RESPONSIBILITY OF SOFTWARE PROJECT MANAGER

Write in brief: responsibilities and skills of software project manager. (S-2017, W-2015,S-2015)

Job responsibility

- Software project managers take the overall responsibility of project to success.
- The job responsibility of a project manager ranges from invisible activities like building up team morale to highly visible customer presentations.
- Most managers take responsibility for project proposal writing, project cost estimation, scheduling, project staffing, deciding software process, project monitoring and control, software configuration management, risk management, interfacing with clients, and presentations.
- These activities can be classified into project planning, and project monitoring and control activities.
- **Project planning**
  - ✓ Project planning involves estimating characteristics of the project and then planning the project activities based on the estimates made.
  - ✓ The project planning activity is undertaken after the feasibility study phase.
  - ✓ The initial project plans that are made are revised from time to time as the project progresses and more project data become available.
- **Project monitoring and control activities**
  - ✓ The project monitoring and control activities are undertaken once the development activities start with the aim of ensuring that the development proceeds as per plan and changing the plan whenever required.

Skills necessary for software project management

- A theoretical knowledge of different project management techniques is certainly necessary to become a successful project manager.
- In addition to having a good understand of the latest software project management techniques such as cost estimation, risk management, configuration management, project managers need good communication skills.
- However, some skills such as tracking and controlling the progress of the project, customer interaction, managerial presentations, and team building are largely acquired through experience.

METRICS FOR SOFTWARE PROJECT SIZE ESTIMATION

List metrics for Project Size Estimation. Explain any one Project Size Estimation metric. OR Discuss the Metrics for Size Estimation.(S-2017,S-2016)

- Estimation of the problem size is estimation of effort, time duration and cost of a software project.
- The size of a problem is obviously not the number of bytes that the source code occupies. It is neither the byte size of the executable code.
- Currently two metrics are popularly being used widely to estimate size:
  - ✓ Lines of code (LOC)
  - ✓ Function point (FP)

LINES OF CODE (LOC)

Explain Line of Code (LOC) and Function Point Technique. OR Explain line of code in size estimation metric.(S-2018,W-2017,W-2016,W-2015)

- LOC metric is very popular because it is the simplest to use. Using this metric, the project size is estimated by counting the number of source instructions in the developed program.
- Lines used for commenting the code and the header lines should be ignored.
- Determining the LOC count at the end of a project is a very simple job.
- To estimate the LOC count at the beginning of a project, project managers usually divide the problem into modules and each module into sub modules and so on, until the sizes of the different leaf-level modules can be approximately predicted.

Shortcomings (or Disadvantages) of Lines of Code (LOC) metric

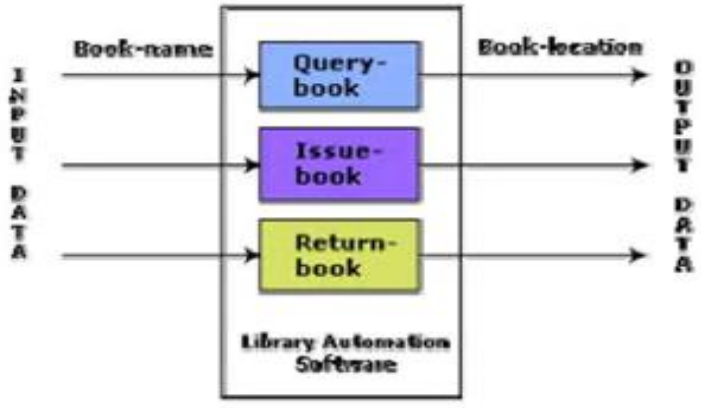
- LOC gives a numerical value of problem size that can vary with individual coding style – different programmers lay out their code in different ways. For example, one programmer might write several source instructions on a single line whereas another might split a single instruction across several lines. Of course, this problem can be easily overcome by counting the language tokens in the program rather than the lines of code.
- However, a more difficult problem arises because the length of a program depends on the choice of instructions used in writing the program. Therefore, even for the same problem, different programmers might come up with programs having different LOC counts.
- A good problem size measure should consider the overall complexity of the problem and the effort needed to solve it. That is, it should consider the effort needed to specify, design, code, test, etc. and not just the coding effort.
- LOC focuses on the coding activity alone; it only computes the number of source lines in final program.
- Larger code size does not necessarily imply better quality. Some programmers produce lengthy and complicated code as they do not make effective use of the available instruction set.

