5 UNIT

Project Management

PART-1

Software Configuration Management : Software Configuration Items and Tasks, Baselines, Plan for Change, Change Control, Change Request Management and Version Control.

CONCEPT OUTLINE : PART-1

- Software Configuration Management ensures that all people involved in the software process know what is being designed, developed, built, tested and delivered.
- Four major SCM activities are :
  a. Configuration identification
  b. Change control
  c. Software configuration status reporting
  d. Audits and reviews
- A configuration item is any component of an information technology infrastructure, that is, under the control of configuration management.
- The change request management is the process of requesting, determining attainability, planning, implementing and evaluation of change to a system.
- Version control combines procedure and tool that manages different versions of configuration items.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.1. What do you understand by Software Configuration Management (SCM) ? Explain the goal and objectives of SCM.

UPTU 2013-14, Marks 10

Answer

1. Software Configuration Management (SCM) is one of the foundations of software engineering. It is used to track and manage the emerging product and its versions.
2. This is to assure quality of the product during development, and operational maintenance of the product.
3. SCM ensures that all people involved in the software process know what is being designed, developed, built, tested, and delivered.

4. Through SCM, the design requirements can be traced to the final software product.
5. Thus, the Software Configuration Management (SCM) can be defined as a process of defining and implementing a standard configuration, which results into the primary benefits such as easier setup and maintenance, less down-time, better integration with enterprise management, and more efficient and reliable backups and also maximize productivity by minimizing mistakes.

Goals of software configuration management :

a. Software configuration management activities are planned.
b. Selected software work products are identified, controlled, and available.
c. Changes to identified software work products are controlled.
d. Affected groups and individuals are informed of the status and content software baselines.

Objectives of software configuration standards :

i. **Remote system administration :**
   a. The configuration standard should include necessary software and/or privileges for remote system administration tools.
   b. A remote administration client, that is, correctly installed and configured on the client side is the cornerstone of the remotely administered network.
   c. These remote tools can be used to check the version of virus protection, check machine configuration, or offer remote help-desk functionality.

ii. **Reduced user down-time :**
   a. A great advantage of using a standard configuration is that system becomes completely interchangeable resulting in reduced user down-time.
   b. If a given system experiences an unrecoverable error, an identical new system can be dropped into place.
   c. User data can be transferred if the non-functional machine is still accessible, or the most recent copy can be pulled off the backup tape with the ultimate goal being that the user experiences little change in the system interface.

iii. **Reliable data backups :**
   a. Using a standard directory for user data allows backup systems to selectively backup a small portion of a machine, greatly 'reducing the network traffic and tape usage for backup systems.
   b. Also, should a catastrophic failure occur, the data directory could be restored to a new machine with little time and effort.

   c. A divided directory structure, between system and user data, is one of the main goals of the configuration standard.

iv. **Easy workstation setup :**
   a. Any sort of standardized configuration streamlines the process of setting up the system and insures that vital components are available.
   b. If multiple machines are being setup according to a standard setup, most of the setup and configuration can be automated.

v. **Multi-user support :**
   a. Although, it is common for users to share a workstation, the system configuration is designed to allow multiple users to use the same workstation without interfering each other's work.
   b. Some software packages do not support completely independent settings for all users, however, users can have independent data areas.
   c. The directory structure implemented does not impose limits on the number of independent users a system, can have.

vi. **Remote software installation :**
   a. Most modern software packages are installed in factory pre-defined directories. While software installed in the manner will function correctly for a single user, it will lead to non-uniform configuration among a collection of machines.
   b. A good configuration standard will have software installed in specified directory areas to logically divide software on the disk.
   c. This will lead to easier identification of installed components and the possibility of automating installation procedures through the use of universal scripts.
   d. With software installed into specific directories, maintenance and upgraded running software becomes less complex.

Que 5.2. Give the format of Software Configuration Management (SCM) plan.

Answer

The procedures for software configuration management are laid down in the form of a SCM plan document. Sample structure of a configuration management plan (source IEEE standard for software configuration management plans) is given here :

i. **Introduction :**
   a. Purpose
   b. Scope
   c. Definitions and acronyms
   d. References

ii. **Management :**
   a. Organizations
   b. Configuration management responsibilities
   c. Interface control
   d. Implementation of software
   e. Configuration of software
   f. Applicable policies, directives and procedures

iii. **Configuration management activities :**
   a. Configuration identification
   b. Configuration control
   c. Configuration status accounting
   d. Audits and reviews

iv. Tools, techniques and methodologies

v. Supplier control

vi. Records collection and retention

Que 5.3. **Illustrate the various SCM activities.**

Answer

Following are the four major SCM activities :

1. **Configuration identification :**
   a. SCM process manages all the software entities and their related representations in documentation.
   b. Basically, SCM should manage all software related components that are used during development, testing, and production.
   c. The identification and structuring of the software configuration management plan is done when the software process is being defined.
   d. Software configuration items (SCIs) added to the SCM are things such as tools, design documents, requirements documents etc.
   e. The SCI that are identified, determine the baseline(s), that is, associated with the project.
   f. The number and types of baselines will depend upon the project. Baselines are the core of SCM, they provide a stable platform for work to continue from.
   g. Only authorized changes can be made to the baseline, and all changes are recorded as deltas until such time as a new baseline is generated.

2. **Change control :**
   a. Change control involves procedures and tools to bring in order to change process.
   b. Larger projects have a formal change control board (CCB), whose responsibility is to review and approve and disapprove changes.
   c. It is the CCB responsibility to provide the mechanism to maintain orderly change process.
   d. The CCB includes the following individuals :
      i. Executive sponsor
      ii. Customer representative
      iii. Marketing representative
      iv. Program manager
      v. Project manager
      vi. Quality assurance manager
      vii. Documentation leader

3. **Software configuration status reporting :**
   a. Software configuration status reporting (or accounting) is a record keeping activity of software configuration management.
   b. From the time the first SCI's were identified, all changes and the current status of changes and documents are recorded in a status accounting database.
   c. The record in the database provides management with reports as to the current state of the project.
   d. The configuration status accounting database holds the records showing the products history, how the product has evolved, and where the system is at anytime in the relation to the current baseline.
   e. Administrator uses status accounting to track and report on all SCI's formally identified and controlled. The status accounting database also maintains records to supports SCM auditing.

