COCOMO – II Model of Estimation

COCOMO – II Model

- The COCOMO-II Model uses 3 estimation models to estimate effort and cost models are as follows:-
  - Application Suite Model
  - Early Designed Level Model
  - Post Architectural Model

Application Suite Model

- The model is used where software can be decomposed into several components and each components can be described in object point.
- The objects are screen, report and 3GL components which are easy to identify and count when the software system is split into sub-system components.
- Object point should be noted that are not object classes.

Application Suite Model …

- The candidates for the object points are screen, report and the number of 3GL modules to supplement the 4GL code.
- The objects are given object points depending on the level of complexities.
- The advantage of using object point is that they can be estimated from high level design of the software and they are only number of screen, report and 3GL modules.
- The complexity level is the judgement of the Software Engineering.

Application Suite Model …

Object Points			
Objects	**Simple**	**Complex**	**Very Complex**
• Screen	1	2	3
• Report	2	5	8
• 3GL Module	4	10	-

Application Suite Model …

- The object point counts need modification by the way of reduction and the software may use reusable components and library.
- Therefore revised object point (ROP) is given as ROP = OBJECT POINT X (100 – (% of Reuse))/100.
- For this ROP, the effort in man month is computed using the productivity constant based on the software development team's experience and capacity.

Application Suite Model …

Developers level of Experience and Maturity	Very low	Low	Normal	High	Very High
Productivity constant(NOP per month)	4	7	13	25	50

Application Suite Model …

- Man Month Effort (MME)=ROP/Productivity Constant.
- Example: - Let Object Point=40, Reuse(%)=10, then find ROP=?
- ROP=Object Point x (100 – (reuse (%)))/100 = 40*(100 – 10) / 100 = 40 * 90 / 100 = 36.
- Now, if the development team experience and maturity level is normal, the productivity constant is 13 and hence MME = ROP / Productivity Constant = 36 / 13 = 3 Man Months.
- The model is used to estimate the effort at the prototype level where the requirements are not clear.

COCOMO – II Model

- The COCOMO-II Model uses the base equation **$MME=A*(SIZE)^B$**, Where
  - **MME** is the **M**an **M**onth **E**ffort
  - **A** is the Constant representing nominal productivity
  - **B** is the Factor integrating economy/dis-economy of scale(Also known as Scale Factor)
  - SIZE = KLOC (Thousand Lines of Codes).
- If B=1, the software does not have any impact on MME.
- If B<1, the MME have positive impact.
- If b>1, the MME have negative impact.

COCOMO – II Model …

- This model uses five factors for arriving at economics and dis-economics of scale which together concludes the B-factor.
  - Precedentness i.e. PREC
  - Flexibility i.e. FLEX
  - Risk Resolution i.e. RESL
  - Team Cohesion i.e. TEAM
  - Process Maturity i.e. PMAT

COCOMO – II Model …

Factor Code	Factor Name	Very Low	Low	Normal	High
PREC	Precedentness	6.20	4.96	3.72	2.48
FLEX	Flexibility	5.07	4.05	3.04	2.03
RESL	Risk Resolution	7.07	5.65	4.24	2.83
TEAM	Team Cohesion	5.48	4.38	3.29	2.19
PMAT	Process Maturity	7.80	6.24	4.68	3.12

COCOMO – II Model …

- **PERC i.e. Precedentness**
  - Understanding and experience of developing similar software.
  - If the degree of learning benefit which can be given to new software is very low, then rating is 6.20.
- **FLEX i.e. Flexibility**
  - Flexibility measured based on the degree of freedom and comfort level the developer has, based on the level of conformance required to either pre – established or customer-laid down standards, specifications, tools and schedules.

COCOMO – II Model …

- **RESL i.e. Risk Resolution**
  - If the organization and the software development team has considerable experience in risk management and is in a position to develop an RMMM plan for the project, then the level of risk resolution is very high and the rating value is 2.83.
- **TEAM i.e. Team Cohesion**
  - If the capacity of the organization to provide a development team whose members will work towards common objectives in cohesion is nominal, then the rating value is 3.29.

COCOMO – II Model …

- **PMAT i.e. Process Maturity**
  - This is the SEI-CMM level used for describing and organization's development maturity.
  - If the CMM level is very low i.e. '1', then the rating value is 7.80.