- If a programmer uses several library routines, then the LOC count will be lower. This would show up as smaller program size.
- It is very difficult to accurately estimate LOC in the final product from the problem specification. The LOC count can be accurately computed only after the code has been fully developed.
- Therefore, the LOC metric is little use to the project managers during project planning, since project planning is carried out even before any development activity has started. This is the biggest shortcoming of the LOC metric.

FUNCTION POINT (FP)

Explain function point (FP) in size estimation metric.(W-2017)

- Function Point metric overcomes many of the shortcomings of the LOC metric. One of the important advantages of using the function point metric is that it can be used to easily estimate the size of a software product directly from the problem specification.
- LOC metric, where the size can be accurately determined only after the product has fully been developed. The conceptual idea behind the function point metric is that the size of a software product is directly dependent on the number of different functions or features it supports.
- A software product supporting many features would certainly be of larger size than a product with less number of features. Each function when invoked reads some input data and transforms it to the corresponding output data.
- For example, the issue book feature (as shown in figure.) of a Library Automation Software takes the name of the book as input and displays its location and the number of copies available.



- The size of a product in function points (FP) can be expressed as the weighted sum of these five problem characteristics. The weights associated with the five characteristics were proposed empirically and validated by the observations over many projects.

Function point is computed in two steps. The first step is to compute the unadjusted function point (UFP).

UFP = (Number of inputs)*4 + (Number of outputs)*5 + (Number of inquiries)*4 + (Number of files)*10 + (Number of interfaces)*10

Number of inputs

- Each data item input by the user is counted. It must be noted that individual data items input by the user are not considered in the calculation of the number of inputs, but a group of related inputs are considered as a single input.
- For example, while entering the data concerning employee to pay roll software; the data items name, age, address, phone number, etc. are together considered as a single input. All these data items can be considered to be related, since they count as to a single employee.

Number of outputs

- The outputs considered refer to reports printed, screen outputs, error messages produced, etc. While outputting the number of outputs the individual data items within a report are not considered, but a set of related data items is counted as one input.

Number of inquiries

- Inquiries are user commands such as print-account-balance. Inquiries are counted separately.

Number of files

- Each logical file is counted. A logical file means groups of logically related data.

Number of interfaces

- Here the interfaces considered are the interfaces used to exchange information with other systems. Examples of such interfaces are communication links with other systems etc.
- Once the unadjusted function point (UFP) is computed, the technical complexity factor (TCF) is computed next. TCF refines the UFP measure by considering 14 other factors such as high transaction rates, throughput, and response time requirements, etc.
- Each of these 14 factors is assigned from 0 (not present) to 6 (strong influence). The resulting numbers are summed, gives the total degree of influence (DI).
- Now, TCF = (0.65+0.01*DI). As DI can vary from 0 to 70, TCF can vary from 0.65 to 1.35.
- Finally, FP=UFP*TCF.

PROJECT ESTIMATION TECHNIQUES

What is Estimation(S-2018)

- Estimation of various project parameters is a basic project planning activity. The important project parameters that are estimated include: project size, effort required to develop the software, project duration, and cost.
- These estimates not only help in deciding the project cost to the customer, but are also useful in resource planning and scheduling. There are three broad categories of estimation techniques:
  - ✓ Empirical estimation techniques
  - ✓ Heuristic techniques
  - ✓ Analytical estimation techniques

EMPIRICAL ESTIMATION TECHNIQUES

Explain Empirical Estimation Techniques. (W-2017,S-2017,S-2016)

- Empirical estimation techniques are based on making an educated guess of the project parameters. While using this technique, prior experience with development of similar products is helpful. Although empirical estimation techniques are based on common sense. Two popular techniques are:
  - ✓ Expert judgment technique and
  - ✓ Delphi cost estimation.

Expert Judgment Technique

- Expert judgment is one of the most widely used estimation techniques. In this approach, an expert makes an educated guess of the problem size after analyzing the problem. Usually, the expert estimates the cost of the different components (i.e. modules or subsystems) of the system and then combines them to compute the overall estimate.
- However, this technique is subject to human errors and individual partiality. Also, it is possible that the expert may not know some factors. Further, an expert making an estimate may not have experience and knowledge of all aspects of a project.
- For example, he knows the database and user interface parts but may not be very knowledgeable about the computer communication part. A more refined form of expert judgment is the estimation made by group of experts. Estimation by a group of experts minimizes factors such as individual oversight, lack of familiarity with a particular aspect of a project, personal partiality.