4. **Audit and reviews :**
   a. Audits and reviews are used to ensure that changes have been properly implemented. The formal review looks at the technical correctness of any SCI that has been modified.
   b. An SCI is looked as to determine any omissions, potential side effects, and its consistency with other SCIs.
   c. Formal reviews are conducted for all but the most trivial changes.
   d. Auditing gives us a picture of how close the current software system mirrors the software systems pictured in the baseline and the requirements documents.

e. Auditing provides the mechanism for establishing a new baseline. The final stages of an audit are used to sanction the new baseline.

f. Auditing increases the software visibility and established traceability. It helps to avoid costly re-design and refits.

Que 5.4. **Explain software configuration items, tasks and baselines.**

UPTU 2014-15, Marks 10

Answer

Configuration item :

1. A configuration item (CI) is any component of an information technology infrastructure, that is, under the control of configuration management.
2. Configuration items (CIs) can be individually managed and versioned, and they are usually treated as self-contained units for the purposes of identification and change control.
3. All configuration items (CIs) are uniquely identified by names, version numbers, and other attributes. The lowest level CI is usually the smallest unit that will be changed independently of other components.
4. CIs vary in complexity, size, and type. They can range from an entire service which may consist of hardware, software and documentation to a single program module or a minor hardware component.

Configuration tasks : Refer Q. 5.3, Page172C, Unit-5.

Configuration baselines :

1. Configuration Management (CM) plans, establish and document the requirements, standards, practices, and procedures for configuration management.
2. This includes defining baselines and establishing the labeling scheme for configuration items.
3. A baseline is a group of configuration items (products, deliverables) developed during a specific phase of the development process that has been formally accepted.
4. Once the baseline is established, changes to the items can only be done through a formal change process.
5. In other words we can say, a software baselines library is established containing the software baselines as they developed.
6. Changes to baselines and the release of software products built from the software baseline library are systematically controlled via the change control and configuration auditing functions of software configuration management.

Que 5.5. **What is change and change control process ?**

Answer

Change : Change is defined as anything hardware, software, system components, services, documents, or processes that is deliberately introduced into the production environment and which may affect a Service Level Agreement (SLA) or otherwise affect the functioning of the environment or one of its components.

Change control process :

1. Change control procedure ensures that the changes to the system are controlled and that their effect on the system can be predicted.
2. It will be appropriate if changes to software can be predicted. Change control process comes into effect when the software and associated documentation are delivered to configuration management change request form (also known as change control form or software problem report), which should record the recommendations regarding the change.
3. The recommendations may include assessment of the proposed change, the estimated costs and how the change should be implemented.
4. This form is submitted to a Change Control Authority (CCA), which decides whether or not the change is to be accepted. If change is approved by the CCA, it is applied to the software.
5. The revised software is revalidated by the Software Quality Assurance (SQA) team to ensure that the change has not adversely affected other parts of the software.
6. The changed software is handed over to the software configuration team and is incorporated in a new version of the system.

Que 5.6. **What is change management ?**

Answer

1. Change management is a structured approach to shifting/transitioning individuals, teams, and organizations from a current state to a desired future state.
2. Successful change management is more likely to occur if the following are included :
   a. Benefits management and realization to define measurable stakeholder aims, create a business case for their achievement (which should be continuously updated), and monitor assumptions, risks, dependencies, costs, return on investment, dis-benefits and cultural issues affecting the progress of the associated work.
   b. Effective communications that informs various stakeholders of the reasons for the change (why ?), the benefits of successful implementation (what is in it for us) as well as the details of the change (when ? where ? who is involved ? how much will it cost ? etc.).

c. Devise an effective education, training and/or skills upgrading scheme for the organization.
d. Counter resistance from the employees of companies and align them to overall strategic direction of the organization.
e. Provide personal counseling (if required) to alleviate any change related fears.
f. Monitoring of the implementation and fine-tuning as required.

Que 5.7. **Write a short note on change request management.**

Answer

1. A change request is considered to be addition, deletion, or modification to the scope as was defined in the contract.
2. A change request usually results in change in baselined plan.
3. A baselined plan consists of :
   a. negotiated scope,
   b. quality of work products,
   c. effort and cost for the project, and
   d. timeline for delivery.
4. When customer sends a change request, asking for additional modified features for the application, or asking in schedule changes, a change request is logged.
5. Usually, a unique number is given to this request and logged by the project manager.
6. An impact analysis is done to determine the impact of change request on configuration items.
7. A list of impacted configuration items is prepared. These configuration items include programs as well as documents that need to be modified to take care of change request. Along with these, cost effort and schedule impact are also analyzed.
8. Change request management has two important processes :
   a. negotiation with the customer after the impact analysis and
   b. executing change request process.
9. Once impact analysis has been done, customer has to be shared with information related to extra cost, schedule, and work products that need to be changed (some of these work products could have been delivered earlier).
10. These changes are given in the form of a proposal for getting customer's approval and also project management plan is reflected with these changes.

11. All these are usually authorized by the project manager as he/she is answerable to the top management for meeting project objectives (such as schedule, effort, quality, etc.).
12. The change request is recorded so that in future reference can be given for the same.
13. The steps involved in change request logging are as follows : log the change request in a standard template, carry out impact analysis, find impact on effort and schedule, get approval from customer for extra cost and modified schedule of deliveries, and then start working on the change request.
14. A change request log is maintained to keep track of each change request. The tracker uses unique change request number to track change requests.
15. Once the change request has been accepted and negotiation with customer for effort and schedule is completed, execution starts.

Que 5.8. **Explain initiate change and authorizing change request.**

Answer

Initiate change :

1. The objective of initiate change is to formally initiate a change through the submission of a Change Request (CR).
2. Change requests may be initiated :
   a. Before a project is completed *i.e.*, a project change request made after a project has started but before it has completed and transitioned to production.
   b. After a project has moved to production *i.e.*, a production change request.
   c. By any member of the project team or any other stakeholder with a vested interest in the project.
3. The change request management process ensures that change requests are created with consistent quality and completeness and discards irrelevant requests.
4. Although a project change request can be requested by any stakeholder, it is typically initiated and submitted by one of the project team members who submit requests for changes relating to its area of responsibility.
5. For example, the data movement team may initiate database change requests required to satisfy new or changed data sources.

Authorizing change request :

1. The objective of authorize change requests is to establish a formal process for approving change.

2. A Change Management Team (CMT) should be established and responsible for reviewing change requests and voting on the changes according to predefined voting logic.
3. The following change request management process is required to authorize a change request :
   a. **Authorize standard change request :** A standard changes do not require CMT approval since they are automatically approved and move directly to change deployment.
   b. **Authorize minor change request :** The change manager is responsible for determining if the change is minor and, if so, may approve it or submit it to the CMT for review/approval.
   c. **Authorize significant or major change request :** The change manager is responsible for submitting all significant and/or major changes to the CMT. For approval, the CMT shall vote to :
      i. Accept the change.
      ii. Reject the change.
      iii. Escalate the change to senior management.
   d. **Log authorization :** The change manager shall update change log with the CMT decision (status) and status date.

