SOFTWARE PROJECT MANAGEMENT

DEEPAK KUMAR PARIDA

ASST. PROFESSOR

DEPT. OF COMPUTER SCIENCE & ENGINEERING

SRINIX COLLEGE OF ENGINEERING, BALASORE

Mobile: +91 9090190434, +91 9853993398

E-mail:dpkkmrprd3@gmail.com

Module-I

Software:

- A set of programs which is designed for a specific operation is called software.
- There two types of software:
  1. System software: OS, Device drivers
  2. Application software: MS-Word, Excel, VLC media player, MX player etc.

Project:

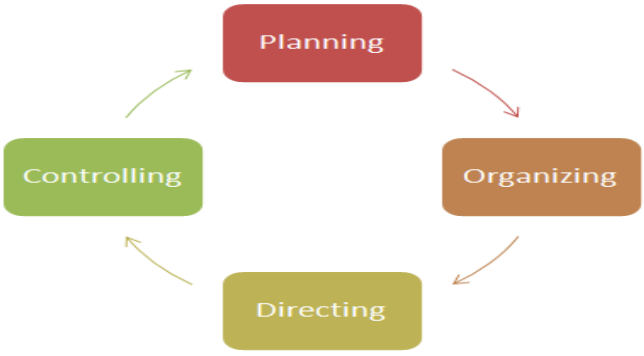
- A project is well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery).
- A Project can be characterized as:
  1. Every project may has a unique and distinct goal.
  2. Project is not routine activity or day-to-day operations.
  3. Project comes with a start time and end time.
  4. Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
  5. Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

Software Project:

- A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

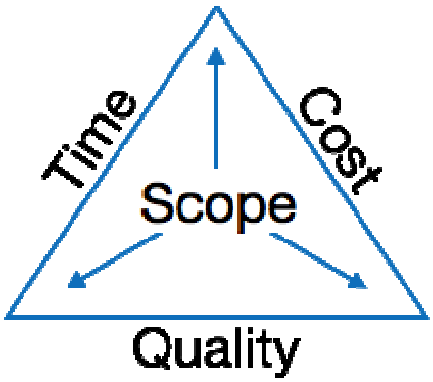
Management:

- The process of dealing with or controlling things or people is called management.
- Management is a set of principles relating to the functions of planning, organizing, directing and controlling a process or job to complete successfully within the time period.



Software Project Management:

- Software project management is an art and discipline of planning and supervising software projects.
- It is a sub-discipline of software project management in which software projects planned, implemented, monitored and controlled.
- It is a procedure of managing, allocating and timing resources to develop computer software that fulfills requirements.
- There are three needs for software project management. These are:
  1. Time
  2. Cost
  3. Quality



- The image above shows triple constraints for software projects. It is an essential part of the software organization to deliver a quality product, keeping the cost within the client's budget and deliver the project as per schedule.
- There are various factors, both external and internal, which may impact this triple factor. Any of three-factor can severely affects the other two.

Product:

- A product is something that is deliverable to an internal or external customer.

Product Life Cycle:

- We demonstrate this with the help of a model which we will call the **Big League Model**.
- The products that succeed in the market can be compared to the players who make it to the big league.
- Not every player can make it to the Big league. Based on his performance he will pass through various tougher and more stringent filters and if he is successful, graduate to the Big League.

Big League Model:

- It is the transformation of ideas to products:
  1. There are always more aspirants than those who make it to the Big League.

2. Aspirants have to pass through several filters before they can make it to the Big League.There are multiple filters at different levels that act as blockades to transformation of ideas into revenue generating products or services. Only those ideas that pass through these very demanding filters get converted into worthwhile products.
3. Some of the filters are basic competence, whether the idea conforms to organization strategy, relative merits and market needs.
4. Even among the aspirants who make it to the Big League (i.e. index transformed to products), not all are equally successful. The relative performance, team performance and other factors determine the future of a player.

Process of Applying the Big Language Model in S/W Product Development:

- Index comes from various sources- customers, prospects, marketing, engineering, manufacturing, suppliers etc.
- These ideas are evaluated for:
  i. **Feasibility** (Can we do it?)
  ii. **Consistency** (Which of these don't contradict each other and tie into the organizational vision?)
  iii. **Viability** (Can we do this profitably?)

Feasibility:

- It is established through studies, experiments and prototypes.

Consistency:

- Ideas that pass the feasibility filter are evaluated for consistency.
- The idea must fit in with or contribute to the organizational goals or the technology directions for it to be taken up for further consideration.

Viability:

- An idea must also satisfy the viability filter for it to go to the next level.
- Viability refers to being successful in the market place.
- For a product or service to be viable (and profitable), some of the questions that need to be answered include:
  - What would be the **unique selling proposition** (USP) of my offering over that of the competitors?

Attributes of project:

1. A project is a set of things that an organization decides to do achieve its vision and goals as well as to enhance revenues and profits. In other words, each project serves a definite purpose that fits in with the big picture of the organization.
2. Each project has a definite beginning and a definite ending time.
3. Each project has to be executed under certain constraints. These constraints may change with time and that is a part of the project dynamics.
   e.g. Availability of manpower, capital budget, drop dead dates for project completion etc.

A product development life-cycle:

- It consists of following phases:
  1. Idea generation
  2. Initial prototype development
  3. Alpha testing
  4. Beta testing
  5. Production
  6. Maintenance and Obsolescence

1. Idea Generation:

- **Input from customers which contain their multiple requirements:**
  - S/w products come out in new and upgraded versions.
  - Customers may say what they linked about the previous and these can translate into ideas for new versions.
- **Inputs from supplies that may open up new opportunities to introduce new features or exploit technology better:**
  - For example , if a new hardware that has ten times more processing power or memory comes into the market , a piece of s/w like a database can be optimised to exploit these additional resources.
- **Inputs from employees:**
  - This is an important source and if harnessed carefully can have immense benefits.
  - Valuing employees ideas and considering them for further development, if found promising, could have a significant positive impact on increasing the creativity moral of employees.
  - This is especially valid if their ideas finally end up in the market in the form of quality products.
  - Respecting such employee inputs can also result in loyalty to the organisation and enhance employee organisation bonding.
- **Competitive information & marketplace demand:**
  - Sometimes one has to consider new ideas to counter market place advance by competition.
  - The time lag for the world to know the marketplace trends is drastically reduced. This puts an additional pressure on time to market consideration
  - The type of projects that the idea generation phase produces are:
    1. Study the requirements of customer xyz and come up with a list of features they want.
    2. What new enhancements can be made to feature ABC in our product?
    3. What kinds of bugs were filled against our product in the previous version?
    4. “Study the competition and find out Why we lost those sales”

2. Initial Prototype Development:

- The ideas that are found promising go into the prototyping phase.
- Prototyping means building a simplistic model of the final product and putting together a demo.
- The demo address issues such as user interface, look and feel and work flow.
- This prototype is generally shown to the senior, management or select customers to elicit feedback on the general look-and-feel and the functionality to be supported by the product.
- The result of this evaluation can be a "Go" or "No Go" decision.
- If it is a "Go" decision, the product proceeds to the next step; if it is a "No-Go" decision, it could be either categorical or conditional.
- Some of the projects that are generated during the prototyping phase are:
  a. User interface specification & design.
  b. Target Market positioning.
  c. Work flow specification.

3. Alpha Testing:

- The purpose of the Alpha phase is to move from a skeleton prototype to a somewhat usable product.
- Before a product can be put to use in a real life system, a basic trial needs to be done .
- Unless the product passes this trial, even in its infancy, it is unlikely that it will be fit enough for use in real life.
- Alpha testing is carried out in a lab environment and usually, the testers are internal employees of the organization.
- To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing.
- Some of the projects that originate from the Alpha phase are:
  - Migrate data from the existing system.
  - Get the internal application xyz on the new product.
  - Formulate some of the standards to be followed during the product development.

4. Beta Testing:

- The purpose of the Beta phase is to iron out the kinks in the product and to start adding the necessary supporting infrastructure needed to roll out the product (e.g. Documentation, install ability, manufacturability etc).
- Based on the Feedback from the Alpha phase, more functionality is added and the existing functionality is refined.
- The external customers work as trusted partners, testing the untried functionality of the product, being fully aware that the product may have problems.
- The products teams within the development organisation in turn organize themselves into project teams.

- The choice of the Beta customers determines the success of the eventual product.
- Some of the project originate from the Beta phase are :
  - Getting documentation ready.
  - Planning the install ability of the product.
  - Setting of supports teams and processer to attend to customer's problems.
  - Formations of special teams dedicate to attend to specific customers.

5. Production Phase:

- After the Beta phase (and the development that follows) the product is ready for prime time (or the big League) .
- This is also called the production phase.
- The type of projects the production phase produces are:

a. process Teams:

- The process-oriented projects gain momentum during the production phase.
  i.e. be a standard team , a configuration control team and a tooling team.

b. Training :

- Users have to be trained on the new products.
- The need for training becomes more critical, especially if it is a breakthrough or a revolutionary product.

c. Documentation:

- The Documentation just like the product-matures and this is usually a significant project.
- This project will encompass decision such as the formal of the documentation, choice of the distribution etc.

d. Testing:

- Before the product goes into production, it is very important to test it exhaustively.
- Testing projects covers taking lessons from the alpha and Beta phases and putting them into test suits.

e. Supportability:

- This generates a turn of projects, engineering the code to be more diagnosable and putting in the support infrastructures becomes very critical at this juncture.

f. Marketing programs:

- There would be a set of projects requiring the product designers & the marketing teams to work in unison in order to put together marketing programs.