Delphi cost estimation

Explain Delphi Cost Estimation Technique.(S-2016,S-2015)

- Delphi cost estimation approach tries to overcome some of the shortcomings of the expert judgment approach.
- Delphi estimation is carried out by a team of a group of experts and a coordinator. In this approach, the coordinator provides each estimator with a copy of the software requirements specification (SRS) document and a form for recording his cost estimate.
- Estimators complete their individual estimates and submit to the coordinator. In their estimates, the estimators mention any unusual characteristic of the product which has influenced his estimation. The coordinator prepares and distributes the summary of the responses of all the estimators, and includes any unusual noted by any of the estimators.
- Based on this summary, the estimators re-estimate. This process is iterated for several rounds.
- However, no discussion among the estimators is allowed during the entire estimation process.
- After the completion of several iterations of estimations, the coordinator takes the responsibility of combining the results and preparing the final estimate.

HEURISTIC TECHNIQUE

Explain Heuristic Estimation Technique. OR Describe Heuristic Technique for Project Estimation.(S-2017,S-2016)

- Heuristic techniques assume that the relationships among the different project parameters can be modeled using suitable mathematical expressions. Once the basic (independent) parameters are known, the other (dependent) parameters can be easily determined by substituting the value of the basic parameters in the mathematical expression.

COCOMO

Explain COCOMO (S-2018)

- COCOMO (Constructive Cost Estimation Model) was proposed by Boehm (1981). According to the Boehm any software development project can be classified into one of the following three categories based on the development complexity: organic, semidetached, and embedded.
- **Organic:** A development project can be considered of organic type, if the project deals with developing a well understood application program, the size of the development team is small, and the team members are experienced in developing similar types of projects.
- **Semidetached:** A development project

- can be considered of semidetached type, if the development consists of a mixture of experienced and inexperienced staff.
- **Embedded:** A development project is considered to be of embedded type, if the software being developed is strongly coupled to complex hardware.
- According to Boehm, software cost estimation should be done through three stages: Basic COCOMO, Intermediate COCOMO, and Complete COCOMO.

Basic COCOMO Model

- The basic COCOMO model gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by the following expressions:
  Effort = a1 × (KLOC)^{a2} PM
  Tdev = b1 × (Effort)^{b2} Months

Where

  - ✓ KLOC is the estimated size of the software product expressed in Kilo Lines of Code
  - ✓ a1, a2, b1, b2 are constants for each category of software products
  - ✓ Tdev is the estimated time to develop the software, expressed in months
  - ✓ Effort is the total effort required to develop the software product, expressed in person months (PMs)
- According to Boehm, every line of source text should be calculated LOC.
- The values of a1, a2, b1, b2 for different categories of products (i.e. organic, semidetached, and embedded) as given by Boehm [1981] are summarized below. He derived the above expressions by examining historical data collected from a large number of actual projects.
- **Estimation of development effort:** For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:
  - ✓ Organic : **Effort = 2.4(*KLOC*)$^{1.05}$ PM**
  - ✓ Semi-detached : **Effort = 3.0(*KLOC*)$^{1.12}$ PM**
  - ✓ Embedded : **Effort = 3.6(*KLOC*)$^{1.2}$ PM**
- **Estimation of development time:** For the three classes of software products, the formulas for estimating the development time based on the effort are given below:
  - ✓ Organic : **Tdev = 2.5(*Effort*)$^{0.38}$ Months**
  - ✓ Semi-detached : **Tdev = 2.5(*Effort*)$^{0.35}$ Months**
  - ✓ Embedded : **Tdev = 2.5(*Effort*)$^{0.32}$ Months**
- From the effort estimation, the project cost can be obtained by multiplying the required effort by the manpower cost per month.
- **Example:** Assume that the size of an organic type software product has been estimated to be 32,000 lines of source code. Assume that the average salary of software engineers

be Rs. 15,000/- per month. Determine the effort required to develop the software product and the nominal development time.