Que 5.9. **What do you mean by version control and explain the activities participating in version control ?**

Answer

1. Version control combines procedure and a tool that manages different versions of configuration items and that are created during the software engineering process.
2. A version control tool is the first stage towards being able to manage multiple versions. Once it is in place, a detailed record of every version of the software must be kept.
3. This comprises the name of each source code component, including the variations and revisions. Each version of software is a collection of software configuration items (source code, documentation and data) and each version may be composed of different variants.
4. Version control activity is split into mainly four sub-activities :
   a. **Identifying new version :**
      i. A software configuration items will receives a new version number when there has been a change to its established baseline.
      ii. Each previous version will be stored in a corresponding directory such as version0, version1, version2 etc.

   b. **Numbering scheme :**
      i. The numbering scheme will have the following format : Version X.Y.Z
      ii. The first letter (X) represents the entire SCI. Therefore changes made to the entire configuration items or changes large enough to warranty a completely new release of items will causes the first digit to increase.
      iii. The second letter (Y) represents a component of SCI. The digit will sequentially increase if a change is made to a component, or small changes to multiple components.
      iv. The third letter (Z) represents a section of components of SCI. This number will only be possible if component of an SCI can be broken down into individual sections.
   c. **Visibility :**
      i. The version number will be visible either in a frame or below the title.
      ii. The decision for this depends upon the group decision to code all the documents for a frame capable browser or allow for non frame capable browser.
      iii. In either case, number will always be made available.
   d. **Tracking :** The best way to keep track of the different versions is with a version evolution graph which shown in Fig. 5.9.1.

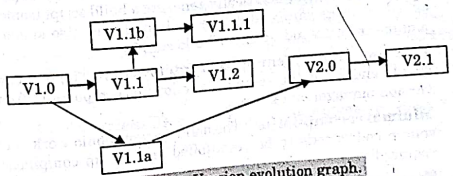

Fig. 5.9.1. Version evolution graph.

Que 5.10. **Describe two ways in which system building tools can optimize the process of building a version of a system from its components.** **UPTU 2012-13, Marks 10**

Answer

1. System building is a process of assembling program components, data and libraries and then compiling and linking these to create an executable system.

2. System building tools and version management tools must communicate as the build process involves checking out component versions from repository managed by the version management system.
3. The configuration description used to identify a baseline is also used by the system building tool.
4. System building is a complex process, which is potentially error-prone, as there may be different platforms involved such as :
   a. The development system, which includes development tools such as compilers, source code editors etc. Developers check out code from the version management system into a private workspace before making changes to the system. They may wish to build a version of a system for testing in their development environment before committing changes that they have made to the version management system.
   b. The build server, which is used to build definitive, executable versions of the system. This interacts closely with the version management system. Developers check in code to the version management system before it is built. The system built may rely on external libraries.
5. There are many build tools available and build system may provide following features :
   a. **Build script generation :** If necessary, the build system should analyze the program that is being built, identify dependent components, and automatically generate a build script (sometimes called a configuration file). The system should also support the manual creation and editing of build scripts.
   b. **Version management system integration :** The build system should check out the required versions of components from the version management system.
   c. **Minimal recompilation :** The build system should work out what source code needs to be recompiled and set up compilations if required.
   d. **Executable system creation :** The build system should link the compiled object code files with each other and with other required files, such as libraries and configuration files, to create an executable system.
   e. **Test automation :** Some build systems can automatically run automated tests using test automation tools such as JUnit. These check that the build has not been 'broken' by changes.
   f. **Reporting :** The build system should provide reports about the success or failure of the build and the test that have been run.
   g. **Documentation generation :** The build system may be able to generate release notes about the build and system help pages.

PART-2

Risk Management : Risks and Risk Types, Risk Breakdown Structure (RBS), Risk Management Process : Risk Identification, Risk Analysis, Risk Planning, Risk Monitoring and Cost Benefits Analysis.

CONCEPT OUTLINE : PART-2

- Software risk is a problem that could cause some loss or threaten the success of a software project, but which has not happened yet.
- Types of risks :
  a. Estimation errors
  b. Planning assumptions
  c. Eventualities
- Risk management involves basically two steps :
  a. Risk assessment
  b. Risk control
- Risk planning is to identify strategies to deal with risk.
- Risk monitoring is the continually reassessing of risk as the project proceeds and conditions change.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.11. Define risk. What are various categories of risks ?

Answer

1. Software risk is a problem that could cause some loss or threaten the success of a software project, but which has not happened yet.
2. These potential problems might have an adverse impact on the cost, schedule, or technical success of the software project, the quality of software products, or project team morale.
3. Software developers are extremely optimists. When planning software projects, we often assume that everything will go exactly as planned.
4. Alternatively, we take the other extreme position. The creative nature of software development means we can never accurately predict what is going to happen, so what is the point of making detailed plans ?

5. Both these perspectives can lead to software surprises, when unexpected things happen that throw the project completely out of track. Software surprises are never good news.
6. Risk management is becoming recognized as an important area in the software industry to reduce this surprise factor.
7. Risk management means dealing with a concern before it becomes crisis. Therefore, most of the software development activities include risk management as a key part of the planning process and expect the plan to highlight the specific risk areas.
8. The object planning is expected to quantify both probability of failure and consequences of failure and to describe what will be done to reduce the risk.

Risks can be categorized as follows :

a. **Project risks :** Risks that threaten the project (or the project schedule).
b. **Product risks :** Risks that threaten the quality of the software developed.
c. **Business risks :** Risks that threaten the development (or client) organization.

Que 5.12. List and discuss some of the points specific for identifying the risks during software development. Also, give some of the category of risks that are to be identified.

UPTU 2014-15, Marks 10

Answer

For the purpose of identifying and managing those risks that may cause a project to overrun its time scale or budget, it is convenient to identify three types of risk :

1. Those caused by the inherent difficulties of estimation.
2. Those due to assumptions made during the planning process.
3. Those of unforeseen (or at least unplanned) events or hazards occurring.

1. **Estimation errors :**
   a. Some tasks are harder to estimate than others because of the lack of experience of similar tasks or because of the inherent nature of the task.
   b. Producing a set of user manuals is reasonably straightforward and, given that we have carried out similar tasks previously, we should be able to estimate with some degree of accuracy how long the task will take and how much it will cost.
   c. On the other hand, the time required for program testing and debugging might be difficult to predict with a similar degree of accuracy, even if we have written similar programs in the past.

   d. Estimation can be improved by analyzing historic data for similar activities and similar systems. Keeping records comparing our original estimates with the final outcome will reveal the type of tasks that are difficult to estimate correctly.