6. Maintenance and Obsolescence Phase:

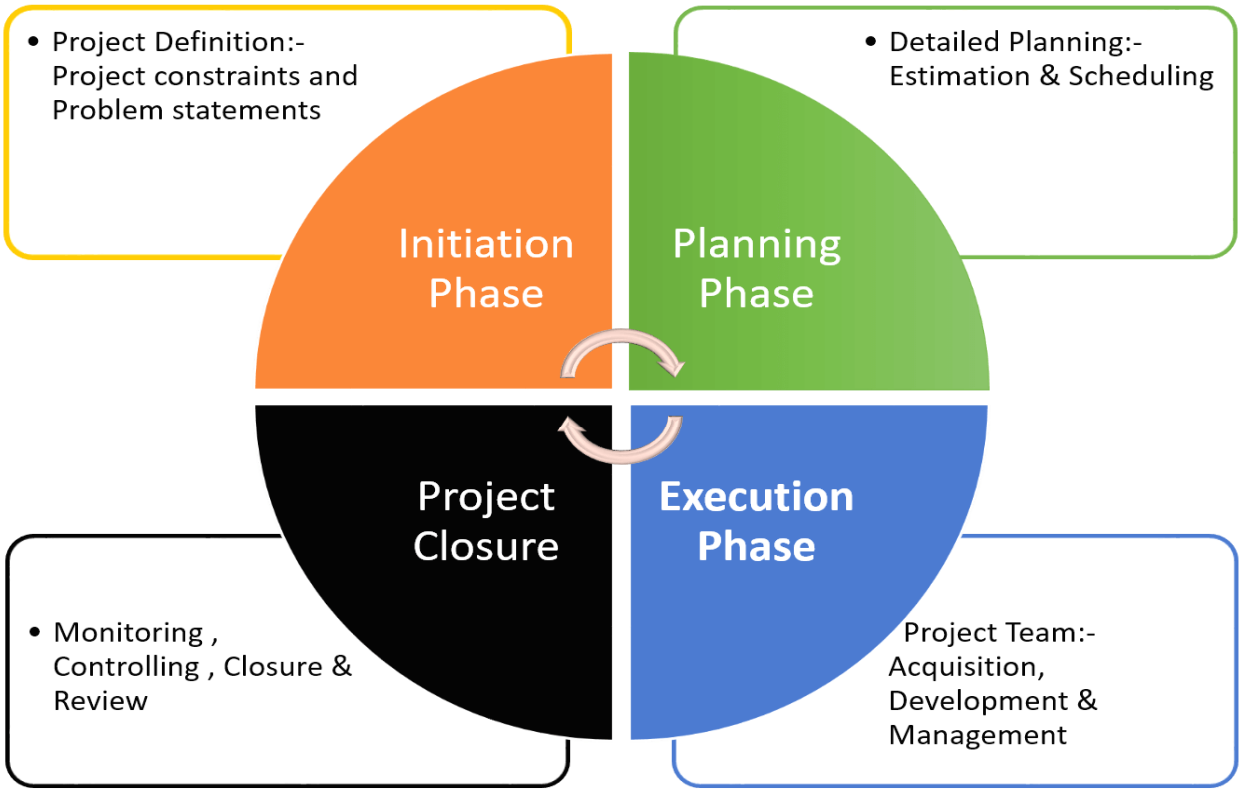
- Normally the product remains in the production phase for a certain length of time, during which it undergoes periodic revisions and up gradation.
- After that the product enters into the maintenance mode.
- During the maintenance phase , some of the issues that arise are:
  - How do you classify on incoming problem as a bug or a request for a new features.

- How do you priorities the incoming problem?
- How do you balance the workload on development of future versions with the maintenance activity?

- The maintenance phase activities are usually bud fixes and therefore each bug fix itself can be considered a separate project.

Project Life Cycle:

- The Project Life Cycle is a series of activities which are essential for accomplishing project objectives or targets.
- Projects may have different dimensions and difficulty level, but, whatever the size: large or small, may be all projects could be mapped to the given lifecycle structure.
- This life cycle for the project includes four phases:
    1. Initiation Phase
    2. Planning Phase
    3. Execution Phase
    4. Monitoring, Controlling & Closing Phase

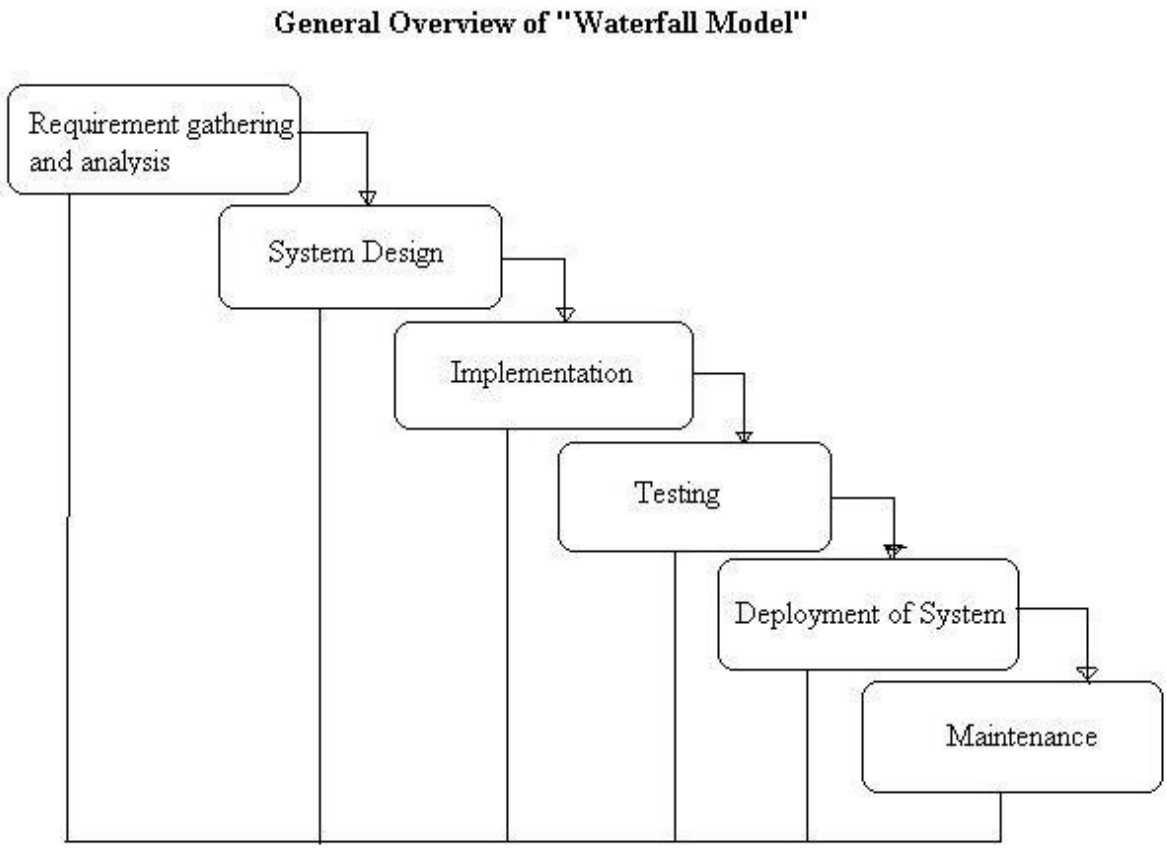


(Project Life Cycle Diagram)

Project Lifecycle Models:

Waterfall Model:

- Waterfall Model is a sequential model that divides software development into different phases.
- Each phase is designed for performing specific activity during SDLC phase.
- It was introduced in 1970 by **Winston Royce**.



Different Phases of Waterfall Model in Software Engineering:

1. **Requirement Gathering stage:**

- During this phase, detailed requirements of the software system to be developed are gathered from client

2. **Design Stage:**

- Plan the programming language, for Example Java, PHP, .net
- Or database like Oracle, MySQL, etc.
- Or other high-level technical details of the project

3. **Built/Implementation Stage:**

- After design stage, it is built stage that is nothing but coding the software.
- Here numbers of small programs are designed and go for their testing is called unit testing.

4. **Test Stage/Integration Testing:**

- In this phase, you test the software to verify that it is built as per the specifications given by the client.

5. **Deployment stage:**

- Deploy the application in the respective environment i.e. Customer environment or market.

6. **Maintenance stage:**

- Once your system is ready to use, you may later require change the code as per customer request.

When to use SDLC Waterfall Model?

- Requirements are not changing frequently.
- Application is not complicated and big.
- Project is short.
- Requirement is clear.
- Environment is stable.
- Technology and tools used are not dynamic and is stable.
- Resources are available and trained.

Advantages of the Waterfall Model:

- Before the next phase of development, each phase must be completed.
- Suited for smaller projects where requirements are well defined.
- They should perform quality assurance test (Verification and Validation) before completing each stage.
- Elaborate documentation is done at every phase of the software's development cycle.
- Project is completely dependent on project team with minimum client intervention.
- Any change in software is made during the process of the development.