- From the basic COCOMO estimation formula for organic software:
  Effort = **$2.4 \times (32)^{1.05} = 91$ PM**
  Development time = **$2.5 \times (91)^{0.38} = 14$ months**
  Cost required to develop the product = **91 x 15000**
  = **Rs. 1365000/-**

Intermediate COCOMO model

- Basic COCOMO model assumes that effort and development time are functions of the product size.
- Therefore, in order to obtain an accurate estimation of the effort and project duration, the effect of all relevant parameters must be taken into account.
- The intermediate COCOMO model recognizes this fact and refines the initial estimate obtained using the basic COCOMO expressions by using a set of 15 cost drivers (multipliers) based on various attributes of software development.
- For example, if modern programming practices are used, the initial estimates are scaled downward.
- If there are reliability requirements on the software product, this initial estimate is scaled upward.
- In general, the cost drivers can be classified as being attributes of the following items:
- **Product:** The characteristics of the product like reliability requirements of the product.
- **Computer:** Characteristics of the computer like execution speed required, storage space required etc.
- **Personnel:** experience level of personnel, programming capability, analysis capability, etc.
- **Development Environment:** use of the automation (CASE) tools used for software development.

Complete COCOMO model

Explain complete COCOMO Model.(W-2016,S-2015)

- A major shortcoming of both the basic and intermediate COCOMO models is that they consider a software product as a single entity.
- However, most large systems are made up several smaller sub-systems. These subsystems may have widely different characteristics.
- For example, some subsystems may be considered as organic type, some semidetached, and some embedded. Not only that the development complexity of the subsystems may be different, but also for some subsystems the reliability requirements may be high, for

some the development team might have no previous experience of similar development, and so on.

- The complete COCOMO model considers these differences in characteristics of the subsystems and estimates the effort and development time as the sum of the estimates for the individual subsystems.
- The following development project can be considered as an example of application of the complete COCOMO model.
- A distributed Management Information System (MIS) product for an organization having offices at several places across the country can have the following sub-components:
  - ✓ Database part
  - ✓ Graphical User Interface (GUI) part
  - ✓ Communication part
- Of these, the communication part can be considered as embedded software. The database part could be semi-detached software, and the GUI part organic software. The costs for these three components can be estimated separately, and summed up to give the overall cost of the system.

ANALYTICAL ESTIMATION TECHNIQUES

Explain Analytical Estimation Technique (W-2017)

- Analytical estimation techniques derive the required results starting with basic assumptions regarding the project. Analytical techniques do have scientific basis. Halstead's software science is an example of an analytical technique.
- Halstead's software science is an analytical technique to measure size, development effort, and development cost of software products. Halstead used primitive program parameters to develop the expressions for overall program length, volume, effort, and development time.
- So for predicting software estimation it performs both empirical and heuristic techniques.
- For a given program, you need to calculate:
- **Length:** The length of a program as defined by total usage of all operators and operands.
- **Vocabulary:** The program vocabulary is the number of unique operators and operands.
- **Program Volume:** The program volume is the total number of operators and operands.
- **Difficulty:** The difficulty metric indicates how difficult a program is to write or understand.
- **Effort:** The effort required to develop a program can be obtained by E = Difficulty *Volume.