2. **Planning assumptions :**
   a. At every stage during planning, assumptions are made which (if not valid) may put the plan at risk.
   b. Our activity network is likely to be built on the assumption of using a particular design methodology which may be subsequently changed.
   c. We generally assume that, following coding, a module will be tested and then integrated with others.
   d. We might not plan for module testing showing up the need for changes in the original design but, in the event, it might happen.
   e. At each stage in the planning process, it is important to list explicitly all of the assumptions that have been made and identify what effects they might have on the plan if they are inappropriate.

3. **Eventualities :**
   a. Some eventualities might never be foreseen and we can only resign ourselves to the fact that unimaginable things do, sometimes, happen. They are, however, very rare.
   b. The majority of unexpected events can, in fact, be identified.
   c. The requirements specification might be altered after some of the modules have been coded, the senior programmer might take maternity leave, the required hardware might not be delivered on time.
   d. Such events do happen from time to time and, although the likelihood of anyone of them happening during a particular project may be relatively low, they must be considered and planned for.

Que 5.13. Write a short note on Risk Breakdown Structure (RBS).

Answer

1. When planning a project to meet targets for cost, schedule, or quality, it is useful to identify likely risks to the success of the project. A risk is any possible situation that is not planned for, but that, if it occurs, is likely to divert the project from its planned result.
2. For example, an established project team plans for the work to be done by its staff, but there is the risk that an employee may unexpectedly leave the team.
3. In project management, the risk management process has the objectives of identifying, assessing, and managing risks, both positive and negative.

4. All too often, project manager's focus only on negative risk, however, good things can happen in a project, "things" that were foreseen, but not expressly planned.
5. The objective of risk management is to predict risks, assess their likelihood and impact, and to actively plan what should be done ahead of time to best deal with situations when they occur.
6. The risk management process usually occurs in five distinct steps: plan risk management, risk identification, qualitative and quantitative risk analysis, risk response planning, and risk monitoring and control.
7. The central point of risk identification and assessment in risk management is understanding the risk. However, this is also where project managers and risk subject matter experts (SMEs) get the least help from recognized references, best practices, or work standards.
8. Currently, the Project Management Institute (PMIr) has a team of SMEs working on a Practice Standard for Risk Management. This team has identified one very good tool: theRisk Breakdown Structure (RBS).
9. The RBS helps the project manager, the risk manager, and almost any stakeholder to understand, and therefore be able to identify and assess risk.
10. The RBS will prove extremely valuable to better grasp when a project needs to receive special scrutiny, in other words, when risk might happen.
11. The RBS can also help the project manager and the risk manager to better understand recurring risks and concentrations of risk that could lead to issues that affect the status of the project.
12. Following the concept of the Work Breakdown Structure (WBS), the Risk Breakdown Structure provides a means for the project manager and risk manager to structure the risks being addressed or tracked.
13. Just as PMI defines the Work Breakdown Structure as a "deliverable-oriented grouping of project elements that organizes and defines the total work scope of the project", the RBS could be considered as a "hierarchically organized depiction of the identified project risks arranged by risk category."
14. In project management language, risks include anything unplanned and unforeseen that can have a negative impact on the project's costs, timing or quality. A good project manager should be able to manage the risks effectively and get the project on track.
15. Many project managers and risk managers currently use "home-grown" methods for listing, identifying, assessing, and tracking risks in their projects.
16. These methods include:spreadsheets, listing, generic risk taxonomy, based somewhat loosely on various standards and guidelines.

17. An approach that simply places the risks in a list, a simple table, or even in a database does not provide the strength of using a structured, organized method similar to a Work Breakdown Structure.
18. To fully understand the risks and better identify and assess the risk, a "deep-dive" into each risk, recording as many levels of identification as necessary, may be required.

Que 5.14. **Write a short note on risk management process.**

UPTU 2013-14, Marks 10

OR

What do you mean by risk ? Explain. How risk planning and monitoring is carried out ? Discuss. **UPTU 2012-13, Marks 10**

OR

Discuss risk monitoring. **UPTU 2014-15, Marks 05**

OR

Write a short note on risk analysis.

Answer

Risk management is a very tedious task. It involves basically two steps :

1. Risk assessment
2. Risk control

1. **Risk assessment :** It is the process of examining a project and identifying areas of potential risk. The risk assessment consists of three activities :
   a. Risks Identifying.
   b. Risk Analyzing.
   c. Risk Prioritization.

   a. Risk identification :
   i. Risk identification is a systematic attempt to specify threats to the project plan. The purpose of risk identification is to develop a list of risk items called risk statement.
   ii. Risk identification can be facilitated with the help of a checklist of common risk areas for software projects or by examining the contents of an organizational database of previously identified risks and mitigation strategies (both successful and unsuccessful).
   iii. Risk identification is carried out as a team process using brainstorming. To assist the process a list of risk types can be used.
   iv. The end product of this step of the process is a list of risks that could occur and affect the product, the process or the business.

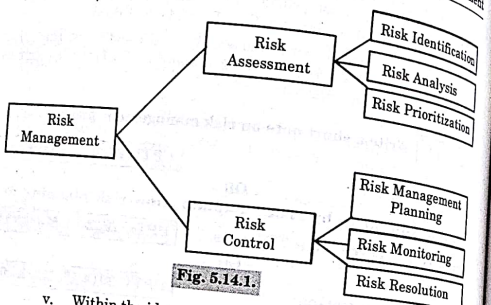

Fig. 5.14.1.

v. Within the identification phase, several activities occur. The main activities are :

1. **Identify risks :** There are many techniques to be used to identify risk. Some of these are checklists, interviews, brainstorm meetings, reviews and surveys. A checklist to be used as a tool for identification of risks is provided.
2. **Define risk attributes :** After the risks are identified, they are evaluated with the criteria : likelihood of occurrence (probability), consequence and time frame for action. These values are initial estimations which are analyzed more in the next phase.
3. **Document :** The risks are then documented. Together with the name of the risks, a risk statement and context are to be specified. In this initial phase, the description of the risk issue, the probability and the consequence are specified in subjective terms.
4. **Communicate :**
   a. Spreading the knowledge to the project members. Inputs to the identification phase are uncertainties, knowledge, concerns and issues.
   b. If projects have been conducted before, the resolution of these inputs may be stored in a database that helps the project managers to detect and find appropriate risk items.
   c. The output of the identification phase is the risk statement that contains identified risks that may affect the project.