Disadvantages of the Waterfall Model:

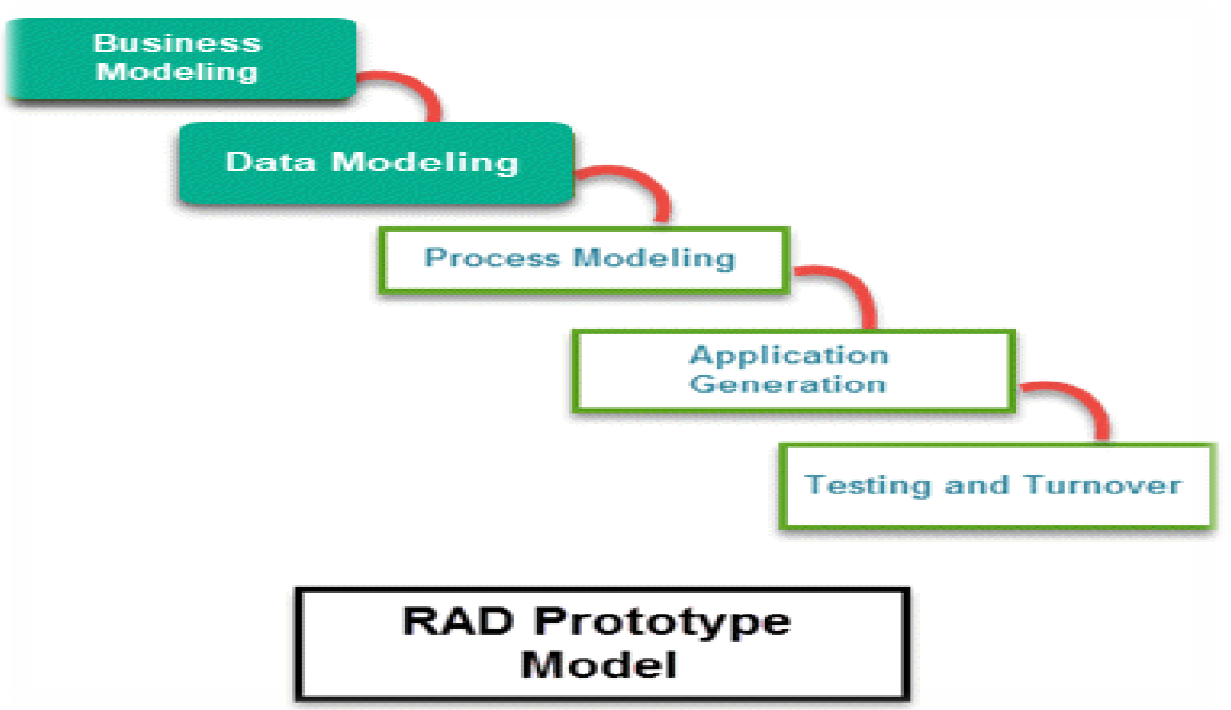
- Error can be fixed only during the phase.
- It is not desirable for complex project where requirement changes frequently.
- Testing period comes quite late in the developmental process.
- Documentation occupies a lot of time of developers and testers.
- Client's valuable feedback cannot be included with ongoing development phase.
- Small changes or errors that arise in the completed software may cause a lot of problems.

❖ **Some names of successful projects using the waterfall model:**

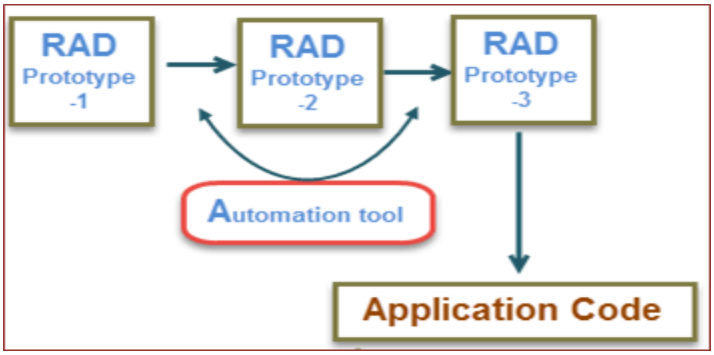
1. Energy management systems for electrical utilities
2. Rail traffic control systems for railways.

RAD (Rapid Application Development) Model:

- RAD or Rapid Application Development process is an adoption of the waterfall model; it targets at developing software in a short span of time. RAD follow the iterative
- SDLC RAD model has following phases:
    1. Business Modeling
    2. Data Modeling
    3. Process Modeling
    4. Application Generation
    5. Testing and Turnover



- It focuses on input-output source and destination of the information. It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects. The main features of RAD model are that it focuses on the reuse of templates, tools, processes, and code.



✷ **The phases in the rapid application development (RAD) model are:**

1. **Business modeling:** On basis of the flow of information and distribution between various business channels, the product is designed.
2. **Data modeling:** The information collected from business modeling is refined into a set of data objects that are significant for the business.
3. **Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.
4. **Application generation:** Automated tools are used for the construction of the software, to convert process and data models into prototypes.
5. **Testing and turnover:** As prototypes are individually tested during every iteration, the overall testing time is reduced in RAD.

When to use RAD Methodology?

- When a system needs to be produced in a short span of time (2-3 months).
- When the requirements are known.
- When the user will be involved all through the life cycle.
- When technical risk is less.
- When there is a necessity to create a system that can be modularized in 2-3 months of time.
- When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation.

Advantages of the RAD model:

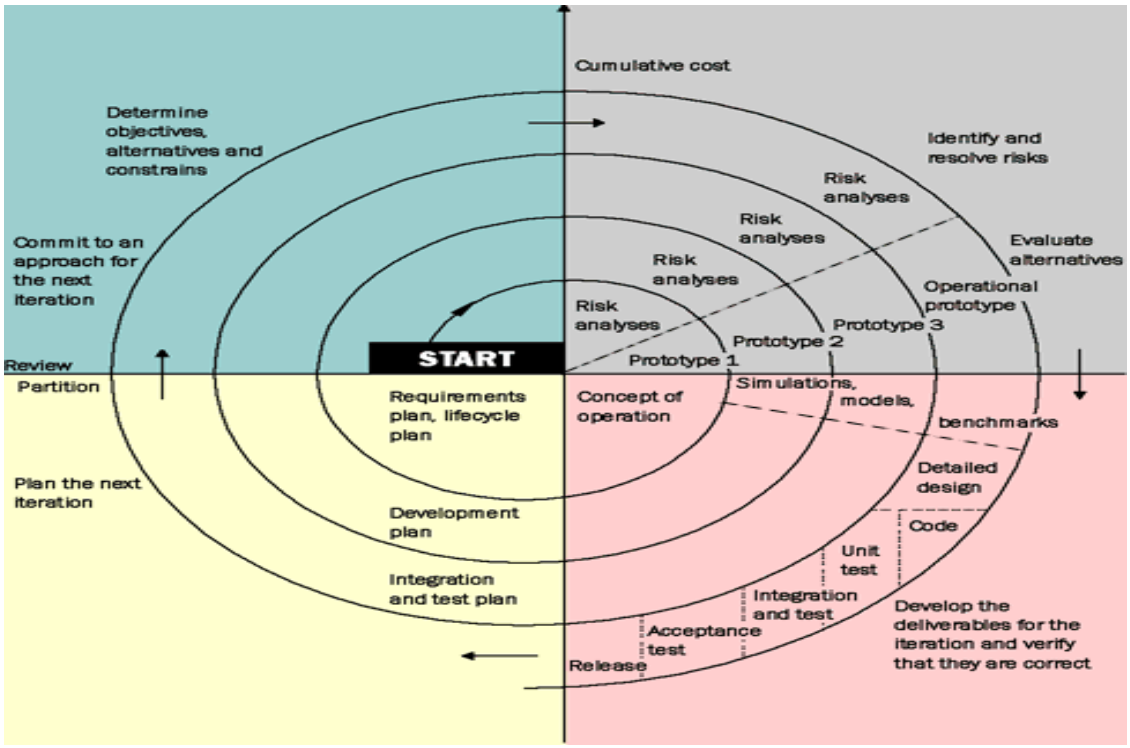
- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

Disadvantages of RAD model:

- Depends on strong team and individual performances for identifying business requirements.
- Only system that can be modularized can be built using RAD
- Requires highly skilled developers/designers.
- High dependency on modeling skills
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

Spiral Model:

- Spiral Model is a combination of a waterfall model and iterative model.
- Each phase in spiral model begins with a design goal and ends with the client reviewing the progress.
- The spiral model was first mentioned by **Barry Boehm** in his **1986** paper.
- The development team in Spiral-SDLC model starts with a small set of requirement and goes through each development phase for those set of requirements.
- The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.



- **Each cycle in the spiral is divided into four parts:**

1. **Objective Setting(First quadrant):**
   - Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists.
2. **Risk Assessment And Reduction(Second quadrant):**
   - The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.
3. **Development and validation(Third quadrant):**
   - The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

4. **Planning and Review(Fourth quadrant):**
   - Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

- The development phase depends on the remaining risks. For example, if performance or user-interface risks are treated more essential than the program development risks, the next phase may be an evolutionary development that includes developing a more detailed prototype for solving the risks.
- The risk-driven feature of the spiral model allows it to accommodate any mixture of a specification-oriented, prototype-oriented, simulation-oriented, or another type of approach.
- An essential element of the model is that each period of the spiral is completed by a review that includes all the products developed during that cycle, including plans for the next cycle.
- The spiral model works for development as well as enhancement projects.

