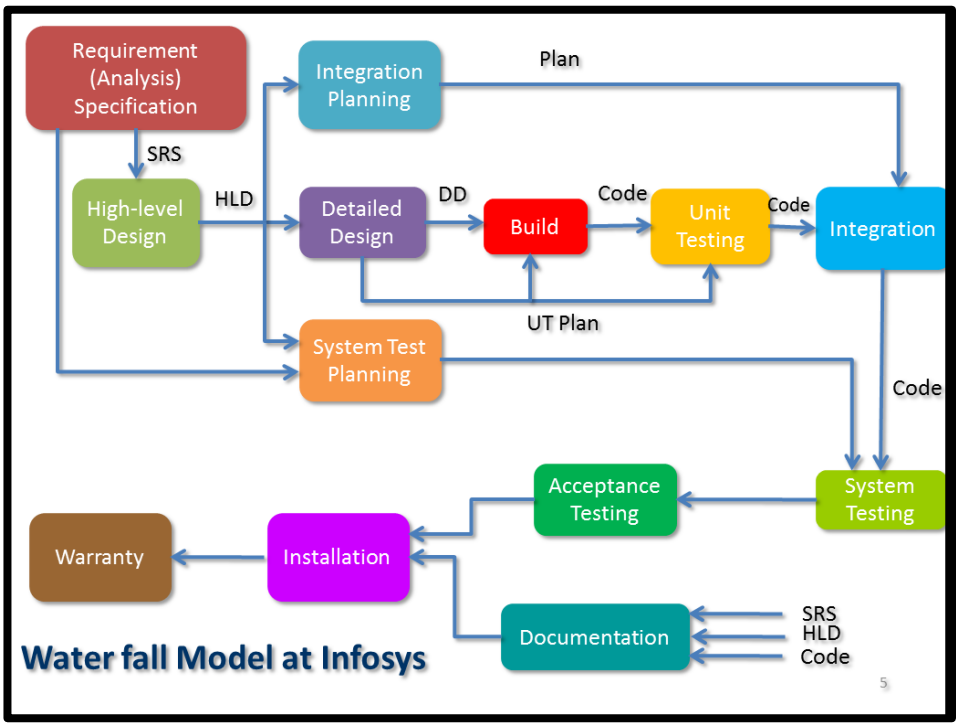
Pankaj Jalote, CMM in Practice, Person Education. **[PJ#]**						
Pankaj Jalote, Software Project Management in Practice, Addison-Pearson. **[SPMIP#]**						
Unit	**Unit Name**	**Sub Unit**	**Topics**	**No. of Lectures**	**Reference Chapter/Additional Reading**	**Teaching Methodology**
2	Project Planning with Effort Estimation	2.1	Development process and Process Tailoring	3	PJ#4, Page No 73-94, SPMIP#2, Page No. 24	Chalk and Talk, Topic Presentation.
		2.2	Software Metrics and Process Management	2	PJ#5, Page No 97-102	
		2.3	Process Database	1	PJ#5, Page No 103-110, SPMIP#8, Page No. 136	
		2.4	Process Capability Baseline	1	PJ#5, Page No 110-117, SPMIP#8, Page No. 140	
		2.5	Types of Effort Estimation	2	PJ#6, Page No 119-135, SPMIP#3, Page No. 40	
		2.6	Scheduling	1	PJ#6 Page No 136-142	

2.1 Development process and Process Tailoring

- It is a process by which the user requirements are stimulated and software satisfying these requirements is designed, built, testes and delivered to the customer.
- When new application is being developed, at that time we can use development process.
- Several models are available for software development like, waterfall model (widely used model), iterative enhancement, prototyping & spiral.

- The development process used at Infosys resembles the waterfall model, except that the traditional phase have been broken into "smaller " phase or stage to allow parallel execution of some phase.



- High-Level Design
  - It is the phase of the life cycle when a logical view of the computer implementation of the solution to the customer requirements is developed.
  - It contain two main components:
    - Functional architecture of the application.
    - Database design.
  - Steps in the process is as follows:
    - Define standard

- Determine/design details of operating environment.
- Do functional design
- Develop physical database design

Title	Description
Participant	1. Design team 2. Review team 3. Customer
Entry Criteria	Requirement Specification Document has been reviewed and authorized.
Exit Criteria	High-level design documents have been reviewed and authorized.
Input	Requirement Specification Document
Output	1. high-level design document 2. project standard 3. functional design document 4. database-design document 5. High-level design review record.
Measurement	1. high-level design effort 2. high-level design defects 3. review efforts 4. Rework effort.

- **Detailed Design**
  - During the detailed design phase, the view of the application developed during the high-level design is broken down into module and programs.

- Logical design is done for every program and then documented as program specification.

- Steps in the process is as follows
  - Break down functions into their components
  - Develop data migration program
  - Design/develop code skeleton(template)
  - Develop utilities/tools
  - Do program design
  - Plan unit testing

Title	Description
Participant	Design team.
Entry Criteria	High-level design documents are reviewed and authorized.
Exit Criteria	program specification and unit test plan have been reviewed and authorized
Input	High-level design document.
Output	1. program specification 2. Unit test plan.
Measurement	1. detailed design effort 2. design defect 3. unit test plan defects 4. program skeleton defects 5. Review and rework effort.