   d. Furthermore, together with the statements, risk context is produced.
   e. The purpose of the context is to describe the risk items events, conditions, constraints, assumptions, circumstances, contributing factors and related issues by answering the question what, when, where, how and why of each risk identified.
   f. One risk may have several answers to each of these questions.
   g. The risk identification should include the project team and the risk management team. In the second iteration, the entire project team and primary stakeholders should participate.
   h. If an unbiased analysis is wanted, a person who is not involved in the project may perform a third iteration of the risk identification.

b. Risk analysis :

i. When the risks have been identified, all items are analyzed using different criteria. The purpose of the risk analysis is to assess the loss probability and magnitude of each risk item.

ii. The input is the risk statement and context developed in the identification phase.

iii. The output of this phase is a risk list containing relative ranking of the risks and a further analysis of the description, probability, consequence and context.

iv. The main activities in this phase are :

1. **Group similar risks :** Detect duplicates and find new risk items by grouping the identified risks into categories.
2. **Determine risk drivers :** The risk drivers are parameters that affect the identified risk. For example, schedule drivers are included in the critical path model. Determining these properties help to assess and prioritize the risks.
3. **Determine source of risks :** The sources of risks are the root causes of the risks. These are determined by asking the question why ? and trying to figure out what may have caused the risk. Several root causes may lead to the same risk.
4. **Estimate risk exposure :** The risk exposure is measure of the probability and the consequence of a risk item. The consequence can also be stated in terms of loss (for example, life, money, property, reputation).

5. **Evaluate against criteria :**
   a. Each risk item is evaluated using the predefined criteria, which are important for the specific project.
   b. Criteria may be stated in terms of the probability of occurrence, the consequence and the time frame.
   c. This information is used to prioritize the risks. Once this is done, risks can be prioritized and the most serious risks can be identified for monitoring.

c. **Risk prioritization :**
   i. Risk prioritization helps the project focus on its most severe risks by assessing the risk exposure.
   ii. Exposure is the product of the probability of incurring a loss due to the risk and the potential magnitude of that loss.
   iii. This prioritization can be done in a quantitative way, by estimating the probability (0.1–1.0) and relative loss, on a scale of 1 to 10.
   iv. Multiplying these factors together provide an estimation of risk exposure due to each risk item, which can run from 0.1 to 10.
   v. The higher the exposure, the more aggressively the risk should be tackled. It may be easier to simply estimate both probability and impact as High, Medium or Low.
   vi. Those items having at least one dimension rated as high are the ones to worry about first.

2. **Risk control :** Risk control is the process of managing risks to achieve the desired outcomes. Risk control process involves the following activities :
   a. Risk planning
   b. Risk mitigation
   c. Risk resolution
   d. Risk monitoring

a. **Risk planning :**
   i. Risk management planning produces a plan for dealing with each significant risk, including mitigation approaches, owners, and time lines.
   ii. Risk resolution is execution of the plans for dealing with each risk. Finally, risk monitoring involves tracking your progress toward resolving each risk item.
   iii. Risk planning is to identify strategies to deal with risk. These strategies fall into three categories :
      1. Risk avoidance
      2. Risk minimization
      3. Risk contingency plans.

b. **Risk mitigation :**
   i. The risk mitigation is plan that would reduce or eliminate the highest risks. The key question is : what should be done and who is responsible to eliminate or minimize the risk ?
   ii. The mitigation plan includes a description of the actions that can be taken to mitigate the red rated risk and assigns a primary handler for the action.

c. **Risk resolution :**
   i. When a risk has occurred, it has to be solved. Risk resolution is the execution of the plans for dealing with each risk.
   ii. 'If the risk is at the watch list, a plan of how to resolve the risk already had taken place. The project manager has to respond to the trigger and execute the action plan.
   iii. The project manager also needs to report progress against the plan and correct for deviation.
   iv. The input to this phase is the lists of risk and its implementation. The output is risk action plan.

d. **Risk monitoring :**
   i. Risk monitoring is the continually reassessing of risks as the project proceeds and conditions change.
   ii. For example, successful completion of beta testing means that the risk of the client organization rejecting the system is minimal, while large turnover in development staff usually increases project and product risks.

Que 5.15. What are the factors which affects the risk identification procedure of any software project ?

Answer

The categories of factors that will need to be considered include the following :

1. **Application factors :**
   a. The nature of the application, whether it is a simple data processing application, a safety-critical system or a large distributed system with real-time elements is likely to be a critical factor.
   b. The expected size of the application is also important because the larger the system, the greater is the likelihood of errors and communication and management problems.

2. **Staff factors :**
   a. The experience and skills of the staff involved are clearly major factors an experienced programmer is, one would hope, less likely to make errors than one with little experience.

b. We must, however, also consider the appropriateness of the experience - experience in coding small data processing modules in COBOL may be little value if we are developing a complex real-time control system using C++.

c. Such factors as the level of staff satisfaction and the staff turn-over rates are also important to the success of any project - demotivated staff or key personnel leaving unexpectedly have caused many a project to fail.

3. **Project factors :**

a. It is important that the project and its objectives are well defined and that they are absolutely clear to all members of the project team and all key stakeholders.

b. Any possibility that this is not the case will pose a risk to the success of the project.

c. Similarly, an agreed and formal quality plan must be in place and adhered to by all participants and possibility that quality plan is inadequate or not adhered to will jeopardize the project.

4. **Project methods :**

a. Using well-specified and structured methods (such a PRINCE 2 and SSADM) for project management and system development will decrease the risk of delivering a system that is unsatisfactory or late.

b. Using such methods for the first time, though, may cause problems and delays it is only with experience that the benefits accrue.

5. **Hardware/Software factors :**

a. A project that requires new hardware for development is likely to pose a higher risk than one where the software can be developed on existing (and familiar) hardware.

b. Where a system is developed on type of hardware or software platform to be used on another there might be additional (and high) risks at installation.

6. **Changeover factors :** The need for an 'all-in-one' changeover to the new system poses particular risks. Incremental or gradual changeover minimizes the risks involved but is not always practical. Parallel running can provide a safety net but might be impossible or too costly.

7. **Supplier factors :**

a. The extent to which a project relies on external organizations that cannot be directly controlled often influences the project's success.

b. For example, delays in the installation of telephone lines or delivery of equipment may be difficult to avoid, particularly if the project is of little consequence to the external supplier.

8. **Environment factors :** Changes in the environment can affect a project's success. For example, a significant change in the taxation regulations could have serious consequences for the development of a payroll application.