When to use Spiral Model?

- When deliverance is required to be frequent.
- When the project is large
- When requirements are unclear and complex
- When risk and costs evaluation is important.
- When changes may require at any time
- Large and high budget projects

Advantages:

- Additional functionality or changes can be done at a later stage.
- Cost estimation becomes easy as the prototype building is done in small fragments.
- Continuous or repeated development helps in risk management.
- Development is fast and features are added in a systematic way.
- There is always a space for customer feedback.

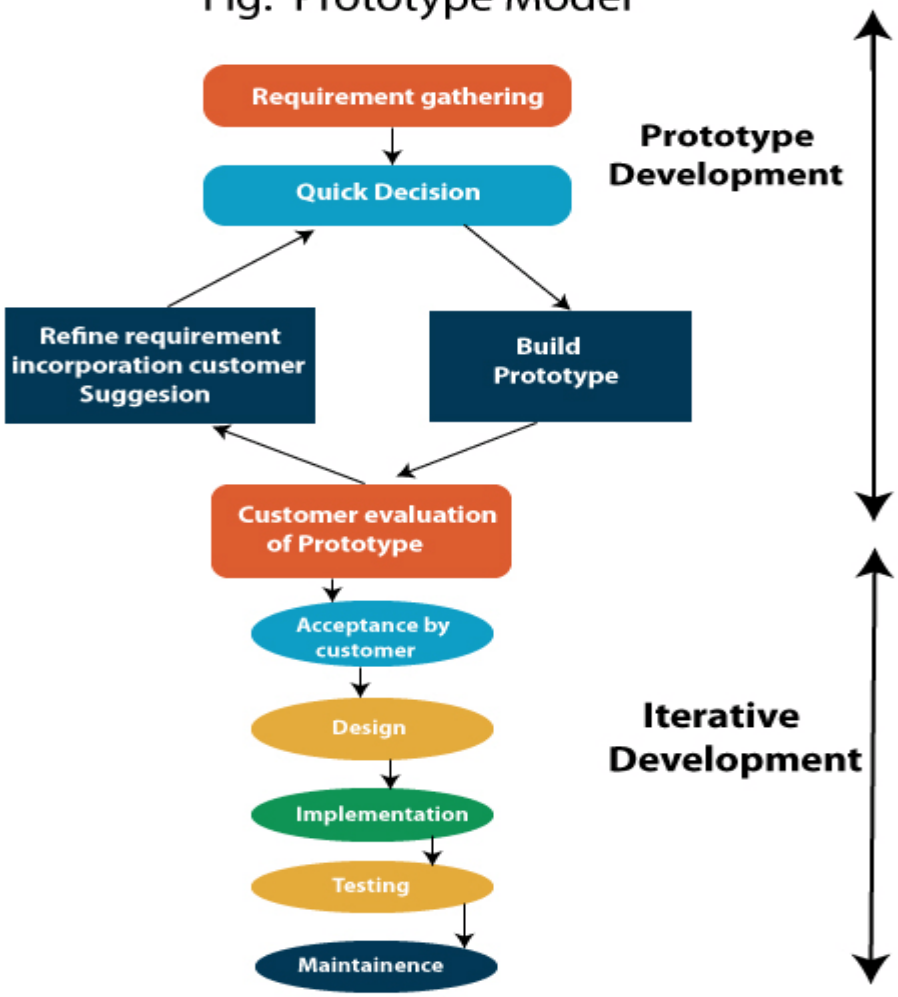
Disadvantages:

- Can be a costly model to use.
- Risk analysis needed highly particular expertise
- For its smooth operation spiral model protocol needs to be followed strictly.
- Doesn't work well for smaller projects.

Prototype Model:

- It is a popular life cycle model.
- It is the extension of the waterfall model.
- It is a toy model of a project before development of the actual S/W.
- It can be build very quickly by using several short cuts.

Fig: Prototype Model



- **Steps of Prototype Model:**
  1. Requirement Gathering and Analyst
  2. Quick Decision
  3. Build a Prototype
  4. Assessment or User Evaluation
  5. Prototype Refinement
  6. Engineer Product

Advantage of Prototype Model:

- Reduce the risk of incorrect user requirement
- Good where requirement are changing/uncommitted

- Regular visible process aids management
- Support early product marketing
- Reduce Maintenance cost.
- Errors can be detected much earlier as the system is made side by side.

Disadvantage of Prototype Model:

- An unstable/badly implemented prototype often becomes the final product.
- Require extensive customer collaboration.
  - Costs customer money
  - Needs committed customer
  - Difficult to finish if customer withdraw
  - May be too customer specific, no broad market
- Difficult to know how long the project will last.
- Easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation, and feedback.
- Prototyping tools are expensive.
- Special tools & techniques are required to build a prototype.
- It is a time-consuming process.

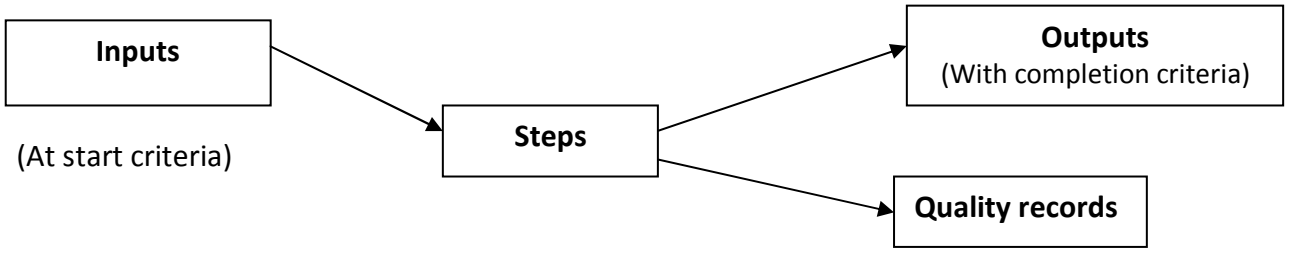
Process Models:

Product:

- A product is something that is deliverable to an internal or external customer.

Process:

- A process is a documented and followed way that constitutes one or more steps in producing a product.
- **Characteristics of a process:**



1. **Each process has a well defined set of inputs that it operates on:**
   - The purpose of a process is to produce a product.
   - For this, it needs a set of inputs.
   - For example: For the coding process, the design document constitutes an input.
   - An input is usually an output from another process.
   - Some of the inputs are mandatory to a process (i.e. the process can't proceed without such inputs) and some are optional inputs.

2. **Each process produces a set of outputs that are used as inputs to other processes:**
   - A process transforms its inputs to pre defined outputs.
   - Ex: The design process takes the requirement specifications as input and produces design document as output.
   - This design document in turn becomes an input to the coding process.
3. **Each process has a set of steps that are to be followed to produce outputs from the inputs:**
   - The steps document what need to be done by whom and in what sequence to transform the inputs to outputs
   - One can view this as the algorithm of the process.
4. **Each process get triggered by a set of start criteria:**
   - The process itself assumes a form only when it is executed.
   - A process is a program in execution.
   - The process can start being executed only when certain events defined as start criteria happen.
   - For instance, design can start only when the requirement specifications are signed off.
5. **Each process has a set of success measure:**
   - These enable us to measure the consistency and the degree of success of executing that process.
6. **The successful execution of each process is verifiable by a set of quality records:**
   - In a sense, these are like audit trails i.e. they are proofs that the process was carried out in compliance with the steps and meeting the success measures.
   - Many times the production of outputs may itself suffice as quality records.
7. **Finally each process has an owner who is authorised to change the process:**
   - It is assumed that the document processes are the final / developed of collective experience of the organisation.
   - Hence, it is important that great care be exercised in maintaining these processes and in making the necessary changes to these processes as and when required.

What constitutes/form/originate an effective process?

- A process should be useful for the organisation. Experience shows that for a process to be effective and useful the following attributes must be satisfied.

i. **Start with what is being practiced:**
   - Any process may be practiced in an organization before it can be institutionalized.
   - We may end up with only project specific processes which will serve no purpose.
   - The challenge is to achieve the right level of abstraction that would generally be applicable to most of the projects but at the same time, not be so generic that it does not add value.

ii. Document what is being practiced:

- One of the objectives of a process is to make the production egoless i.e. to reduce dependency on the person executing the process.
- A modest/egoless execution process must be documented.
- As the objective of these processes is that in day to day operations, it is important that such documentation is done by a group of practitioners who actually use the processes.
- By involving the practitioners, there will be a better buy-in and a better chance of success.

iii. Ensure that the processes are adaptable to a number of projects:

- The processes should not straight jacket the people.
- People should not feel that they have to force fit projects to processes

iv. Train the staff on the use of the documented processes:

- Even of the process is documented it would serve no purpose if new-comers can't be trained on its use.
- Training is the first step towards adaptation and widespread acceptance of the processes across the organisation.

v. Demand and enforce the use of processes:

- There will be always new-comers who may not know why the processes are needed or why they are used, the way they are inspite of this the amount of training they receive.
- It might require enforcement of some rules by the senior management for the processes to be followed in view of the benefits that an organisation has gained in the past.

vi. Measure the effectiveness of command or compliance to the process:

- All sports have a clearly documented process of scoring.
- There are no sports or game with no scoring.
- On a periodic basis, a random sample of projects is taken by an unbiased party (external or internal) and an objective assessment – called the audit is made as to how much the practice matches with what is documented.
- Deviations, non-conformances are raised and then the practitioners can decide whether what is documented needs to be changed or what was done was not right and accordingly take appropriate corrective action.

vii. Ensure that the processes are responsive , reactive , vulnerable , flexible for continuous improvement:

- It is very important that the processes are amenable for continuous improvement to provide sustainable competitive advantage.
- By analysing the information from the audits by bench marking against the industry practices, the processes must be continuous revisited to bring about improvements in cycle time, quality, productivity and cost.