- **Build and Unit Testing**
  - The detailed design is used to produce the required programs in a programming language.

- This stage produce the source code, executable and database (if applicable) following the appropriate coding standard.

- Steps in the process is as follows
  - Generate test database
  - Generate code
  - Conduct independent unit test

Title	Description
Participant	members of the team and the team leader
Entry Criteria	Program specifications are reviewed and authorized.
Exit Criteria	All test cases in the unit test plan are successfully executed.
Input	1. physical database 2. design document 3. project standards 4. program specification 5. unit test plan 6. program skeletons 7. Any utilities and tools and their associated documents.
Output	1. test data 2. source code 3. Executable 4. code review report/review records 5. Independent unit test report/review records.
Measurement	1. capture the build and unit testing effort 2. code review defect 3. independent unit test defects 4. Review and rework effort.

- **Integration Planning and Testing**

- Integration plan must be specify the order in which the modules will be integrated.
  - Integration is performed in the order specified in the integration plan and corresponding test case for each integration phase is executed.
  - Integration plan describe the sequence of integration, overhead software, test environment and resource required.
- Steps in the process is as follows
  - Identify environmental needs
  - Determine integration process
  - Develop integration test plan

Title	Description
Participant	**Members of the integration team.**
Entry Criteria	**High-level design documents are reviewed and authorized.**
Exit Criteria	**Integration plan and integration test plan have been reviewed and authorized.**
Input	**1. high-level design document** **2. Programs.**
Output	**1. Integration plan.**

- **System Testing Planning and Testing**
- In this phase, we can validate the system against the requirement specification.

- Attributes such as external interfaces, performance, security, configuration sensitivity, coexistence, recovery and reliability are validated during this phase.
- System test planning can be occur before coding is completed.
- **Steps in the process is as follows**
  - Determine environmental needs
  - Determine system test procedure
  - Develop test cases

Title	Description
Participant	**Members of the system test team.**
Entry Criteria	**Requirements specification document and the high-level design documents are reviewed and authorized.**
Exit Criteria	**System test plan is reviewed and authorized.**
Input	**1.high-level design document** **2.Requirement specification document.**
Output	**1. system test plan** **2. Test result.**

- **Documentation**
  - This phase is generally performed towards the end of development effort.
  - It refers to operational manuals, user manuals and other documents needed by the customer.
  - It may be developed by development team or by a documentation team with the input from development team.
  - Steps in the process is as follows
    - Prepare user manuals

User manuals are mostly depends on the requirement specification

- Prepare operational manuals
- Prepare data conversion manuals
- Prepare online help
- Review documentation/manuals

- **Acceptance and Installation**
  - This phase has two basic task:
    - Getting software accepted
    - Installing the software at the customer site.
  - Acceptance consists of formal testing conducted by customer based on the acceptance test plan.
    - If the result of the analysis satisfies the acceptance criteria then user accept the software.
  - Steps in the process is as follows
    - Perform acceptance
    - Perform installation

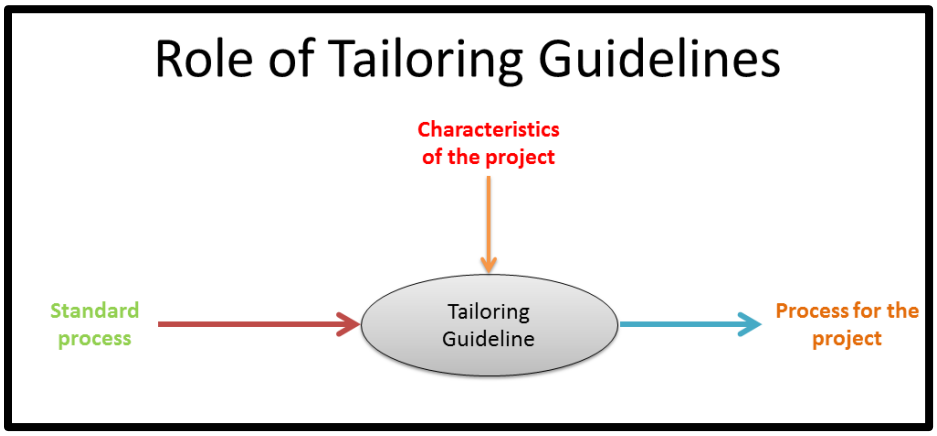
Title	Description
Participant	**Installation team, customer and project leader.**
Entry Criteria	**System test has been successfully completed.**
Exit Criteria	**Customer sings off on the acceptance letter.**
Input	**1. tested software** **2. Acceptance criteria.**
Output	**Installed software.**
Measurement	**Effort spent and the defect found.**

- **Warranty Support**
  - It is phase in which the installed application is supported until it stabilizes in the production environment.

