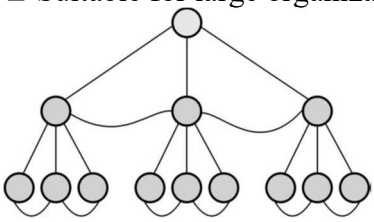
3. The Hierarchical Team (the controlled decentralized team, and project team) :

o Has a top-down flow of authority
o Project leaders manage senior engineers (senior programmers).
o Senior engineers manage junior engineers (junior programmers).
o Most commonly used team structure today.

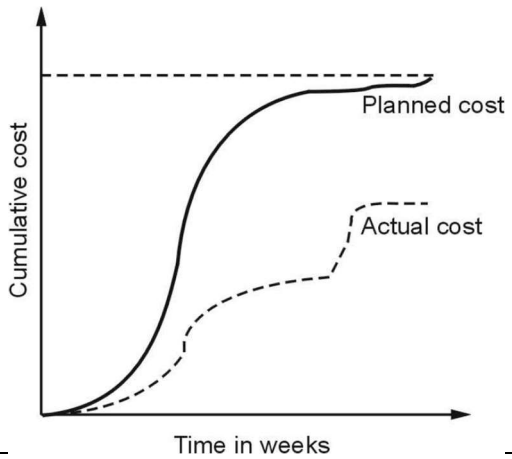


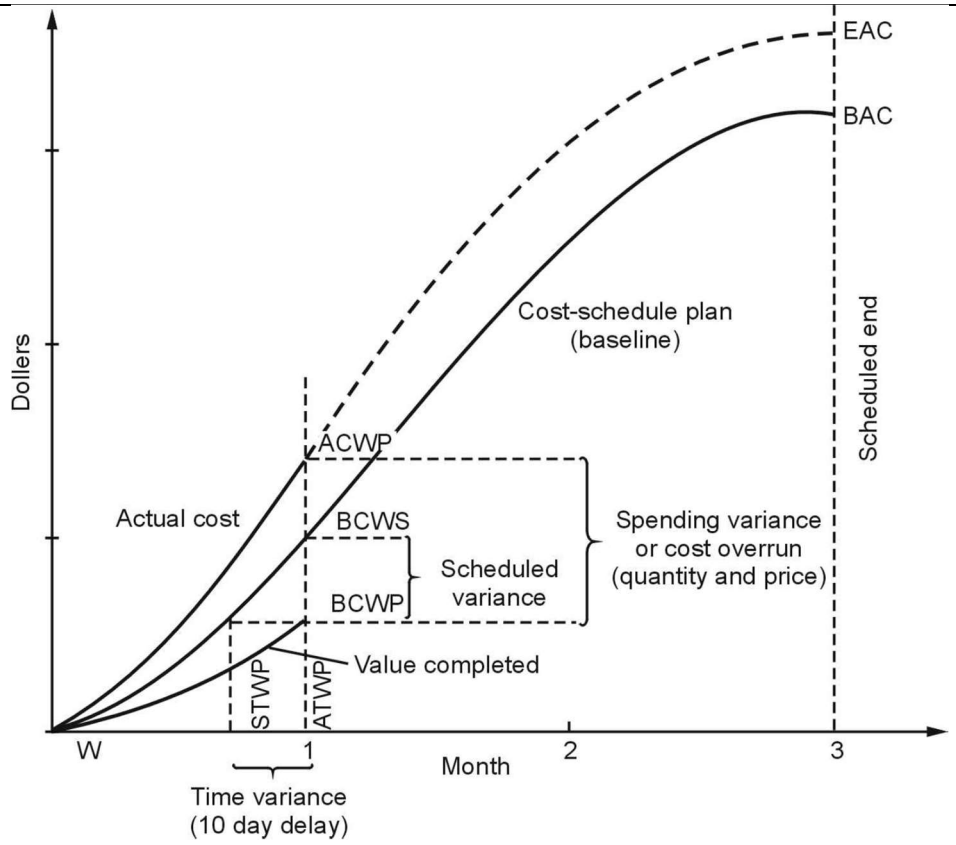
4. Mixed Control Team Organization

 Draws upon ideas from both :
o democratic organization and
o chief-programmer team organization.