Why are the processes important?

1. **Reduced time to market:**

- As one does not have to re-invent the wheel every time, the time to market a product is reduced significantly.
- In today's context of short product life cycles, anything one can do to reduce the time to market would certainly add to the top and the bottom line of an organisation.

2. **Simplification of training of employees:**

- The s/w industry is characterized by a significantly higher turnover of employees than other industries.
- Hence, getting new employees upto speed quicker increases the chances of on-time deliveries even if with attrition.

3. **Quality is more predictable & consistent:**

- Without documented processes, the quality of the final product & the steps taken to produce it are likely to be very dependent on the person in charge.
- With documented processes, this is less likely to happen as everyone follows the same set of processes.

4. **Scope of improvement:**

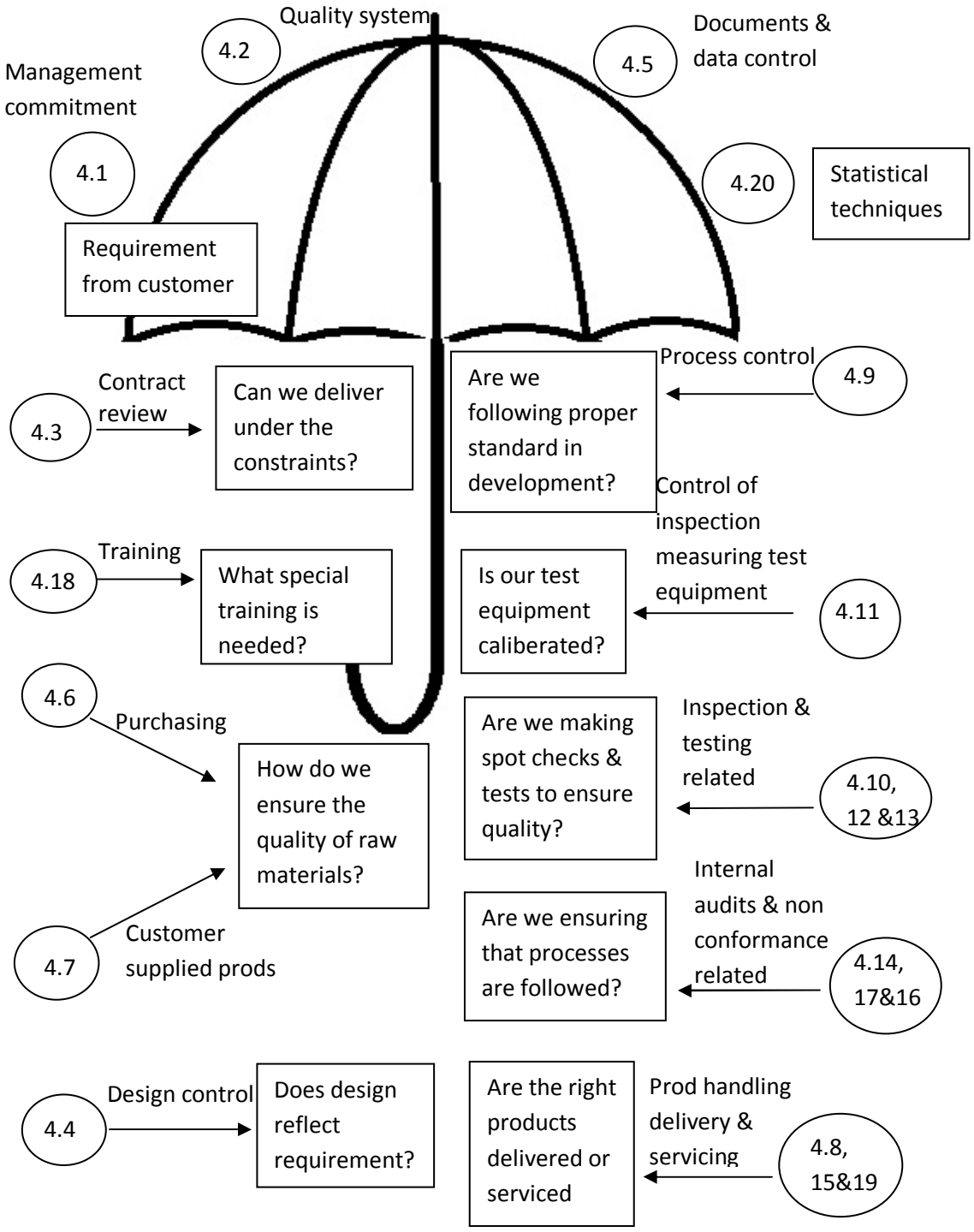
- Documentation is the first step towards improvement.
- It is only when we have a map that we can try to optimize the same time taken for travel by finding the shortest possible route.

Process Model

- Taking this basic modelling of the process several important organisational process model have been developed by various association.
- There are 2 most popular models:
    1. ISO-9001
    2. CMM(Capability Maturity Model)

1. ISO-9001:

- It was originally targeted at producing consistency, predictability and repeatability in the manufacturing sector.
- During the mid-1980's, the model was adapted for the s/w industry , with certain differences from the model for the manufacturing industry.
- The ISO-1980 model has a set of 20 clauses that create a framework for enhancing consistency and predictability of processes followed in an organisation.
- ISO come up with ISO-9001-3 mapping the 20 clauses to specific activities in s/w development where applicable.



(Fig. ISO-9001, 20 clauses as 20 good business practices)

- The intent of a s/w organisation is to understand the requirement of a customer and effectively translate them into a products that meets with his expectations.

- The first step is to ensure that we have indeed understood the customer's requirements and secondly that we have the resources and wherewithal to execute review tries to accomplish.
- Contract review (4.3) ensures that the project fits into the aggregate project plan of the organisation.
- The training clauses (4.18) address mechanisms to identify and provide appropriate training to the people.
- We have to ensure that the design is in accordance with the requirements before embarking on development.
- Design control (4.4) accomplishes this by introducing design review , design verification etc.
- During the execution various resources like machines , hardware , software and people are needed.
- Purchasing (4.6) and control of customer supplied products(4.7) ensure that the incoming material (or people) meet the criteria laid down for satisfactory completion of a quality product.
- Process control (4.9) lays down the framework to accomplish these important aspects.
- One of the principles for effective and efficient delivery of products as to ensure that any defect in the product be identified and rectified close enough to the point of time when it is injected.
- To facilitate this , ISO-9001 has introduced the concept of inspection. Inspection and testing (4.10) coves processes to test the product for defects while it is in process or after completion.
- Whenever we test a product we have to first ensure that the test equipment itself is error free.
- In the s/w context test equipment refers to things like testing tools, spreadsheets or database used for lagging, analysing the results.
- This function is addressed in control of inspection, measuring and test equipment(4.11).
- When a product fails some of the tests, it is essential to have procedures for identifying and segregating such products.
- Control of non-conforming products (4.13) pretty much focuses on this issue.
- The inspection and test status (4.12(puts in place process to ensure that the records of testing are properly maintained.
- Product conformance is also equally important to ensure process compliance.
- Internal audits (4.17) are a means of ensuring that the organisation follows the documented processes.
- Any process has to have quality records to demonstrate its compliance and that is what control of quality records (4.16) is all about.

- The activities are addressed by product identification and testing (4.8), handling, storage and packaging (4.15) and servicing (4.19) respectively.
- First of all a very well defined quality system (also called the Quality Management System or QMS) and secondly the implementation of the QMS should be taken seriously by the management.
- Last but not the least, there should be effective document and change control throughout the lifecycle of the project.
- Document and Data Control (4.5) addresses this very important umbrella activity.