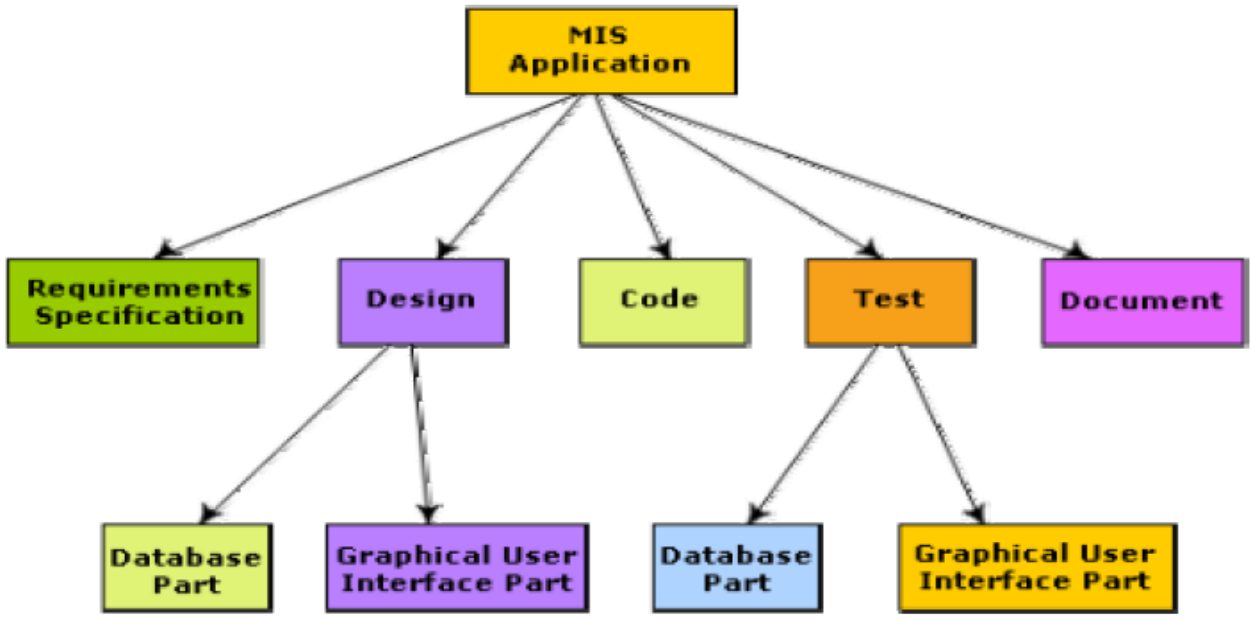
PROJECT SCHEDULING

- Project scheduling is an important project planning activity. It involves deciding which tasks would be taken up when. To schedule the activities, a software project manager needs to do the following:
  - ✓ Identify all the tasks needed to complete the project.
  - ✓ Break down large tasks into small activities.
  - ✓ Determine the dependency among different activities.
  - ✓ Estimates for the time durations necessary to complete the activities.
  - ✓ Allocate resources to activities.
  - ✓ Plan the starting and ending dates for various activities.
- The first step in scheduling a software project involves identifying all the tasks necessary to complete the project. A good knowledge of the project and the development process helps the managers to effectively identify the important tasks of the project.
- Next, large tasks are broken down into a logical set of small activities which would be assigned to different engineers. The work breakdown structure helps the manager to breakdown the tasks.
- After the project manager has broken down the tasks and created the work breakdown structure, he/she has to find the dependency among the activities. Dependency among the different activities determines the order in which the different activities would be carried out. If an activity A requires the results of another activity B, then activity A must be scheduled after activity B.
- In general, the task dependencies define a partial ordering among tasks, i.e. each task may precede a subset of other tasks, but some tasks might not have any precedence ordering defined between them. The dependencies among the activities are represented in the form of an activity network.
- Once the activity network representation has been worked out, resources are allocated to each activity. Resource allocation is typically done using a Gantt chart.
- The PERT chart representation is suitable for program monitoring and control. For task scheduling, the project manager needs to decompose the project tasks into a set of activities. The time frame when each activity is to be performed is to be determined.
- The end of each activity is called milestone. The project manager tracks the progress of a project by monitoring the timely completion of the milestones. If he observes that the milestones start getting delayed, then he has to carefully control the activities, so that the overall deadline can still be met.

Work breakdown structure

Explain work breakdown structure (W-2017,W-2016) OR Write a short note on activity network, Work breakdown structure. OR Explain the work break down structure.(S-2017,W-2015)