9. **Health and safety factors :** While not generally a major issue for software projects, the possible effects of project activities on the health and safety of the participants and the environment should be considered.

Que 5.16. How to prioritize the monitoring ? Explain.

Answer

So far we have assumed that all aspects of a project will receive equal treatment in terms of the degree of monitoring applied. We must not forget, however, that monitoring takes time and uses resources that might sometimes be put to better use.

Following are the priorities applied in deciding levels of monitoring :

1. **Critical path activities :**

a. Any delay in an activity in the critical path will cause a delay in the completion date for the project.

b. Critical path activities are therefore likely to have a very high priority for close monitoring.

2. **Activities with no free float :**

a. A delay in any activity with no free float will delay at least some subsequent activities. It might not delay the project completion date.

b. These subsequent delays can have serious effects on our resource schedule as a delay in a subsequent activity could mean that the resources for that activity will become unavailable before that activity is completed because they are committed elsewhere.

3. **Activities with less than a specified float :**

a. If any activity has very little float it might use up this float before the regular activity monitoring brings the problem to the project manager's attention.

b. It is common practice to monitor closely those activities with less than, say one week free float.

4. **High risk activities :**

a. A set of high risk activities should have been identified as part of the initial risk profiling exercise.

b. If we are using the PERT three-estimate approach we will designate as high risk those activities that have a high estimated duration variance.

c. These activities will be given close attention because they are most likely to overrun or overspend.

5. **Activities using critical resources :**
   a. Activities can be critical because they are very expensive (as in the case of specialized contract programmers).
   b. Staff or other resources might be available only for a limited period, especially if they are controlled outside the project team.
   c. In any event, an activity that demands a critical resource requires a high level of monitoring.

Que 5.17. Explain cost benefit analysis.

UPTU 2014-15, Marks 05

Answer

1. The most common way of carrying out an economic assessment of a proposed information system or other development, is by comparing the expected costs of development and operation of the system with the benefits of having it in place.
2. Assessment focuses on whether the estimated costs are exceeded by the estimated income and benefits.
3. Additionally, it is usually necessary to ask whether or not the project under consideration is the best of a number of options.
4. There might be more candidate projects that can be undertaken at any one time and, in any case, projects will need to be prioritized so that resources are allocated effectively.
5. The standard way of evaluating the economic benefits of any project is to carry out a cost benefit analysis, which consists of two steps :
   a. **Identifying and estimating all of the costs and benefits of carrying out the project and operating the delivered application :**
      i. These include the development costs of the system, the operating costs and the benefits that are expected to accrue from the operation of the new system.
      ii. Where the proposed system is replacing an existing one, these estimates should reflect the change in costs and benefits due to the new system.
      iii. For example, a new sales order processing system could not claim to benefit an organization by the total value of sales, only by the increase due to the use of the new system.
   b. **Expressing these costs and benefits in common units :**
      i. We need to evaluate the net benefit, that is, the difference between the total benefit accruing from the system and the total cost of creating and operating it.

ii. To do this, we must express each cost and each benefit in some common unit. The fundamental common unit of measurement is money.

iii. We therefore need to express each of the expected benefits and costs in monetary terms.

6. Most direct costs are relatively easy to identify and quantify in approximate monetary terms. It is helpful to categorize costs according to where they originate in the life of the project.
   a. **Development costs :** Include the salaries and other employment costs of the staff involved in the development project and all associated costs.
   b. **Setup cost :** Include the costs of putting the system into place. These consist mainly of the costs of any new hardware and ancillary equipment, but will also include costs of file conversion, recruitment and staff training.
   c. **Operational costs :** Consist of the costs of operating the system once it has been installed.
7. Benefits, on the other hand, are often quite difficult to quantify in monetary terms even once they have been identified. Benefits may be categorized as follows :
   a. **Direct benefits :** These accrue directly from the operation of the proposed system. These could, for example, include the reduction in salary bills through the introduction of a new, computerized system.
   b. **Assessable indirect benefits :** These are generally secondary benefits, such as increased accuracy through the introduction of a more user-friendly screen design where we might be able to estimate the reduction in errors, and hence costs, of the proposed system.
   c. **Intangible benefits :** These are generally longer term or benefits that are considered very difficult to quantify. Enhanced job interest can lead to reduced staff turnover and, hence, lower recruitment costs.

Que 5.18. What are the cost benefit evaluation techniques ? Explain each.

OR

Write short notes on :

1. **Net profit**
2. **Payback period**
3. **Return on investment**
4. **Net present value**

Answer

We would consider proceeding with a project only when the benefits outweigh the costs. However, in order to choose among projects, we need to take into account the timing of the costs and benefits as well as the benefits relative to the size of the investment.

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. **Payback period :**
   a. The payback period is the time taken to break even or payback the initial investment.
   b. Normally, the project with the shortest payback period will be chosen on the basis that an organization will wish to minimize the time that a project is 'in debt'.
   c. The advantage of the payback period is that it is simple to calculate and is not particularly sensitive to small forecasting errors.
   d. Its disadvantage as a selection technique is that it ignores the overall profitability of the project - in fact, it totally ignores any income (or expenditure) once the project has broken even.
3. **Return on investment :**
   a. 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.
   b. There are some variations on the formula used to calculate the return on investment, but a straightforward common version is :

$$ROI = \frac{\text{average annual profit}}{\text{total investment}} \times 100$$

4. **Net present value :**
   a. The calculation of net present value (NPV) is a project evaluation technique that takes into account the profitability of a project and the timing of the cash flows that are produced.
   b. It does so by discounting future cash flows by a percentage known as the discount rate.
   c. The present value of any future cash flow may be obtained by applying the following formula :

$$\text{Present value} = \frac{\text{Value in year } t}{(1+r)^t}$$

where r is the discount rate, and t is the number of years into the future that the cash flow occurs.

Que 5.19. **What do you mean by cash flow forecasting ? Explain.**

Answer

1. As important as estimating the overall costs and benefits of a project is the forecasting of the cash flows that will take place and their timing.
2. A cash flow forecast will indicate when expenditure and income will take place.
3. We need to spend money, such as staff wages, during the development stages of a project.
4. Such expenditure cannot be deferred until income is received (either from using the software if it is being developed for in-house or from selling it).
5. It is important that we know that we can fund the development expenditure either from the company's own resources or by borrowing from the bank.
6. In any event, it is vital to have some forecast of when expenditure such as the payment of salaries and bank interest will take place and when any income is to be expected, such as payment on completion or, possibly, stage payments.