CMM (Capability Maturity Model):

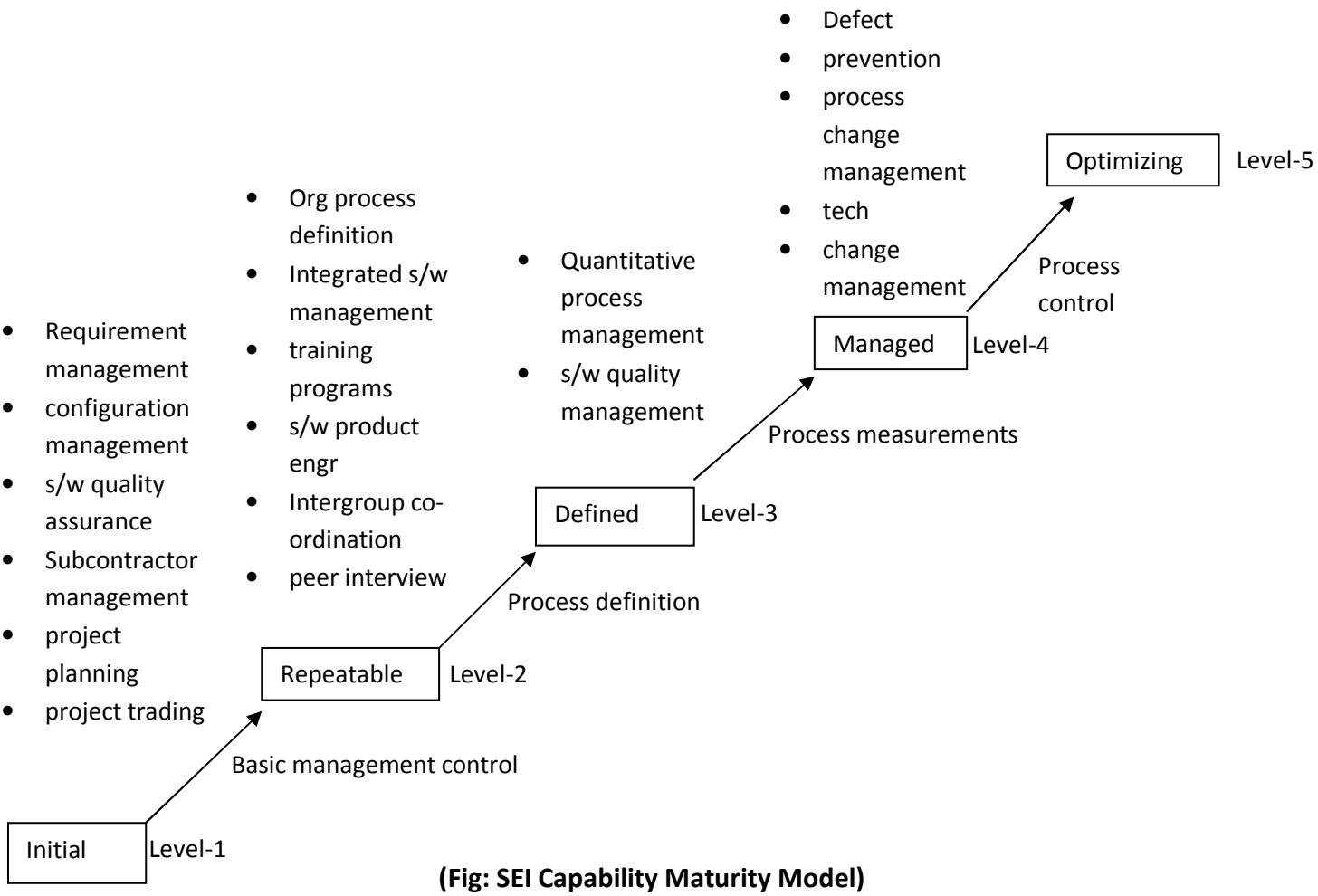
- CMM was evolved at the software engineering institute (SEI) by studying the development processes and practices followed in several large projects in organizations like the US department of defence.
- SEI abstracted the common features about how to produce s/w products with consistent, repeatable and predictable quality and how to institutionalize the best practices.
  Thus software CMM was born.
- CMM also strives to achieve predictability and consistency as a precursor to continuous improvement in the product by following a set of processes in a well defined framework.
- Unlike ISO-9001 which applies to all industries, CMM applies only to the s/w industry.
- The CMM defines five level of maturity of a s/w organisation.
- At each level, there are certain **Key process Areas (KPA)** that an organisation must satisfy.

Level-1:

- Level-1 of CMM is called the **initial level**.
- There are no processes that are either defined or followed in the organisation in a consistent manner.
- It does not mean that organisation can't be successful, but there is no predictability as to how they not perform in any future project based upon passed results.
- One of the main results is that these organizations thrive on the personal heroic of a selective, which may or may not continue working in the organisation.
- There is a clear need for established processes in areas like support, servicing and maintenance and these are generally not the forte of the heros of Level-1 organizations.

Level-2:

- It is called the **Repeatable Level** puts in place the management processes that help achieve repeatability of performance and quality, should the organisation undertake a similar project again.
- The KPAs of this level are directly relevant to life cycle activities; they start off with Requirements Management putting in place mechanisms to ensure that we understand the requirements properly, and then they take on s/w project planning & project tracking.



(Fig: SEI Capability Maturity Model)

Level-3:

- Level-3 spread the process culture to the entire organisation.
- Organizational process Definition makes the organisation define the process for execution of its day to day functions (This is done by the practitioners).
- A drive to organizational process focus is started and maintained by activities such as forming of a Software Engineering process Group (SEPG).
- In Large projects, one of the most important factors of success is that the different groups work together effectively.
- As a first step towards correcting the defects as close to the point of injection as possible, peer reviews are introduced.
- Finally, integrated software management addresses issues that tie the entire life cycle together.
- When an organisation reaches level-3, it would have achieved not only stable management practices but also well defined processes on how the actual business is carried out .

Level-4:

- Quantitative process Management KPA provides an organisation with a framework to define, capture and analyse appropriate metrics.

- A Level-4 organisation is usually able to quantify its processes and quality in terms of numbers.
- These numbers and metrics are most often directly relevant to the performance and profitability of the organisation as they are actually defined by the practitioners.

Level-5:

- It is called the optimizing level.
- Here the processes are continually fine-tuned and people are always working towards some targets to achieve better and still better results as they go along.

Characteristics of CMM:

1. **Commitment to perform**: Does the organization display the commitment needed to perform in the KPA?
2. **Ability to perform:** Do the people have the ability to perform what is needed to achieve the KPA?
3. **Activities performed:** what are the actual activities performed to satisfy the KPA requirements?
4. **Verification:** How do you actually verify to know that you are on the right back**?**

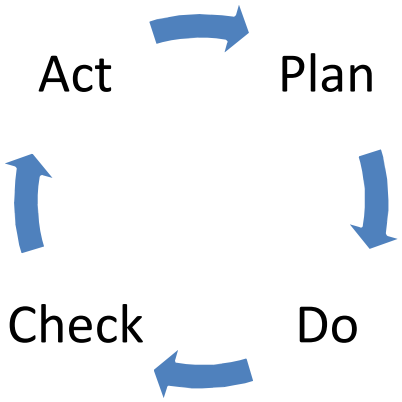
Module-II

Metrics:

- The best motivation for Metrics comes from a quotation by Lord Kelvin(of the Kelvin Temperature scale fame).
- "Metrics in a project Management context is about measurements-Measuring our progress in order to know where we are and what mid-course correction we need to take to achieve our goals".
- There are 2 components:
    1. Metrics are measurements to measure our progress.
    2. The Measurement of how well we have achieved goals.
- When we say that metrics is all about measurement , several important questions arise:

i. What should be the ideal roadmap for metrics?
ii. What should be the strategy for implementing metrics?
iii. What are the individual steps that constitute metrics?
iv. What are the common management issues and pitfalls to watch out for in implementing metrics?

(i) The Metrics Roadmap:



- The strategy for a successful metrics program falls in line with **Deming's PDCA quality model** as illustrated.
- P.D.C.A stands for **plan-Do-Check-Act**.

1. **Plan:**
    - For any project we plan what you want to achieve.
    - While making plan you have to consider the current position and your eventual destination.
2. **Do:**
    - Then we do or carry out the plan.

- This entails carrying out the necessary steps for the actual execution and also carries out the measurements for health check.

3. **Check:**
   - The check phase compares the measurements made against the expected norms set in the plan phase.
   - If things are found to be out of line then the corrective actions need to be taken.
4. **Act:**
   - The corrective actions need to be taken in the Act phase and we start all over again by refining the plan.

(ii) A typical Metrics strategy:

(a) Decide what we want to measure & how we are going to measure it:

- It is very important for all the constituents to know the rules of the game i.e; what are the measurements that matter and how they are carried out.

(b) Set targets and track them:

- It is important to set specific quantitative targets.
- This step is about how to quantify and track the targets.

(c) Understand variability and work towards minimizing it:

- One of the fundamental texts of quality performance is achieving consistency and predictability.
- Variability is the opposite of predictability.
- This step is about how to track variability & what to do to minimize it.

(d) Act on data and strive for continuous improvement:

- Measurements are useful only when we act on them and take conscious decisions based on them.
- This step discusses what kind of action is needed and how to achieve this action.

(e) Consider the human angle:

- Unlike most other industries, the s/w industry is very people-oriented .
- We can't measure people's progress the same way we would measure a machine's performance.
- There are significant human issues that need to be addressed and this step covers some of these issues.

(iii) What should we measure?

- Is the entity that we want to measure directly relevant to the goals of the project/organisation?
- Is the entity easily or "naturally" measurable?
- Can we quantify the cost and benefits of measuring?
- Is the entity controllable?
- Can we afford NOT to measure the entity?

(iv) Set Targets and track them:

- After the measurement process, the next step is to ensure that certain targets are set for these entities.