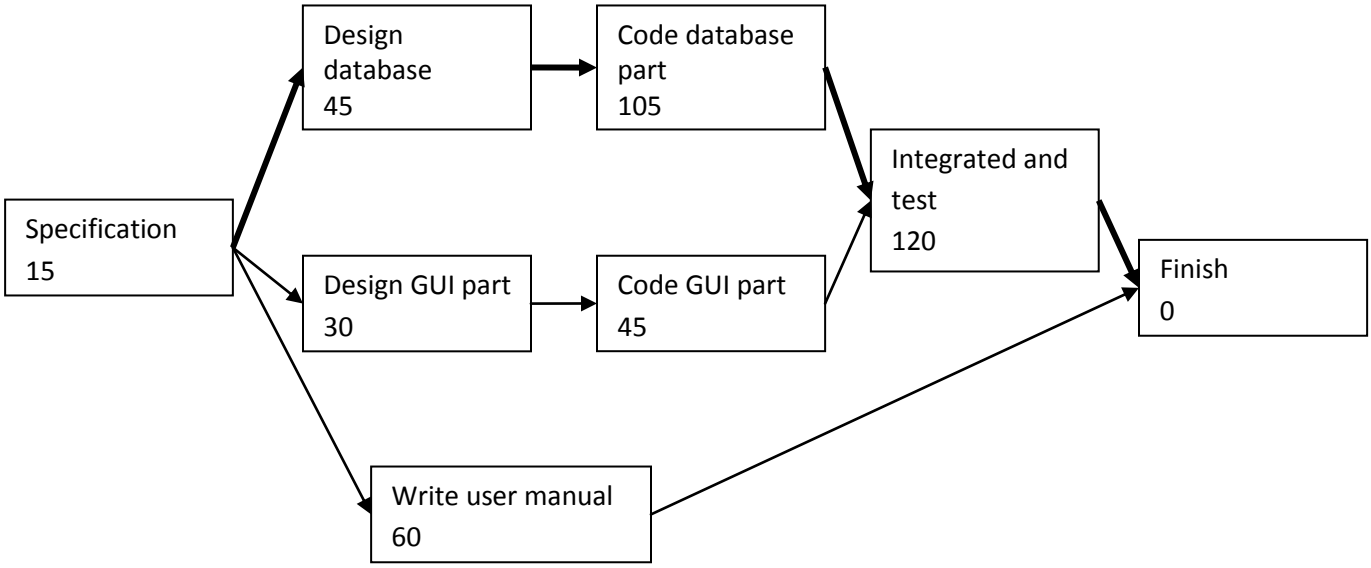
- Work Breakdown Structure (WBS) is used to decompose a given task set recursively into small activities. WBS provides a notation for representing the major tasks need to be carried out in order to solve a problem.
- The root of the tree is labeled by the problem name. Each node of the tree is broken down into smaller activities that are made the children of the node.
- Each activity is recursively decomposed into smaller sub-activities until at the leaf level, the activities requires approximately two weeks developing.
- Figure represents the WBS of MIS (Management Information System) software.
- While breaking down a task into smaller tasks, the manager has to make some hard decisions.
- To complete a project in the time, the manager needs to break large tasks into smaller ones.
- However, it is not useful to subdivide tasks into units which take less than a week.



Activity networks and critical path method

- WBS representation of a project is transformed into an activity network by representing activities identified in WBS along with their interdependencies.
- An activity network shows the different activities making up a project, their estimated durations, and interdependencies (as shown in figure). Each activity is represented by a rectangular node and the duration of the activity is shown alongside each task.

- Managers can estimate the time durations for the different tasks in several ways. One possibility is that they can assign durations to different tasks.



- A possible alternative is to let engineer himself estimate the time for an activity he can assigned to. However, some managers prefer to estimate the time for various activities themselves.
- Many managers believe that good schedule motivates the engineers to do a better and faster job. However, if not careful schedule then it cause for schedule delays.
- A good way to achieve accurately in estimation of the task durations without creating undue schedule pressures is to have people set their own schedules.