COCOMO – II Model …

- Let us assume that in a given situation, the organization's level on these five factors is very low, then B = 0.91 + 0.01 * (6.20 + 5.07 + 7.07 + 5.48 + 7.80) = 0.91 + 0.01* 31.62 = 0.91 + 0.3162 =1.2262
- A=13, size based on FPA is 10 KLOC
- Then MME=13*$(10)^{1.2262}$ = 220 Man Month
- In case of the organization's level on these five factors is high, then
- B = 0.91 + 0.01*(2.48 + 2.03 + 2.83 + 2.19 + 3.12) = 0.91 + 0.01*12.65 = 0.91 + 0.1265 = 1.0365 = 1.04
- Then MME=13*$(10)^{1.04}$ =142 Man Months

Post Architectural Model

- There are some other factors which are relevant as they have larger impact on MME.
- If we consider these factors, then we calculate MME which is modified value.
- There are 16 factors which are significant.
- They are
  - RELY
  - DATA
  - CPLX

Post Architectural Model …

- RUSE
- DOCU
- TIME
- STOR
- PVOL
- ACAP
- PCAP
- PCON
- AEXP

Post Architectural Model …

- PEXP
- LTEX
- TOOL
- SITE

- COCOMO-II equation for MME (Modified) = MME * (product of ratings of 16 factors).

Post Architectural Model …

Factor	Factor Name	Levels and Factors			
		Very Low	Low	Nominal	High
Product Factor					
RELY	Software Reliability	0.82	0.92	1.00	1.10
DATA	Database Size	0.80	0.90	1.00	1.14
CPLX	Software Complexity	0.73	0.87	1.00	1.17
RUSE	Required Reusability	0.85	0.95	1.00	1.07
DOCU	Documentation	0.81	0.91	1.00	1.11

Post Architectural Model …

Factor	Factor Name	Levels and Factors			
		Very Low	Low	Nominal	High
Platform Factors					
TIME	Time Constraint on Execution	NRA	NRA	1.00	1.11
STOR	Main Storage Constraint	NRA	NRA	1.00	1.05
PVOL	Platform Volatility	NRA	NRA	1.00	1.15

Post Architectural Model …

Factor	Factor Name	Levels and Factors			
		Very Low	Low	Nominal	High
Personnel Factor					
ACAP	Analyst Capability	1.42	1.19	1.00	0.85
PCAP	Programmer Capability	134	1.15	1.00	0.88
AEXP	Analyst Experience	1.22	1.10	1.00	0.88
PEXP	Programmer Experience	1.19	1.09	1.00	0.91
LTEX	Language and Tools Experience	1.20	1.09	1.00	0.91
PCON	Personnel Continuity	1.29	1.12	1.00	0.90

Post Architectural Model …

Factor	Factor Name	Levels and Factors			
		Very Low	Low	Nominal	High
Project Factor					
TOOL	Use of Software Tool	1.17	1.09	1.00	0.90
SITE	Site Environment	1.22	1.09	1.00	093

Post Architectural Model …

- **RELY i.e. Software Reliability**
  - Failure does not cause any inconvenience, rating = very low.
  - Failure is fatal, rating is high.
- **DATA i.e. Database Size**
  - Database size is measured as D/P, where D: database in bytes, P: lines of codes.
  - If D/P<10, Rating is very low, if D/P>1000, rating is high.

Post Architectural Model …

- **CPLX i.e. Software Complexity**
  - Few control options, simple, few calculations, simple data management, then rating is very low.
  - If all these are high, then rating is high.
- **RUSE i.e. Required Reusability**
  - No requirements of reusability i.e. customized software, then the rating is very low.
  - If the reusability is substantial, then the rating is high.

Post Architectural Model …

- **DOCU i.e. Documentation**
  - Documentation need is standard but low, then rating is very low.
  - If documentation need is very high both pre and post development, then rating is high.
- **TIME i.e. Time Constraint on Execution**
  - No constraint, then rating is very low.
  - Available time is almost equal to the execution time, then rating is high.

Post Architectural Model …

- **STOR i.e. Main Storage Constraint**
  - No constraint due to available very high storage, then rating is very low.
  - Available storage is almost equal to required storage, then rating is high.
- **PVOL i.e. Platform Volatility**
  - If platform is stable, then rating is very low.
  - If platform is unstable and may change rapidly, then the rating is high.