	□ Communication is limited o to a small group that is most likely to benefit from it. □ Suitable for large organizations.  **Matrix Organization** □ Organize people in terms of specialties o Matrix of projects and specialist groups o People from different departments allocated to software development, possibly part-time □ **Disadvantage** o Project structure may not match organizational structure o Individuals have multiple bosses o Difficult to control a project's progress

	Real-time programming	Graphics	Databases	QA	Testing
project C	X			X	X
project B	X		X	X	X
project A		X	X	X	X

11.	**Interpret the method Earned value Analysis**. **Earned Value Analysis or Earned Value Management** □ Simply, it is a project monitoring and measurement system that : 1. Establishes a clear relationship between planned accomplishments and actual accomplishments 2. Reinforces and rewards good planning practices □ Basic concepts of Earned Value Management (EVM) 1. Each task in a project earns value as planned work is completed 2. Earned value can be compared to actual cost and budgeted cost to determine variance and predict future performance □ The budgeted cost (e.g., dollars, person-hours, person-days, etc.) in terms of your baseline plan/budget of the work performed up to a specified point in time 1. Also known as Budgeted Cost of Work Performed (BCWP) □ Each task in the Work Breakdown Structure (WBS) is assigned a BCWP based on its individual cost. 1. Project BCWP is total of BCWP for all tasks required to complete the project 



Earned Value Chart

Earned Value	EV	=	BAC	*	% Comp
Cost Variance	CV	=	EV	-	AC
Schedule Variance	SV	=	EV	-	PV
Cost Perf Index	CPI	=	EV	/	AC
Schedule Perf Index	SPI	=	EV	/	PV
Est at Completion	EAC	=	AC	/	% Comp
Est to Complete	ETC	=	EAC	-	AC
Variance at Completion	VAC	=	BAC	-	EAC
New with 4th Edition	**TCPI**	=	**BAC-EV**	/	**BAC-AC**

AC →Actual Cost, BAC →Budget at Completion, CPI →Cost Performance Index (EV/AC), CV → Cost Variance, EAC → Estimate at Completion, ETC → Estimate to complete, EV → Earned Value, PV → Planned Value, SPI → Schedule Performance Index

12. **Explain how new staff can be selected and induced into a project/do staff selection relate to quality of product? Justify with appropriate answer**

Employee selection is the process of putting right men on right job. It is a procedure of matching organizational requirements with the skills and qualifications of people. Effective selection can be done only when there is effective matching.

Factors that may Influence Selection Procedure

	The factors that may influence selection procedure are listed below. Generally, factors vary depending on the application domain, the type of project and the skills of other members of the project team. **1. Application domain experience :** For a project to develop a successful system, the developers must understand the application domain. It is essential that some members of a development team have some domain experience. **2. Platform experience :** This may be significant if low-level programming is involved. Otherwise, this is not usually a critical attribute. **3. Programming language experience :** This is normally only significant for short duration projects where there is not enough time to learn a new language. While learning a language itself is not difficult, it takes several months to become proficient in using the associated libraries and components. **4. Problem solving ability :** This is very important for software engineers who constantly have to solve technical problems. However, it is almost impossible to judge without knowing the work of the potential team member. **5. Educational background :** This may provide an indicator of what the candidate knows and his or her ability to learn. This factor becomes increasingly irrelevant as engineers gain experience across a range of projects. **6. Communication ability :** Project staff must be able to communicate orally and in writing with other engineers, managers and customers. **7. Adaptability :** Adaptability may be judged by looking at the experience that candidates have had. This is an important attribute, as it indicates an ability to learn. **8. Personality :** This is an important attribute but difficult to assess. Candidates must be reasonably compatible with other team members. No particular type of personality is more or less suited to software engineering.