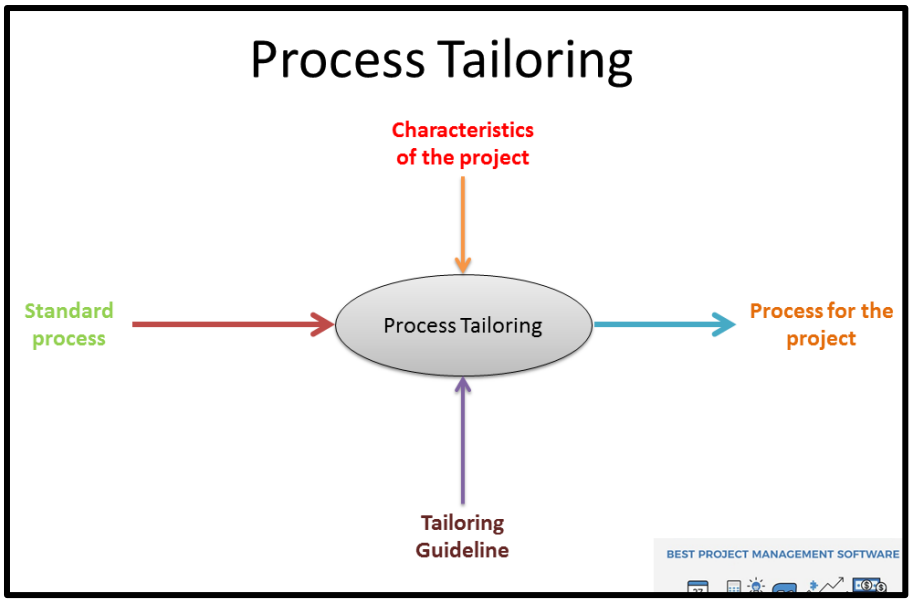
Title	Description
Participant	**installation team**
Entry Criteria	**Application goes live in production.**
Exit Criteria	**Warranty support period and the customer singed-off on the entire project.**
Input	**1. installed application 2. user documents 3. Software trouble reports.**
Output	**Sing-off documents from the customer and installed application.**

- **Process Tailoring**
  - Tailoring is the process of adjusting a previously defined process of an organization to obtain a process that is suitable for the particular business or technical needs of project.
  - We can consider tailoring as adding, deleting or modifying the activities of a process so that the resulting process is better suited to achieving the project's goals.
  - To allow effective reuse of previously defined processes , tailoring guidelines are provided.

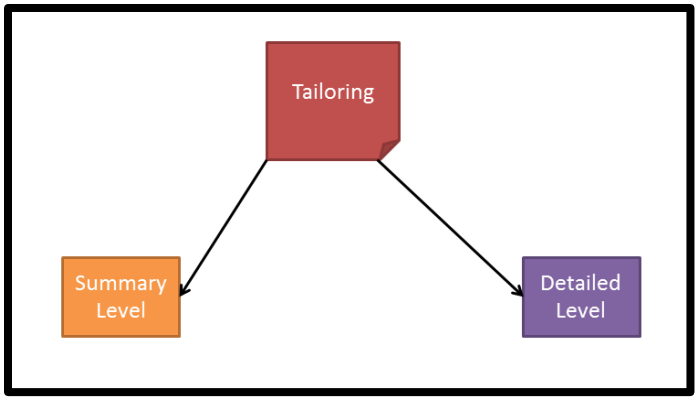
- These guidelines define the conditions and the types of changes that should be made to a standard process.
  - It defines a set of permitted deviations to the standard process in the hope that the optimal process can be defined for a project.
- Do code review adds a great deal of value in many cases, but sometimes its added value is not matching with the effort requires.
  - It would be done by either a group or by one person.
  - Standard process does not specify how code review should be performed.
- **Role of Tailoring Guidelines**



- **Process Tailoring**



- Tailoring guidelines can help a project manager by advising that the activity Do code review be performed only for certain types of programs and by suggesting the optimal form of the review.



- **Summary-Level Tailoring**
  - This is depending on the project characteristics, the project manager applies overall guidelines for tailoring the standard process.
    - It provide some general rules regarding certain types of detailed activities.

- The project manager first identifies certain characteristics of the project.

- For project development following characteristics are used for tailoring:
  - Experience and skill level of the team and the project manager.
    - **If the team members have more than two years of experience with the technology being deployed in the project**, then we can say that the experience level of the team is **high**, otherwise law
  - Peak team size.
  - Clarity of the requirements.
  - Project duration.
    - **If project must be delivered in less than three months**, then we can say that the duration of **project is short**.
  - Application criticality.
    - **If the effect of the application on customer's business is significant**, then we can say that the application criticality is **high**, otherwise law.
- Summary tailoring guidelines are generally
  - review-related
  - effort-related

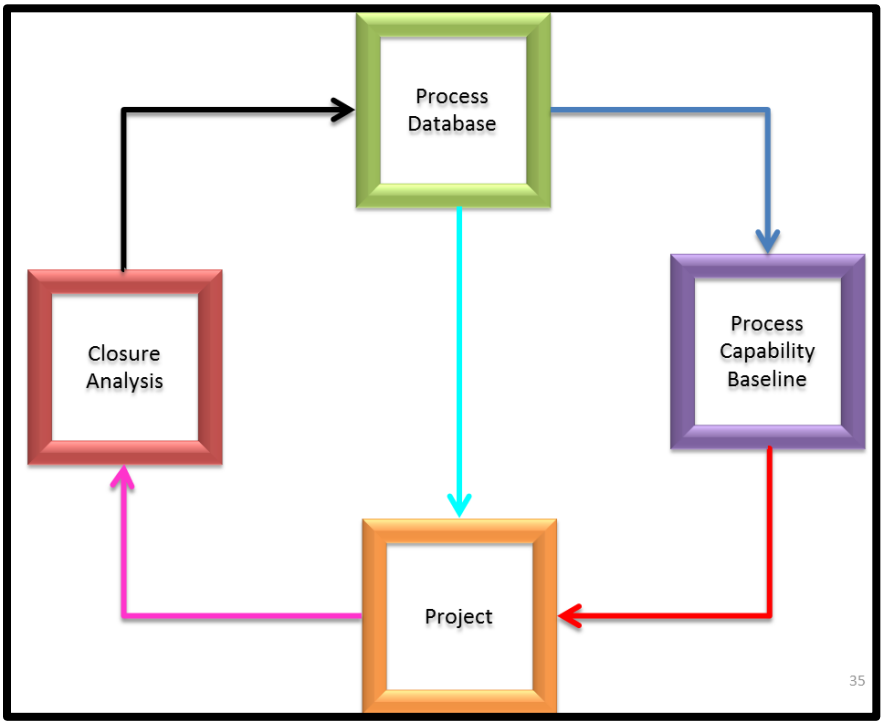
- schedule-related
- resources-related
- formality-related.