Post Architectural Model …

- **ACAP i.e. Analyst Capability**
  - In experience and lack of knowledge, then the rating is very low.
  - If analyst scores high on experience then the rating is high.
- **PCAP i.e. Programmer Capability**
  - In experience and lack of knowledge, then the rating is very low.
  - If analyst scores high on experience then the rating is high.

Post Architectural Model …

- **PCON i.e. Personnel Continuity**
  - Very low turnover, then the rating is high. 50% or more would leave, then the rating is very low.
- **AEXP i.e. Analyst Experience**
  - Minimum Experience, then the rating is very low.
  - More than adequate experience then the rating is high.

Post Architectural Model …

- **PEXP i.e. Programmer Experience**
  - Minimum Experience, then the rating is very low.
  - More than adequate experience then the rating is high.
- **LTEX i.e. Language and Tools Experience**
  - Minimum Experience, then the rating is very low.
  - More than adequate experience then the rating is high.
- **TOOL i.e. Use of Software Tools**
  - No use or occasional use, then the rating is very low.
  - Sustained use of variety of tools, then the rating is high.

Post Architectural Model …

- **SITE i.e. Site Environment**
  - Single site, single location, not more than one or two sponsors, then the rating is very low.
  - Multiple sites, multiple partners, number of locations but supported by good communication infrastructure, then the rating is high.
  - If problem of communication are not there, rating is very low.

Post Architectural Model …

- Now considering the case where all the ratings are very low, then
  - Product of all 16 ratings = 0.82* 0.80* 0.73* 0.85* 0.81* 1.42* 1.34*1.22*1.19*1.20*1.29*1.17*1.22=2.01
  - MME=220*2.01=442 Man Months
- Now considering the case where all the ratings are high, then
  - Product of all 16 ratings = 1.10* 1.14*1.17*1.07*1.11*1.11* 1.05*1.15*0.85*0.88*0.88*0.91*0.91*0.90*0.90*0.93=0.98
  - MME=142*0.98=139 Man Months

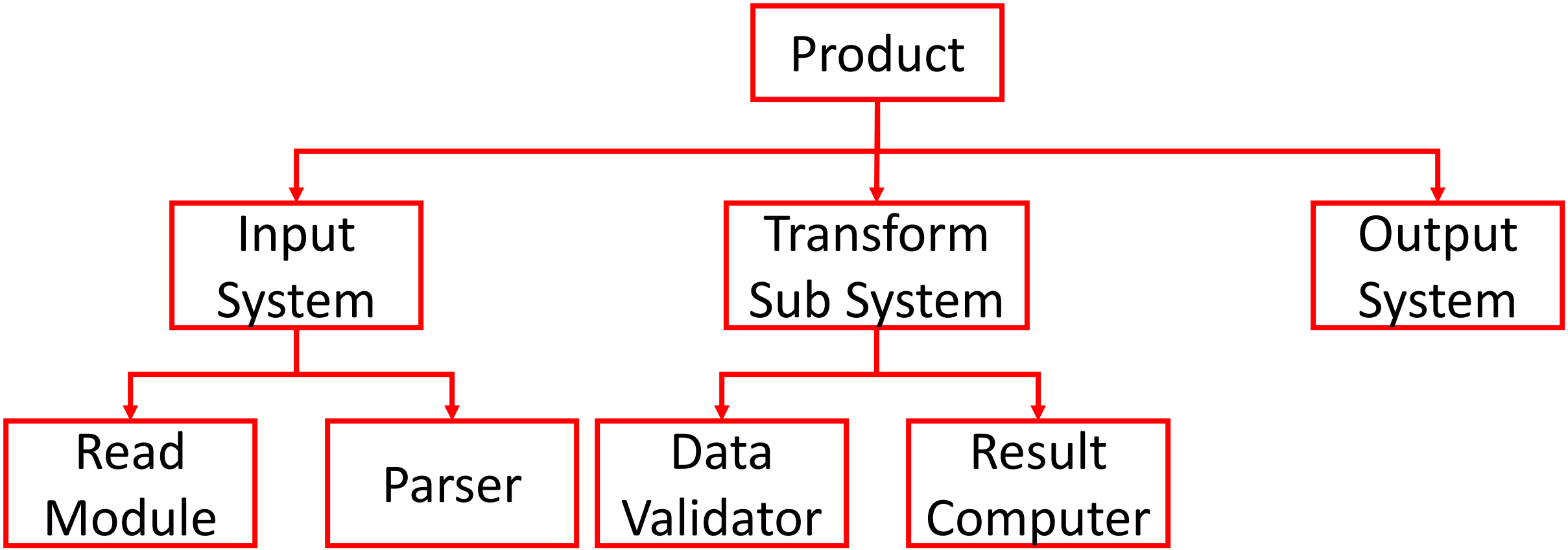
Work Breakdown Structure

- Expert judgement and group consensus are tow down techniques.
- The work breakdown structure is bottom up approach.
- A work breakdown structure is a hierarchical char that accounts for an individual part of the system.
- A work breakdown structure chart indicates product hierarchy and process hierarchy.