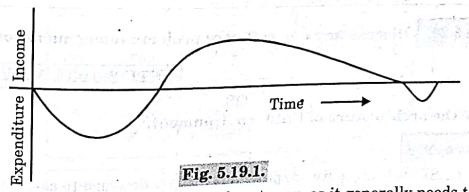

Fig. 5.19.1.

7. Accurate cash flow forecasting is not easy, as it generally needs to be done early in the project's life cycle (at least before any significant expenditure is committed) and many items to be estimated (particularly the benefits of using software or decommissioning costs) might be some years in the future.
8. When estimating future cash flows, it is usual to ignore the effects of inflation.
9. Trying to forecast the effects of inflation increases the uncertainty of the forecasts.
10. Moreover, if expenditure is increased due to inflation it is likely that income will increase proportionately.
11. However, measures to deal with increases in costs where work is being done for an external customer must be in place, for example, index, linked prices where work involves use of raw materials.

PART-3

Software Project Management Tools : CASE Tools, Planning and Scheduling Tools, MS-Project.

CONCEPT OUTLINE : PART-3

- CASE tools are software programs that are designed to assist human programmers with the complexity of the processes and the artifacts of software engineering.
- Microsoft Project helps you achieve your project goal on time and on budget.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.20. **Discuss any one software project management tool.**

UPTU 2012-13, Marks 10

OR

Give the architecture of CASE environment.

Answer

1. CASE tools are software programs that are designed to assist human programmers with the complexity of the processes and the artifacts of software engineering.
2. They constitute the laws of and the automated tools that aid in the synthesis, analysis, modeling, or documentation of software.
3. CASE stands for a large number of applications reaching from simple editing tools to environments supporting the whole life cycle.
4. CASE attacks software productivity problems at both ends of the life cycle by automating many analysis and design tasks, as well as program implementation and maintenance tasks.
5. The major task of a CASE tool is to accept different kinds of specifications, analyze the specifications, transform specifications and maintain a large, ever-growing set of interrelated specifications possibly in several versions.

CASE architecture :

1. Fig. 5.20.1 shows a general architecture of a CASE environment. Central to this architecture is repository, the most critical and important feature of a CASE environment.

2. The complete information about meta-models, methods, project artifacts, baseline, process models etc., is available in the repository.
3. This information can be shared by different team members efficiently thereby improving the project management as a whole.
4. While implementing, the repository can be either centralized or distributed. Normally this is implemented using a database or a file system.

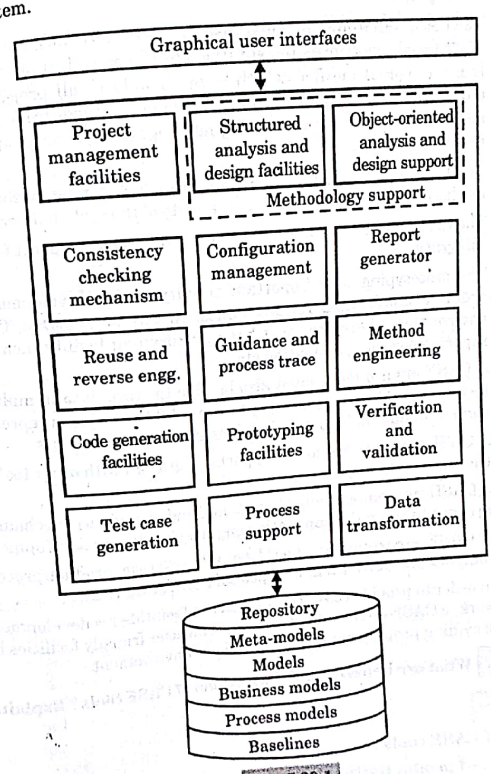

Fig. 5.20.1.

5. As repository holds different types of data from different phases of software life cycle, it must make the data available to all tools that require them.

6. Some of the important properties of a CASE environment are discussed below :
   a. A CASE environment must have facility to update and retrieve information at any point of development.
   b. It must support project management and planning facilities in order to ensure timely completion of a project and to track development in the project.
   c. The CASE environment must also have support for methodology (structured, object-oriented etc.) used by the organization. As there is no universal method which is applicable to all projects, the environment should support building of new methods from existing method components (method engineering) using concept of meta-model.
   d. The CASE environment must have proper consistency checking mechanisms in order to maintain integrity of the system description.
   e. The CASE environment must also have necessary support for tool integration.
   f. The prototyping is an important activity during development in order to understand requirements. As far as possible, CASE environment therefore must have prototyping facility including report generation, animation etc.
   g. A CASE environment must also be able to store data in multiple forms *i.e.*, tables, diagrams, matrix etc. All these forms represent same semantic information but from different viewpoints.
   h. A CASE environment must support integration with other tools in the organization also.
   i. A CASE environment must have some kind of guidance mechanism in order to guide the application engineering during development.
   j. A CASE environment should be able to trace product/process components, store them in repository for future reuse.
   k. In order to provide easy access to user to facilitate his development work, a CASE environment must provide user friendly facilities by providing proper graphics and windows environment.

Que 5.21. **What are benefits and limitations of CASE tools ? Explain.**

Answer

Benefits of CASE tools :

1. **Improved productivity :** CASE tools provide automation and reduce the time to complete many tasks, especially those involving diagramming and associated specifications. Estimates of improvements in productivity after application range from 355 to more than 200%.

2. **Better documentation :** By using CASE tools, vast amount of documentation are produce along the way. Most tools have revisions for comments and notes on systems development and maintenance.
3. **Reduced lifetime maintenance :**
   a. As a result of better design, better analysis and automatic code generation, automatic testing and debugging, overall systems quality improves.
   b. CASE provide re-engineering tools that can be very important because they make this process more efficient, less time consuming and less expensive by discovering older parts of the system that can be reused.
4. **Improved accuracy :**
   a. CASE tools can provide ongoing debugging and error checking which is very vital for early defect removal, which actually played a major role in shaping modern software.
   b. The importance of early defect removal is very important. Less effort and time are consumed if corrections are made at an early stage, such as the design stage.
   c. As the system grows larger, it becomes difficult to modify. Error identification becomes harder.
5. **Opportunity to non-programmers :**
   a. With the increased movement toward object-oriented technology and client-server based, programming can also be done by people who do not have a complete programming background.
   b. By using the lower CASE tools, it could be possible to develop software from the initial design and analysis phase.
6. **Intangible benefits :** CASE tools can be used to allow for greater user participation, which can lead to better acceptance of the new system. This can reduce the initial learning curve.