Conclusion:

- Metrics is an all-pervasive and all encompassing activity that takes place throughout the project lie cycle.
- If we don't know where we are going or how to get there or how to measure the progress in our journey, then it is likely that we will not reach the destination.
- Metrics is the most challenging activities of project management that it requires a very delicate and fine balance between engineering expertise and human finesses.

S/W Configuration Management:

Configuration Management:

- Configuration management is the combination of s/w services and processes that enable each developer to re-create and use the exact set of files and environment for a specific s/w product or platform.
- It is also a mechanism to isolate work environment of the individual developers, ensuring that none of the valid and confirmed changes get lost.

Configuration:

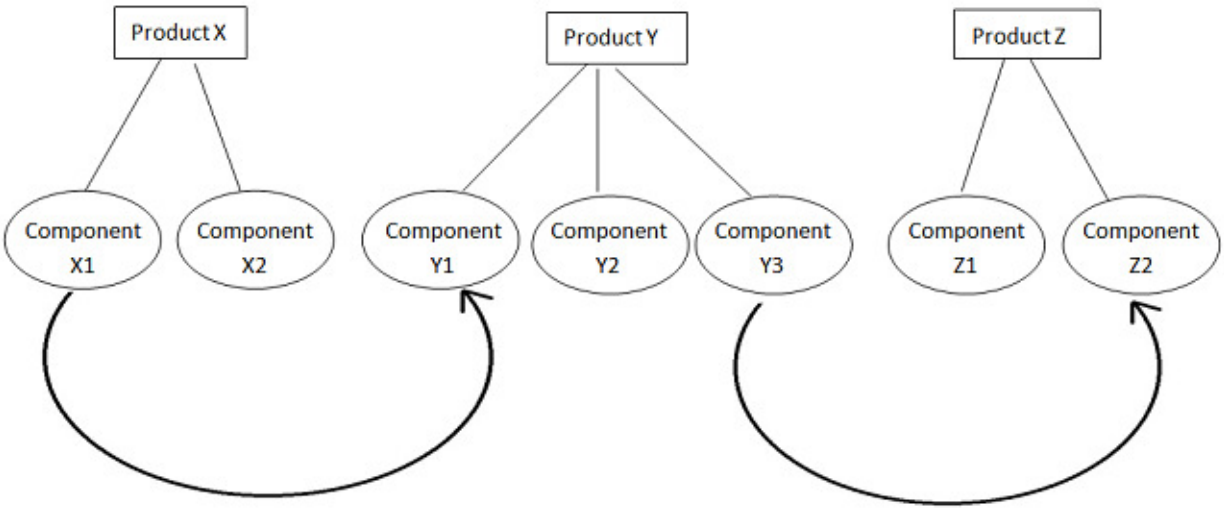
- A configuration is a set of related items satisfying the following criteria:
- The configuration is uniquely identifiable by a configuration Id.
- The items are consistent i.e. the items works with one another in a way that well understanding.
- The set of items is re-creatable as a unit.

Configuration Item:

- It is an elementary part of the configuration that must be :
  1. **Identified or versioned:** Each item has a version number that characterises that change history.
  2. **Tracked:** Any activity on the items is tracked.
  3. **Controlled:** Any update to the item goes through a well documented and controlled process.

✸ Characteristics of s/w development in product development organisation:

1. **Break up of products into components and inter-component dependency.**
   - S/W products are broken down into multiple inter-related and inter dependent components.



(Fig: Products, components and dependencies)

2. **Co-existence of multiple versions of a product:**
   - Each s/w product reaches the market in different versions, each version likely to be more feature-rich and more powerful than earlier versions.
3. **Simultaneous or conflicting working by multiple people on the same work product:**
   - Given the size and complexity of today's s/w products, it is almost guaranteed that many people will touch the same code multiple times during the products lifetime.
   - It is also highly likely that more than one developer would attempt to touch the same code to fine different problems.
   - This scenario is very important and similar to synchronizing updates in multiuser database management systems.
4. **Simultaneous support for multiple hardware and s/w platforms for a product.**

Process and Activities of SCM:

✸ SCM(S/w Configuration Management) is an umbrella activity that applies through all the phases of a s/w development lifecycle.

✸ The steps that constitute SCM are :

1. Initial working.
2. Base lining.
3. Change management.
4. Management of workspaces.
5. Configuration status accounting.
6. Configuration audit.

1. **Initial working:**

- There is a small set of people who do the initial research and communication as well as the spade works that is required.
- During this time, most of the files are held privately by the individuals.

2. **Base lining:**

- It is a process by which a given set of items formally become publicly available in a standard location to the people who are authorised to use it.
- The collection of work products is identified as a configuration as a configuration and given a configuration id.
- Each of the individual files (or configuration items to use a SCM terminology) is associated with this configuration ID, given a unique version number and stored in a standard location as a unique file.
- Base lining involves the following activities:
  - Freezing a current version of the product and its constituent elements like source files, make files etc.
  - Allocation a configuration id to the entire configuration.
  - Allocating version numbers and standard locations to the constituent elements of the configuration.
  - Starting the approval authority information.
  - Finally broadcasting the above information to the concerned people.
- The vehicle used for these storage and communication purpose is the Configuration Management Repository (CMR).
- CMR contains information about all the base lined items.
- Repository also records all the change history information.
- The process of base lining is also referred to as check-in.

3. **Change Management:**

- After any configuration items get checked-in and are base lined, changes are bound to take place.
- These changes can come about because of several reasons change in requirement, changes in design or architecture, changes in the platform or environment or simply changes caused by typical of the programming fraternity.
- The objective of change control (and indeed of SCM) is to ensure that change are well throughout and done consciously.
- One should be fully aware of all the implication rather than make ad-hoc changes.
- Hence the change control process is divided into 4 types:
  1. Change request
  2. Change review and authorization

3. Change execution in a workspace
4. Change check-in and re-base lining

a. Change Request:

- At some point of time after base lining and when the product is in use , someone discovers there is a need to make a change in some of the items.
- When the need for change is realised, the person who realises such a need raises a change request.

b. Change Review and Authorization:

- The review / approval authority review the change request from three dimensions are Necessity, Appropriateness & Impact.

i. Necessity:

- This dimension establishes whether a particular change is required at all what business or technology change warrants this change request , does this change request arise from a genuine requirements change .

ii. Appropriateness:

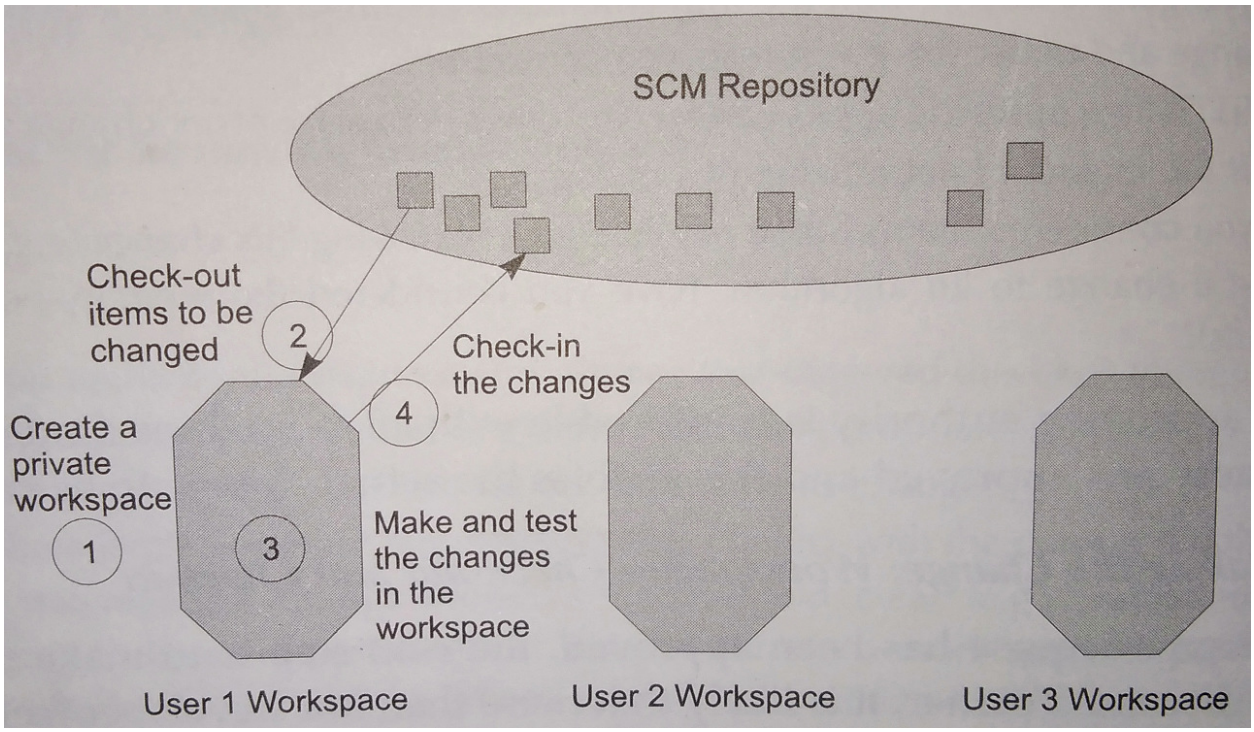
- Having ascertained that the change request is necessary and genuine, the 'appropriateness' ensure that the change is carried out the right way.
  Appropriateness focuses on "How".

iii. Impact:

- The final dimension of an analysis before approval is the impact analysis questions such as "if this change is made in the manner proposed, what impact would it have on other components?", "How much effort would go onto making this change and what should be compromised to ensure that the resources required for it are made available", etc.
- Impact analysis also considers the effect of other pending changes on this change request and vice versa.

c. Change execution in a work space:(Check-out and Check-in)

- Once the change request has been approved, the next step is to make the actual change.
- When 2 users are trying to update the same records, such record/update must preserve the ACID property.
- ACID stands for Atomicity consistency Isolation and Durability.