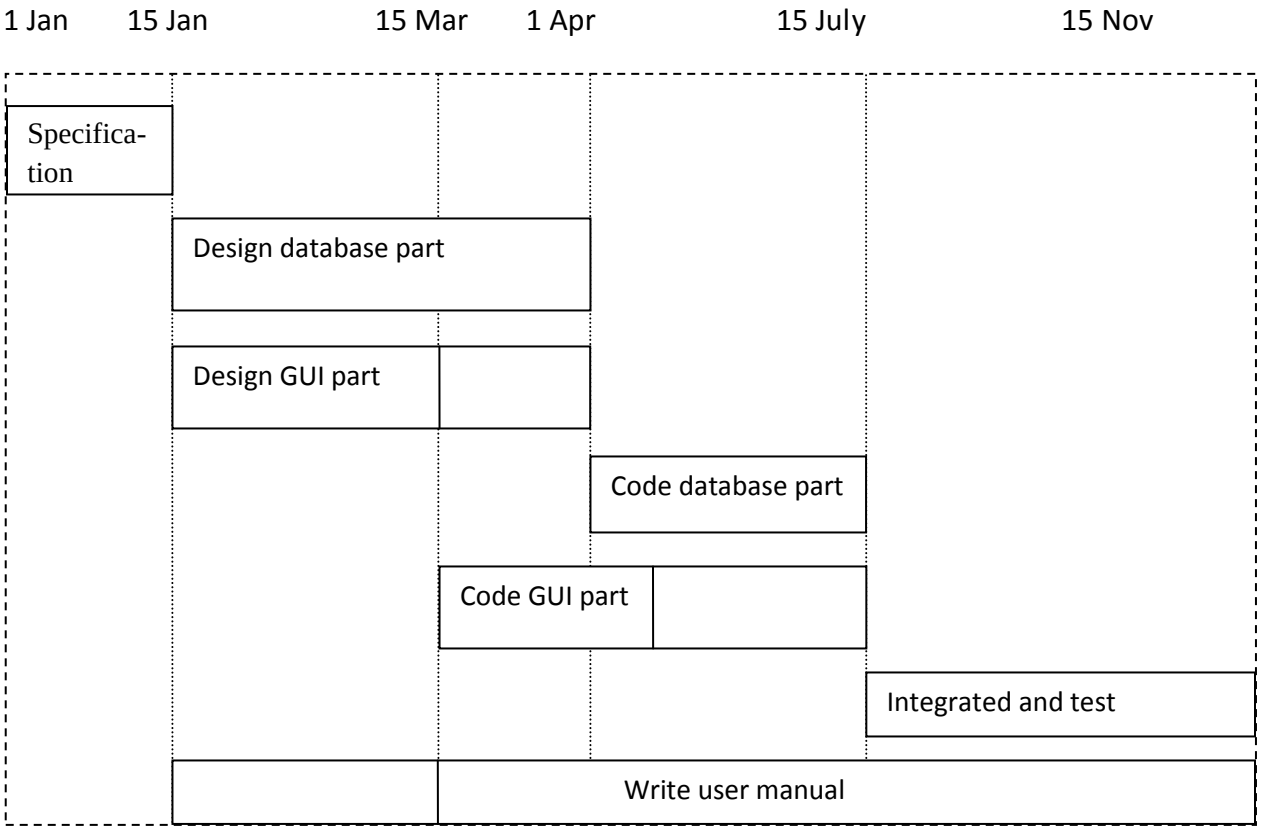
Critical Path Method (CPM)

- CPM is based on mathematical calculations and it is used for scheduling project activities.
- The project manager identifies the critical activities of the project from the beginning of the project.
- The series of critical activity is known a critical path of project. It is longest path through the network.
- In the critical path method, the critical activities of a project are identified. These are the activities that have a direct effect on the completion date of the project.

Gantt chart

Explain Gantt chart (W-2017,W-2016)

- Gantt charts are used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts are useful for resource planning.
- A Gantt chart is a special type of bar chart where each bar represents an activity. The bars are drawn along a time line. The length of each bar is proportional to the duration of time planned for the corresponding activity.
- This allows you to see: various activities, when each activity begins and ends, how long each activity is scheduled, start and end date of the whole project.
- A Gantt chart representation for the MIS problem is shown in the figure.



Project Monitoring and Control

What do you mean by project monitoring and control? (S-2017)

- Project manager has to monitor the project continuously to ensure that it is progressing as per the plan.
- If any delay in reaching a milestone is predicated then corrective action might have to be taken. Action may be reworking all the schedules and producing a new schedule.

- PERT chart are useful in project monitoring and control.
- If any delay occurs in any phase then the entire project would get delayed.

Define Risk. (S-2018, W-2017, S-2016)

Risk is an expectation of loss, a potential problem that may or may not occur in the future. It is generally caused due to lack of information, control or time. A possibility of suffering from loss in software development process is called a software risk. Loss can be anything, increase in production cost, development of poor quality software, not being able to complete the project on time. Software risk exists because the future is uncertain and there are many known and unknown things that cannot be incorporated in the project plan

RISK MANAGEMENT

Explain various categories of Risk Management. (S-2017,W-2016,W-2015)

- The aim of risk management is to reducing the impact of all kind of risks that might affect a project. Risk management consists of three essential activities: risk identification, risk assessment, and risk containment.