- **Detailed Tailoring**
  - It covers execution of activities, their review and documentation needs.
  - In tailoring guideline, if activity specify as optional then project manager can decide whether or not to execute activity.
  - In tailoring guideline, if documentation specify as optional then project manager can decide whether or not to project needs document.
  - For review we can perform either **do group review, do-one person review or Do not review**.

2.2 Software Metrics and Process Management

- There are two mechanisms for encapsulating past experience for use in project planning and management.
  - **Process database (PDB)**
  - **Process Capability Baseline (PCB)**
- PDB and PCB encapsulate the experience with previous projects ,in form of software metrics data and make it available to fresh project
- Information from PDB and PCB is used heavily during project planning.

- PCB employed to analyze the organization's overall process capability and its evolution over time.
- Main source of data for these both components is the completed project.
    - The data collected in the projects is analyzed and then reported in closure report, which forms the main source for the PDB.
    - The data in PDB are then used to compute PCB.



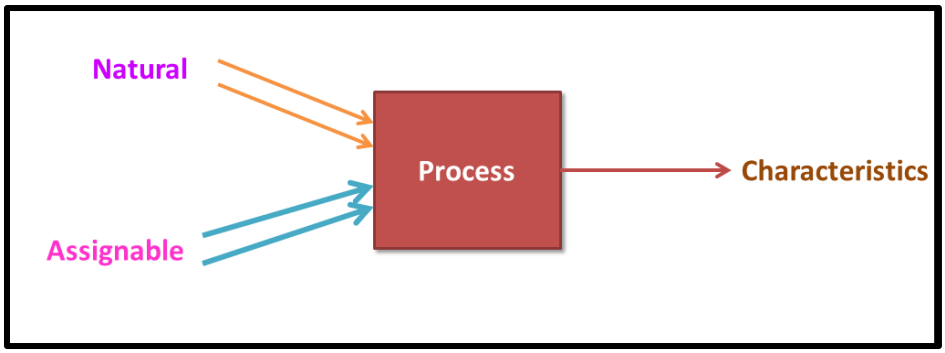
- A PDB is required at **CMM level3** in the **OPF** and **OPD KPAs**.
- At level 4 ,PDB is used for performing process analysis, computing the process capability and quantitatively controlling a project.

- At level 4 ,PCB is used for knowing the process capability in quantitative.
- PDB & PCB are focus on **software metrics.**
- **Software Metrics**
  - **Software metrics characterized as a following,**
    - **Software process (Process metrics)**
    - **Software product (Product metrics)**

Title	Process metrics	Product metrics
Measurement	Software process or development environment.	Software products.
Example	• Productivity • Quality • Resource metrics • Defect injection rate • Defect and removal efficiency	• Size • Reliability • Quality • Complexity of the code • functionality

- **Usage of Metrics data in Software organization**
  - Project Planning.
  - Controlling a project's process.
  - Analyzing and improving the organization's process.

- Statistical Process Control (SPC) is a technique that has been used with great success in manufacturing.
  - Process is basically used to achieve some output, and output can de define in terms of some quality *characteristics.*
  - The purpose of SPC is to control the process of production so as to *reduce the variability* in the value of output's quality characteristics in order to improve the quality over time.
  - The goal of SPC is to keep the production process in statistical control.
  - A number of factors affect the characteristics of process output.
    - This factor classified into two categories:
      - Natural or inherit causes for variability
      - Assignable causes for variability
    - A process said to be under ***statistical control*** if the variability in the quality characteristics is due to natural causes only.
    - For applying SPC, ***control charts*** are a favorite tool.