Limitations of CASE tools :

1. **Cost :**
   a. Using CASE tools is a very costly affair. In fact, most firms engaged in software development on a small scale do not invest in CASE tools because they think that the benefits of CASE are justifiable only in the development of large systems.
   b. The cost of outfitting every systems developer with a preferred CASE tool kit can be quite high.
   c. Hardware and systems software, training and consulting are all factors in the total cost equation of using CASE tools.
2. **Learning curve :**
   a. In most cases, programmer productivity may fall in the initial phase of implementation, because users need time to learn the technology.

b. In fact, a CASE consulting industry has evolved to support uses of CASE tools.

c. The consultants offer training and on-site services that can be crucial to accelerate the learning curve and to the development and use of the tools.

3. **Tool mix :**

a. It is important to make an appropriate selection of tool mix to get cost advantage. CASE integration and data integration across all platforms is also very important.

b. The ability to share the results of work done on one CASE tool with another CASE tool is perhaps the most important type of CASE integration.

Que 5.22. **Give some examples of CASE tools which support the different stage of software life cycle.**

OR

List out three most used CASE tools.

UPTU 2012-13, Marks 05

Answer

Some popular CASE tools supporting different stages of software life cycle are :

1. **Software requirements tools :** A number of tools are proposed for modeling, tracing, and analyzing requirements. CASE diagramming or upper CASE tools represent the system requirements visually using structured or object-oriented methodologies. Examples are :
   a. Turbo-analyst
   b. Oracle's Designer/2000, Argo/UML Rational ROSE
   c. MS Access
   d. Dia
   e. Excelerator, Stalemate
2. **Software design tools :** These tools are used to support the system design stage of SDLC and in most of the cases are extensions of CASE tools used to requirements analysis stage. They can be used to support design, verification and optimization.
3. **Software construction tools :** Software construction tools are the tools which are used to code and implement the software and hence transform the software requirements into working product. These tools are required even after the product is installed so as to remove defects and maintain it. Broadly they can be classified as :
   a. Program editors
   b. Compilers

c. Interpreters
d. Debuggers

4. **Software testing tools :** These are the automated tools that support various activities of a software testing stage of SDLC. They can be classified as :
   a. **Test generators :** These are tools which assist the developers in test case design and their documentation. Example, MS Access, Quick List.
   b. **Test execution and evaluation tools :** These tools support the process of test case execution and also evaluate the results of execution. The various tools under this category are ;
      i. Capture/Playback tools to automate execution of test cases. Example SQA Robot, QA Playback, Java-Star, Ferret, Auto-Tester, Web etc.
      ii. Coverage analysis tools to ensure all parts of code i.e. statement, decision, conditions, paths, loops etc. are executed. Examples are Visual Testing, Java-Scope, Aqua-Prova etc.
      iii. Memory testing tools to test memory related problems example, Bounds-Checker, Heap-Agent etc.
      iv. Simulators replace the actual hardware/software which interacts with the software to be tested and also evaluate system. Examples are Performance Studio, Load-Runner etc.
   c. **Test management tools :** These tools support management of different test artifacts. Examples are Visual Source Safe, CVS etc. There are automated tools to support reviews and inspections also.
5. **Software maintenance tools :** Tools under this category are :
   a. **Comprehensive tools :** These tools are used to assist in human comprehension and visualization of the programs. Examples are Visual Studio, exref etc.
   b. **Re-engineering tools :** These are tools which allow the change of existing format of a program to a new format *i.e.*, a new language or new database or new technology in general.
6. **Software quality tools :** These tools are used to automate techniques like static analysis, reviews, inspection etc., to ensure software quality.
7. **Configuration management tools :** These tools support version control and other activities related to configuration management *i.e.*, management and control of changes made to the documents. Examples are Clear-Case, Change-man etc.
8. **Project management tools :** Tools under this category automate size estimations, cost estimation, schedule estimation, risk management activities etc. Some of the tools are MS Project, Excel, COCOMO, FPA etc.

Que 5.23. **Write a short note on planning and scheduling tools.**

Answer

Planning and scheduling tools are :

1. **WBS :** Refer Q. 2.2, Page 45C, Unit-2.
2. **PERT chart :** Refer Q. 2.29, Page 81C, Unit-2.
3. **Gantt chart :** Refer Q. 2.32, Page 86C, Unit-2.

Que 5.24. **Explain MS-Project.** **UPTU 2013-14, Marks 10**

Answer

1. Project management software such as Microsoft Project helps us to achieve our project goal on time and on budget.
2. Computer software can significantly aid in project management as a tool for recording, calculating, analysing, consolidating and presenting project details.
3. However, it is important to note that the software cannot produce or even guarantee a successful project plan.
4. Despite this, Microsoft Project assists you to develop a better plan. It does so in the following ways :
   a. MS Project requires you to specifically define the tasks in the project, making you think more carefully about project details.
   b. MS Project makes projections easier to calculate and more reliable. Based on the data you enter, the software will calculate a schedule that will show the various dates and the resources required to perform specific tasks.
   c. MS Project helps you detect inconsistencies and problems in the plan. It will detect when resources are scheduled for more hours than are available or when deadlines cannot be met.
   d. MS Project helps you communicate the plan to others as you can generate printed reports that make the "selling" of the plan to upper-level management, who must approve the plan, an easier task.
   e. Likewise, it is easier to communicate the plan to supervisors and workers, which simplifies securing their approval and co-operation.
   f. MS Project helps you track progress and detect potential difficulties once the project is underway. We can replace projected dates for the scheduled tasks with actual dates, as tasks are being performed.
   g. The software revises the schedule and the new projection will provide you with advance warning of potential delays (if any) so that you can take any required corrective measures.

5. Project management involves more than just opening a blank document and typing a list of tasks. There are "housekeeping" chores to be done and choices to be made about how to calculate the project schedule.
6. However, Microsoft Project does not have rigid requirements about the order in which you deal with these preliminaries.
7. We can begin by jotting down some ideas about tasks that you think might be required, and you can later adjust the scheduling calendar, enter the basic project information, revise the calculation and display options, and define the resources. In fact, you can execute all of the previous steps in any order.

VERY IMPORTANT QUESTIONS

Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.

Q. 1. What do you mean by software configuration management ?
Ans: Refer Q. 5.1.

Q. 2. Explain change request management.
Ans: Refer Q. 5.7.

Q. 3. Write a short note on risk management.
Ans: Refer Q. 5.14.

Q. 4. Discuss cost benefit analysis.
Ans: Refer Q. 5.18.

Q. 5. Discuss any software project management tool in detail.
Ans: Refer Q. 5.22.

☺☺☺