RISK IDENTIFICATION

- A software project can be affected by a large variety of risks. In order to be able to systematically identify the important risks which might affect a software project, it is necessary to categorize risks into different classes.
- The project manager can then examine which risks from each class are relevant to the project. There are three main categories of risks which can affect a software project:

Project risks

Discuss Different types of risks involved in software development(S-2018,W-2017,S-2016)

- Project risks concern varies forms of budgetary, schedule, personnel, resource, and customer-related problems. An important project risk is schedule. It is very difficult to monitor and control a software project.
- It is very difficult to control something which cannot be seen. For any manufacturing project, such as manufacturing of cars, the project manager can see the product taking shape. He can for instance, see that the engine is fitted, after that the doors are fitted, and the car is getting painted, etc. Thus he can easily assess the progress of the work and control it.
- The invisibility of the product being developed is an important reason why many software projects suffer from the risk of schedule.

Technical risks

- Technical risks concern design, implementation, interfacing, testing, and maintenance problems.

- Technical risks also include ambiguous specification, incomplete specification, changing specification, technical uncertainty. Most technical risks occur due to the development team's insufficient knowledge about the project.

Business risks

- This type of risks include risks of building an excellent product that no one wants, losing budgetary or personnel commitments, etc.

Example

- Let us consider the satellite based mobile communication product. The project manager can identify several risks in this project. We can classify them appropriately as:
  - ✓ What if the project cost extend to a large extent than what was estimated? – project risk
  - ✓ What if the mobile phones become too large for people to conveniently carry? – Business risk

RISK ASSESSMENT

- Risk assessment involves identifying risk, analyzing them and then assigns priority to them on the basis of the analysis.
- The objective of risk assessment is to rank the risks in terms of their damage. For risk assessment, first each risk should be rated in two ways:
  - ✓ The probability of a risk coming true (denoted as r).
  - ✓ The result of the problems associated with that risk (denoted as s).
- Based on these two factors, the priority of each risk can be computed:
  p = r * s
- Where, p is the priority with which the risk must be handled, r is the probability of the risk becoming true, and s is the result of damage caused due to the risk becoming true. If all identified risks are prioritized, then the most likely and damaging risks can be handled first and reject procedures can be designed for these risks.

RISK CONTAINMENT

- After all the identified risks of a project are assessed, plans must be made to containment the most damaging and the most likely risks.
- Different risks require different containment procedures. In fact, most risks require expertness on the part of the project manager in handling the risk.
- There are three main strategies to plan for risk containment:
  - ✓ **Avoid the risk:** This may take several forms such as discussing with the customer to change the requirements to reduce the scope of the work.

- ✓ **Transfer the risk:** This strategy involves getting the risky component developed by a third party.
- ✓ **Risk reduction:** This involves planning ways to containment the damage due to a risk.

- To choose between the different strategies of handling a risk, the project manager must consider the cost of handling the risk and the corresponding reduction in risk.
- For this we may compute the risk leverage of the different risks. Risk leverage is the difference in risk divided by the cost of reducing the risk.

Risk leverage = (Risk before reducing - Risk after reducing) / cost of reducing

What is RMMM plan(S-2018)

- It is a part of the software development plan or a separate document.
- The RMMM plan documents all work executed as a part of risk analysis and used by the project manager as a part of the overall project plan.
- The risk mitigation and monitoring starts after the project is started and the documentation of RMMM is completed.
- **Risk Mitigation, Monitoring and Management (RMMM)**
  Risk analysis --- support the project team in constructing a strategy to deal with risks.

There are three important issues considered in developing an effective strategy.

- **Risk avoidance or mitigation -** It is the primary strategy which is fulfilled through a plan.
- **Risk monitoring -** The project manager monitors the factors and gives an indication whether the risk is becoming more or less.
- **Risk management and planning -** It assumes that the mitigation effort failed and the risk is a reality.

2 marks QUE-ANS

Explain organic type of project development.(W-2015)

A software development project can be considered of organic type, if the project deals with developing a well understood application program. In organic type, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects. Data processing programs can be considered as Organic type.

Give full form of: LOC, COCOMO (W-2015,S-2015)

- COCOMO - Constructive Cost Model
- LOC - Lines of code

List Project Estimation Techniques.(S-2017,S-2016)

- **EMPIRICAL ESTIMATION TECHNIQUES**
  - ✓ Expert judgment technique and
  - ✓ Delphi cost estimation.
- **HEURISTIC TECHNIQUE**
  - ✓ Cocomo
- **ANALYTICAL ESTIMATION TECHNIQUES**

Give full form of: CPM, WBS, PERT (W-2015, S-2015)

- Critical path method (CPM)
- Work- breakdown structure (WBS)
- Program Evaluation Review Technique