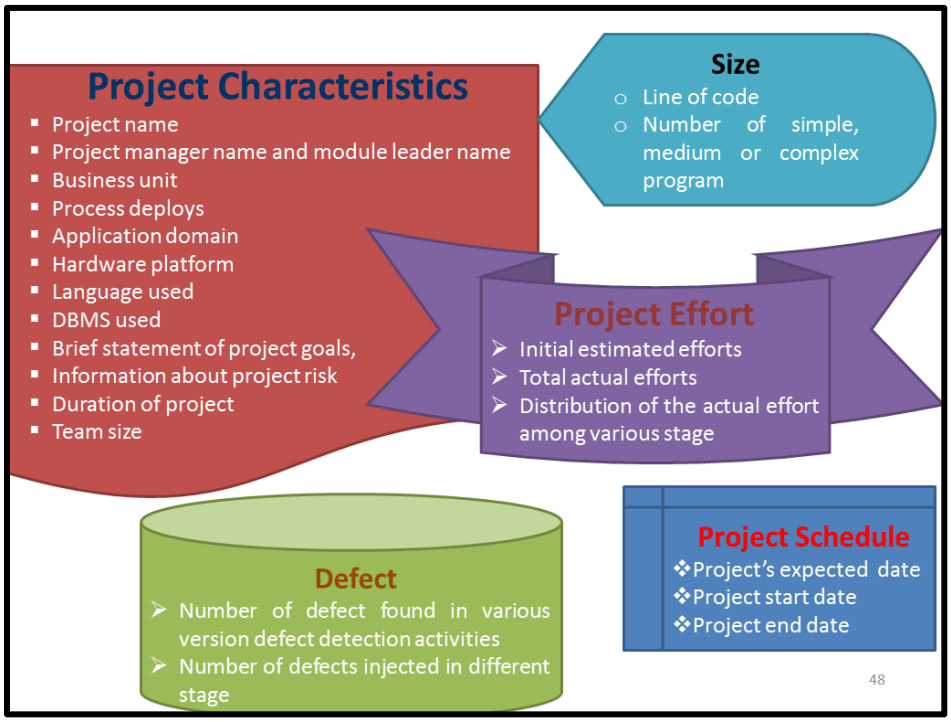
- The control charts are used to continuously monitor the performance of the process and identify an out-of-control situation as soon as it occurs.
  - Two type of actions are performed again out-of-control situation:
    - **Rework this output**, such that it has acceptable characteristics. That is **take corrective action**.
    - **Do further analysis** to identify the assignable causes and eliminate them from the process. That is ,**take preventive action**.
- For building a control chart, the output of a process is considered as a stream of numeric values, representing the values of the characteristics of interest for the outputs being produced by process.
- Subgroups of data are taken from this stream and the mean values for the subgroups are designed, giving the **X-bar chart**.
  - Lower control limit(**LCL**)
  - Upper control limit(**UCL**)

- If point fails outside the control limits, the large variability is considered to be due to assignable causes.
- Another R-Chart is design from the range of the chosen subgroups (difference between minimum value and maximum value).
- The capability of a process, which defines the range of expected outcomes when the process is under statistical control, is the range specified by the LCL & UCL.

2.3 Process Database

- Process database is a permanent repository of the process performance data from projects.
  - It can be used for project planning, estimation, analysis of productivity and quality and other purpose.
  - PDB consist data from completed project, with each project providing one data record.
- **Key purpose of the PDB**
  - To aid a new project in planning, particularly in the estimation of effort and defects.
  - To collect productivity and quality on different types of projects.
  - To aid in creating process capability baselines.
  - To facilitate data analysis for identifying areas and scope of improvement.

- **The data captured in the PDB at Infosys can be classified as follows:**
  - Project Characteristics
  - Project Schedule
  - Project effort
  - Size
  - Defect



- **Sample Entry**
  - Data for a project are stored in four major tables containing the *over all information, information about effort, information about defects, information about size*.
- Process Assets

- After termination of project some assets are collected from projects. like,
  - Project management plan
  - Configuration management plan
  - Schedule
  - Standards, checklists, guidelines, templates and other supports.
  - Developed tools and related notes
  - Training material
  - Other documents that could be reused by future projects.
- Body of Knowledge
  - At Infosys, besides the process database, another system called body of knowledge (BOK) is used to encapsulate experience.
  - Following key topics are included in Bok.
    - Human Resource Development
    - Computer and Communication Service
    - Requirement Specification
    - Builds
    - Tools
    - Methodologies/techniques
    - Education and research
    - Other facilities
    - Design

- Reviews/Inspection and testing
- Quality assurance and productivity
- Project management

2.4 Process Capability Baseline

- PDB contains data for each project, while the Process Capability Baseline (PCB) represents a snapshot of the capability of the process at some point in time in quantitative terms.
  - The capability of a stable process can be determined from past performance of the process.
  - If baseline is regularly established, developments in the process capability can easily be obtained.
  - The data captured in the PCB at Infosys can be specifies as follows:
    - Delivered quality
    - Productivity
    - Schedule
    - Effort distribution
    - Defect injection rate
    - In-Process defect removal efficiency
    - Cost of Quality
    - Defect distribution
- Baseline indicates the capability of a process, a separate baseline must be created for each process in the organization.

- **Baseline Creation**
  - Productivity is defined in terms of function points per person month.
  - Quality is defined as delivered defects per function point.
    - Where the number of defects delivered is approximated by the number of defect found through acceptance testing, during installation and during the warranty period.
  - Defect Injection rate is the total number of defect injected during the life cycle of the project normalized with the size
    - Total number of defects in a project is the sum of the number of defect found by different defect detection stage in the process and the number of defect founds after delivery.
  - Overall defect removal efficiency is the percentage of total defects that are detected inside the process by the various defect detection activities.