Product Hierarchy

- It identifies the product component and indicates the manner in which the components are interconnected.
- A work breakdown chart of process hierarchy identifies the work activity and relationship among those activities using the techniques.
- Costs are estimated by assigning cost to each individual component in the chart and then adding the cost.

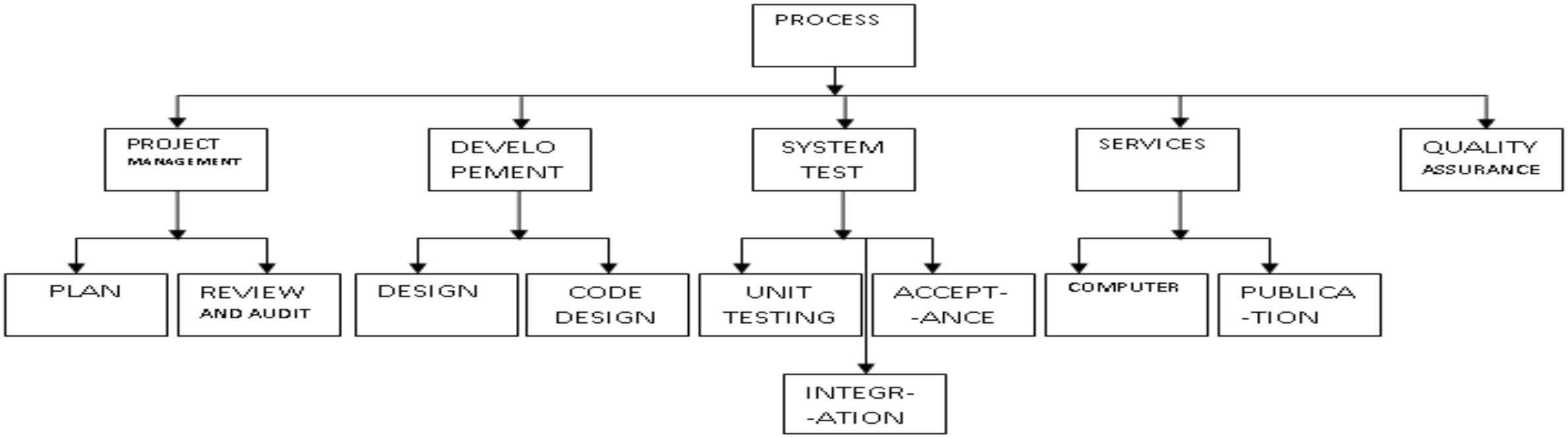
Product Hierarchy …



Process Hierarchy

- Processes can be broken down into components like project management, development, system test, services and quality assurance, which can be further broken down into sub components. Some planners uses both product and project hierarchy of work breakdown structure chart for cost estimation.
- The primary advantage of this technique is, it identifies the account for various process and product factors and in making explicit exact cost which are included in the estimates.

Process Hierarchy …



Cost Estimation

- Human Resource Cost
  - Skills
  - Knowledge
  - Numbers
- Development Resource Cost
  - Platform
  - Software
  - Tools

Cost Estimation …

- Personal Cost
- Hardware Cost
- Software Cost
- Training Cost
- Marketing Cost
- Outsourcing Cost

Planning the Development Process

- The first consideration is to define a product life cycle model.
- The software life cycle includes all activities required to define, develop, test, deliver, operate and maintain a software product.
- The Life cycle model includes
  - Phased Model
  - Cost Model
  - Prototype Model.

Phased Life Cycle Model

- In phased life cycle model segment, the software life cycle is a series of successive activities.
- Each phase requires well defined input information, neutralizes well defined processes and result in well defined product.
- Planning and requirement definition are two major activities of analysis phase.
- They include understanding the customer's problem by performing the feasibility study, developing a recommended study, developing the acceptance criteria and planning the development process.