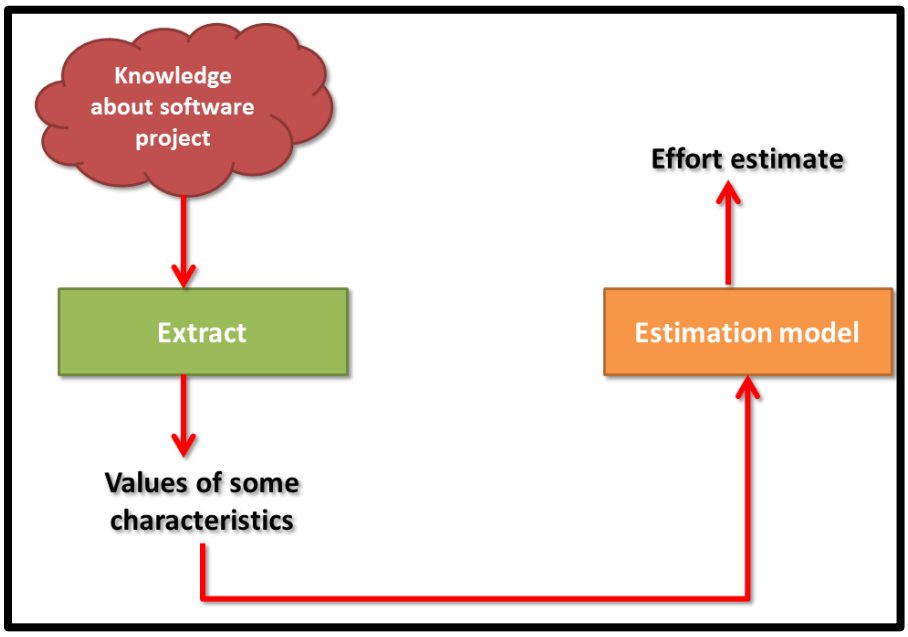
F = Size of the software in function points
E = Total effort spent in the project
D_1 = Total number of defect found before delivery
D_2 =Total number of defects found after delivery.
$D = D_1 + D_2$
Productivity =E/F
Quality = D_2/F

Defect Injection rate = D/F

Overall defect removal efficiency = D_1/D

2.5 Types of Effort Estimation

- Without an estimate of how much effort and time will be needed to execute project, effective project planning and management are not possible.
- **Effort Estimation Models**
  - Basic activity of estimation is to get input in the form of the values of some characteristics of the software under development, and using these input values, estimate the effort for the project.
  - Software estimation model defines precisely which values it needs and how these values are used to compute the effort.



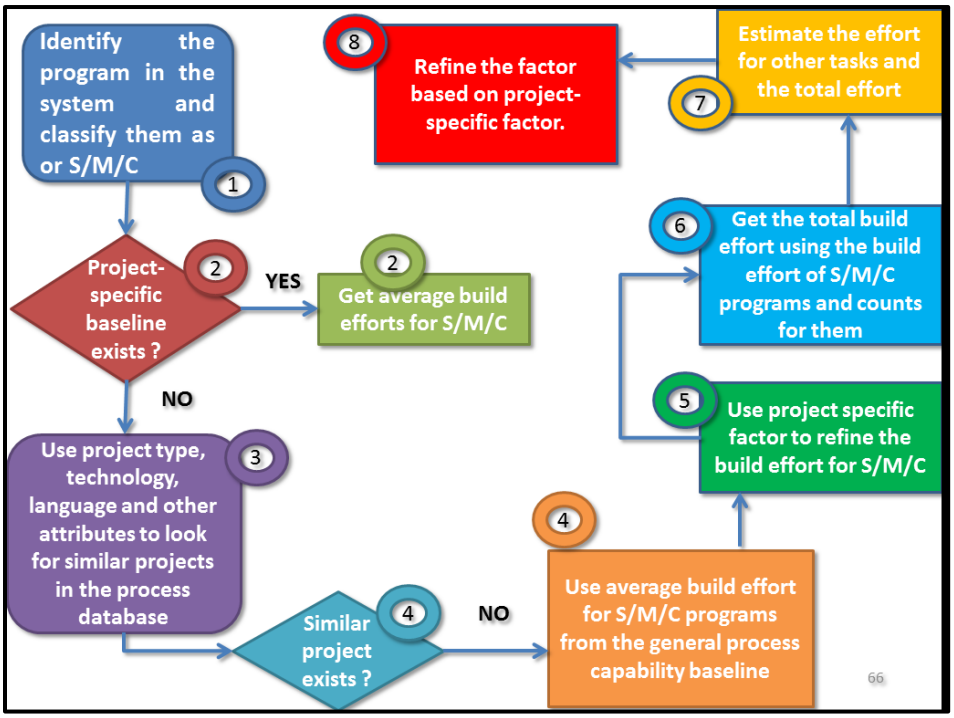
  - At Infosys, estimation taken place after analysis.
    - The Bottom-up Estimation Approach

- The Top–Down Estimation Approach
- The Use Cases Point Approach

- **The Bottom-up Estimation Approach**
  - This approach is preferred and recommended at Infosys.
  - Company employs a **task unit approach.**
  - **Task Unit Approach**
    - Project manager first divides the software under development into major programs (unit).
    - Each unit classified as simple, medium or complex based on certain criteria.
      - For each unit the project manager defines a standard effort for coding and self-testing.
    - The procedure for estimation can be summarized as the following sequence of steps:
      1. Identify the program in the system and classify them as simple, medium, or complex (S/M/C).
      2. If project specify baseline exists, get the average build effort for S/M/C programs from the baseline.
      3. If project specific baseline does not exists, use project type, technology, language and

other attributes to look for similar projects in the process database.

4. If no similar project exists in the process database and no project-specific baseline exists, use the average build effort for S/M/C programs from the general process capability baseline.
5. Use project-specific factors to refine the build effort for S/MC programs.
6. Get the total build effort using the build effort of S/M/C programs and the counts for them.
7. Using the effort distribution given in the capability baseline or for similar projects given in the process database, estimate the effort for other tasks and the total effort.
8. Refine the estimates based on project-specific factor.



- **Classification Criteria**

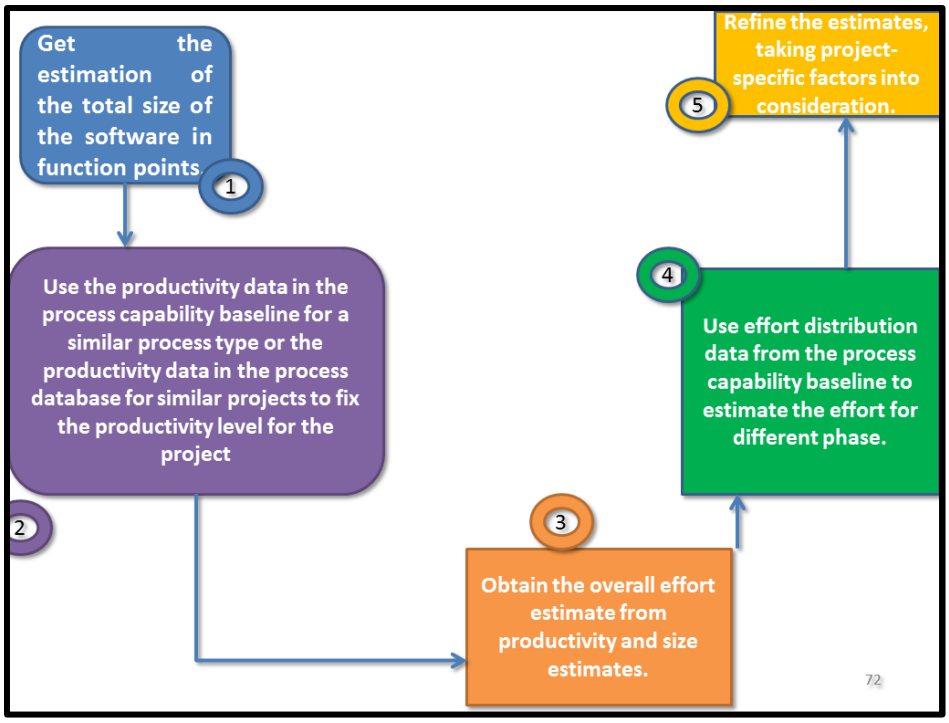
Language/DBMS/Platform	Unit	Simple(S)	Medium (M)	Complex(C)
Ingres ABF on UNIX	Program	Less than 500 LOC	500-1000 LOC	Other case
Cobol, CICS with IMS for Y2K	Program	Less than 700 LOC;JCLs	700-1000 LOC	Other cases
C on Oracle/UNIX	Program	Less than 100 LOC or up to 5	100-500 LOC or 5-20 SQL	Other cases

		SQL statements	statement	
GUI development environment	Screen or form based user interface programs	List display or search criteria screens with not more than two to three criteria	Involved display-data logic having interfaces with two to three screens or functions	General purpose screen called from multiple places for maintaining multiple tables
Cobol/Oracle; Cobol/DMS on mainframe	Report or related programs	Less than 16 fields to print; less than 5 tables to refer; up to 1 summary field to print	16-35 fields to print; 5-8 tables to refer; 2-4 summary fields to print	More than 35 fields to print; more than 8 tables to which to refer,; more than 4 summary fields to print

C++	Class or its member functions	No complex user interaction	Use of data structures such as queues and lists	Multithreading libraries

- **The Top-Down Estimation Approach**
  - This approach is frequently used for conversion or reengineering projects in which the size of the existing system to be converted is known quite precisely.
  - At Infosys, the approach starts with an estimate of the size of the software in function points.
    - It can be counted using standard function point counting rules.
    - Alternatively, if the size estimate is known in terms of LOC, it can be converted into function points.
    - Size estimation requires an estimate of productivity.
    - Productivity is then used to calculate the overall effort estimate.

- The procedure for estimation can be summarized as the following sequence of steps:
    1. Get the estimate of the total size of the software in function points.
    2. Using the productivity data from the project-specific capability baseline, from the general process capability baseline ,or from similar projects, fix the productivity level for project.
    3. Obtain the overall effort estimate from the productivity and size estimates.
    4. Use effort distribution data from the process capability baseline or similar projects to estimate the effort for the various phase.
    5. Refine the estimates, taking project-specific factors into consideration.