Phased Life Cycle Model …

- Requirement definition is concerned with identifying the basic function of the software component. In a hardware / software people system, emphasis is placed on what the software will exactly do and what are its constraints. The product requirement definition is the specification that describe the processing environment, the required software function, performance constraint of the software i.e. size, speed, machine, configuration etc., exception handling, subsets and implementation priorities, probable changes and light modification and the acceptance criteria for the software.

Phased Life Cycle Model …

- In phase model, the software design follows analysis. Architectural design includes identifying the software components, decoupling and decomposing them into processing modules and conceptual data structure and specifying the interconnection among the components. Detail design involves adaptation of existing code, modification of standard algorithm, invention of new algorithms, design of data representation and packaging of the new software products.

Phased Life Cycle Model …

- The implementation phase involves transaction of design specification into source code and debugging, documentation and unit testing of source code. Errors discovered during the implementation test includes the errors in the data interfaces between routines, logical errors in the algorithm, errors in the data structure layout and failures to account for various processing cases.

Phased Life Cycle Model …

- In addition, source code may contain requirement errors that indicate the failure to capture the customer needs, in the requirement document, design error that reflect failure to translate requirement into correct design specification and an implementation errors that reflect the failure to correctly translate design specification into source code. One of the primary goal of phased approval to software development is to eliminate requirement and design errors from an evolving software product before implementation begins.

System Testing

- System Testing involve two types of testing. They are
  - Integration
  - Acceptance

Documentation

- Documentation is an important component of software.
- It can be a paper or electronic document. It could be part of software and available on line.
- It could be delivered separately and available offline.
- A complex software needs normally following documentation: -
  - System Manual
  - User Manual
  - System Maintenance Manual
  - Operations Manual

System Manual

- It provides information about system scope, design, architecture, system flow, technology details, interfaces used in the system etc.
- It is also backed by the requirement analysis and modeling, specification, important functions and features that the system provides.
- The source code is a part of system manual, although in general it is not given to the customer.
- Only executables of the entire system is delivered, demonstrated and implemented.

User Manual

- System user manual is an instruction for the users of the system.
- It provides screen by screen, interface by interface, file by file usage instructions and its impact elsewhere.
- It is also initially used for training as well as guiding users of the system.

System Maintenance Manual

- It deals with the routine system maintenance.
- It is used by the system administrator and coordinator to cater the problems like user problems, system problems, maintenance of files, databases, backups, security issues, system logs etc.

Operations Manual

- It deals with system operations as it functions.
- It provides guidelines to users to understand the implications of any action for the system.
- It provides transparency and insight as to how the system operates or responds to the action taken by the user.

Planning Software Project

- Defining the Problem
- Planning is a primary stage of the development.
- Lack of planning is the cause of schedule slippage, cost over-runs, poor quality and a high maintenance cost for the software.
- Careful planning is required for both development process and the work product in order to avoid these problem.
- Develop a definitive statement of the problem to be solved, which includes the description of the present situation, problem constraints and the statement of the goal to be achieved.

Planning Software Project …

- The problem statement should we trace in customer terminology.
- Justify a computerized strategy solution for the problem. Not the customer problem from the customer point of view. For example Inventory Problem, Payroll Problem etc. and not as the problem of sorting algorithm and relational database.
- Identify the functions to be provided by and the constraints on the hardware system, the software sub system and the people sub system.

Planning Software Project …

- The interaction among sub system must be established and development and operational constraints must be determined by each subsequent sub system.
- The function to be performed by each major sub system must be identified.

Goals and Requirement for Planning Software Project

- Determine system level goals and requirement for the development process and the work product given a consize statement of the problem and an indication of the constraint that exist for a solution, preliminary goals and requirement can be formulated.
- Goals are target to be achieved.
- Goals are applied to both the development process and the work product.
- Goals can either be qualitative or quantitative.

Goals and Requirement for Planning Software Project …

- High level goals and requirement can often be explained in terms of quality attribute that the system should process.
- The quality attributes are as follows:-
  - Portability
  - Reliability
  - Efficiency
  - Accuracy
  - Errors
  - Robustness
  - Correctness

Qualitative Goal Process

- The development process should enhance the professional skills of the quality and assurance personnel.

Quantitative Goal Process

- The system should be delivered on time.
- The system should reduce the cost of the transaction.