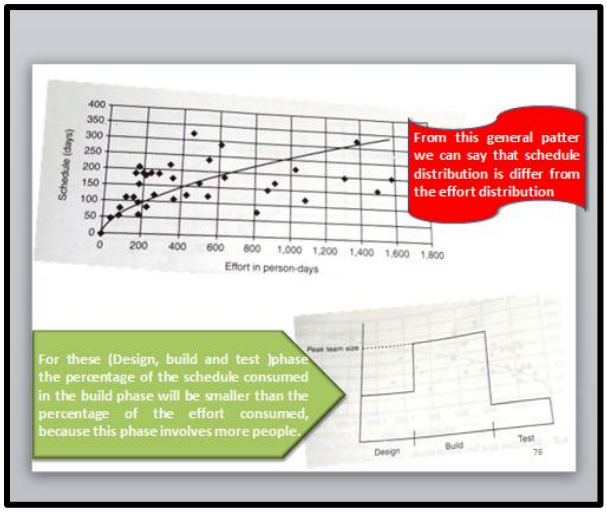


2.6 Scheduling

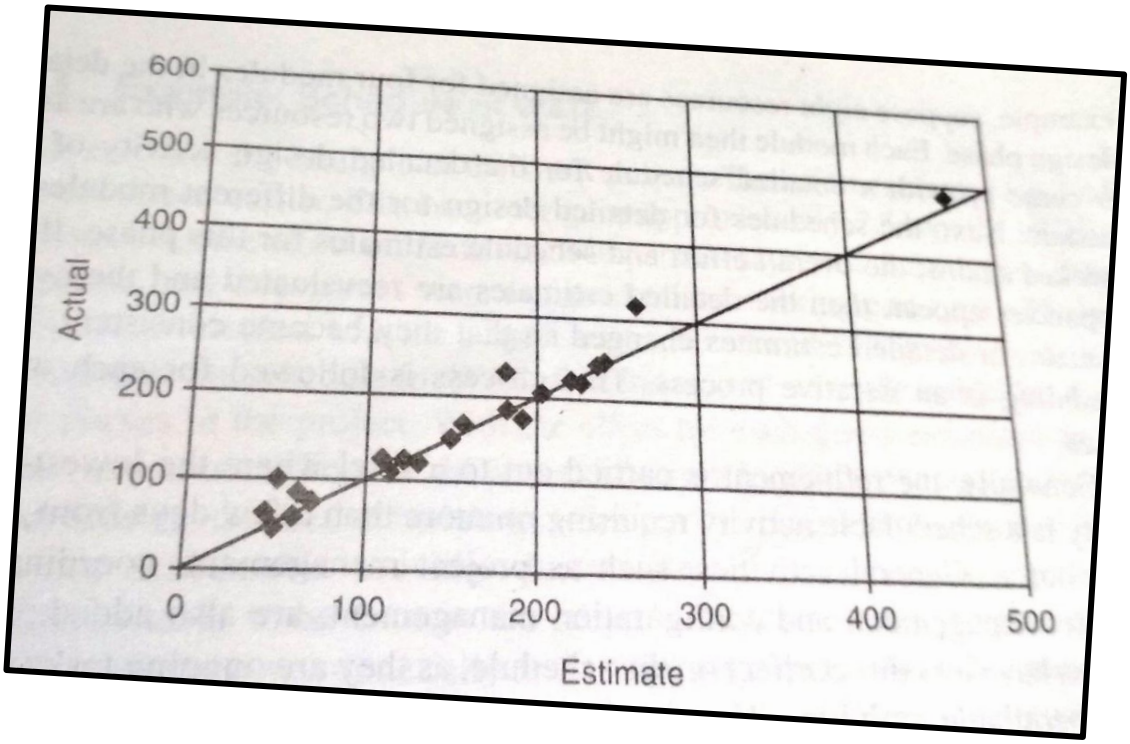
- Scheduling activity can be broken into two sub activities:
  - Determining overall schedule with major milestone (Over Scheduling)
  - Deciding the detailed schedule of the various task.
- **Overall Scheduling**
  - Once the effort is known or fixed , various scheduling are possible, depending on the number of resources put on the project.
  - Overall schedule for the project is often driven more by the business requirements of the customer than by any other factor.
  - If the schedule is set based on the effort estimates of the project, one method for determining it is use some

functions to derive the schedule from the estimated efforts.

- Once the overall duration of the project is fixed, then the schedule fort the major milestones must be decide.
  - To determine the milestone we must first understand the manpower ramp-up that usually take place in a project.
  - The number of people in a software project tends to follow the Rayleigh curve.
    - That is in the beginning and the end few people are needed in the initial phase of requirement analysis and design.
    - Human resource requirement during coding, unit testing, system testing and integration is fewer.
    - All this human resource requirement allocation is based on **peak time size(PTS)**.



- One way of checking the schedule estimates is to plot the actual schedule against the estimated schedule and see how close the points are to the 45-degree line.
  - If they all fall very close to the 45-degree line, then the scheduling approach can be considered to be effective.



- **Detailed Scheduling**
  - Once the high-level schedule is fixed, then
    - The overall project duration is known
    - Major milestones are known
    - Effort required for various phase in known
    - Resource required for various phase in known

- It means like that, once the effort is estimated and a reasonable schedule is fixed, then the value of the remaining variable is fixed
- In setting the detailed schedule, the tasks in the overall schedule are broken down into small schedulable activities in a hierarchical manner.
- For detailed scheduling, features of Microsoft Project (MSP) can be used.
    - For each and every activity listed in MSP. (**Effort Duration, start date, end date ,resources ,activity code, program code, module code, Dependency**)
- After entering all these details ***"resource leveling" or "resource loading analysis"*** of MSP may be used.
- Finally the schedule includes all schedule activities.