(Fig: Workspace, Check-out and Check-in)

- The workspace replicates the environment in which the developer can build the product under the same conditions in which the corresponding baseline was built.
- In order to ensure that no 2 users are making changes to the same file at the same time, a file has to be checked-out from the SCM repository.

d. Change Check-in & Re-base lining:

- The check out items, when change and tested are ready for re-base lining.
- In this level the developer checks-in the files he had checked out earlier.
- This re-check is also called re-base lining.
- It achieves the following:

1. A new copy of the checked out items is created with an updated version number.
2. The items that were marked as "checked out" are marked "free", thus cleaning the way for other people to change the files if their change requests are approved.

5. Configuration Status Accounting:

- Configuration status accounting refers to dissemination of information about changes in configuration that have takes place or are in the pipeline.
- In large development projects, it is quite possible that the developers become so focused on their specific enhancements or bug fixes that the big picture gets lost and no one knows the overall effect of all the changes.

- By storing all the change requests reviews, approvals & the actual changes themselves in a data base and using the decision support & query tools of the database, answers are obtained for questions such as:

1. What were the changes made on a specific module or an item?
2. Why were the changes necessitated?
3. Who made the changes & when?
4. Which of the changes were re-works of previous changes?
5. What were the costs incurred on each change request?

6. Configuration Audit:

- An audit is a means of sampling various representative instances and ensuring process compliance in these instances.
- The principles of a audit-Random sampling, independent authority, sampling for process compliance rather than individual fault-finding, a human, non-policing, non-threatening approach are all applicable in the case of a configuration audit.
- The process of configuration audit should take some sample of the changes in the system and should ask the following questions:

1. Does this change have a change request, review records that document necessity, appropriateness & impact analysis?
2. Has the change request gone through the appropriate authorities for review/approval?
3. Does the actual change confirm to the intend change as mentioned in the change request?

Metrics in SCM (Software configuration management):

- S/w configuration management is an umbrella activity in the project management life cycle.
- The metrics vary with each project & each situation but some general guidelines that would help the (reader) in choosing the appropriate metrics for his project:

1. How many change requests are made?
2. Categorisation of such change requests by module/project: This is an indication of the stability of that module or project.
3. How many change requests are re-work change requests? This is a measure of how well throughout and well implemented the changes.
4. How many change requests are enhancement change requests? This is a measure of the requirement stability
5. How many change requests get approved?
6. How did the resources predict at the time of change request match the actual resources expended?
7. How many changes actually happened on the system that bypassed the request approval process?

SCM Tools and Automation:

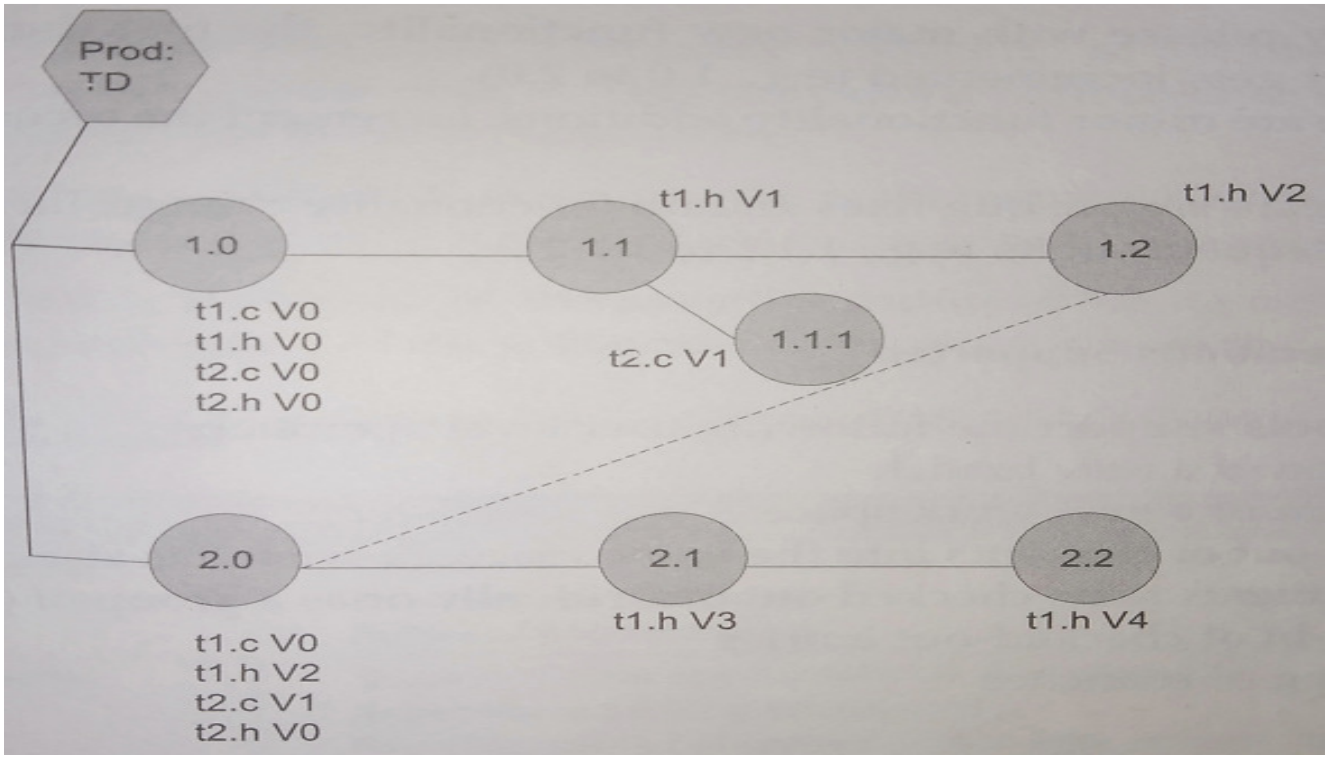
✸ SCM is a fairly labour intensive operation where there is a lot of scope for human error .

Functionalities & concept of SCM tools:

1. Repository structure & identification of elements.
2. Operations supported.
3. Administrative support.
4. Facilities for geographically distributed projects.

1. Repository structure & identification of elements:

- Most SCM tools use a hierarchical tree structure to represent configuration.
- Each tree represents a particular product under development.
- There are several branches in each tree, each branch representing a particular major version of the product.
- Each file by itself is identified by 2 attributes-it's name, which is invariant across multiple releases & it's version number.
- The version number of each file could increase independently-some files could get changed more frequently than the others.
- Usually each file by itself is identified by 2 attributes- it's name, which is invariant across multiple releases and it's version number.
- The version number of each file could increase independently- some files could get changed more frequently than the others.



(Fig: Tree structure to represent object/product versions)

- In the above figure, the file t1.h has gone through 5 revisions while the file t2.c has gone through 2 revisions & others files have not been changed.
- The files can be reconstructed by using the following simple steps:
  1. Start from a leaf of the tree under that branch. Include all the files in that leaf.
  2. Go higher up the tree and include the names of those files which have not been included so far.
  3. When you come to the root, stop.
- The final reconstructed list of files for a given version is called the Bill of materials for that configuration/version.

✷ **Q. When do you increment which part of the product version?**

Ans: That is a convention which varies from product to product or company to company.

- For any release with major new functionality, the first digit of the version number gets incremented.
- If there are minor functionality additions, increment the second digit.
- If there are simple bug fixes and no functionality change, increment the 3rd or subsequent digits (e.g-1.1.1 to 1.1.2).

2. Operations supported:

- Most SCM tools support the following user level operations-
  - Creation of a new branch
  - Creation of a new work, space
  - Check-out of elements into the work-space. Provision is also made to identify the elements to be checked out individually or as a group of objects.
  - Check-in of check-out entries
  - Merging of branches

3. Administrative support:

- Most SCM tools provide system administration support.
- Some of the examples are creation of user ids, association of privileges with objects for users, space management (allocation, expansion, defragmentation, etc) and backup or restore.
- Such Administrative operations are usually privileged operations and not available for all the users.

4. Facilities for geographically distributed projects:

- The SCM tools whether use the centralized model or the distributed model- provide access to all the functions through a web browser.
- Some of the SCM tools allow the system administrator to define multiple sites that would contain copies of the same objects.

- Some of the SCM tools also allow federated repositories wherein one logical repository is spread over several physical repositories.

- The SCM then provides a seamless access to the users to access any object, regardless of the physical location of that particular object.
  - Some of the SCM tools also allow temporary "cacheing" of the objects closer to the user.
- This provides a performance enhancement and places less Load on the network bandwidth.
  - Some of the SCM tools also provide mechanisms like email alerts to broad cast changes.

S/W Quality Assurance:

Quality:

- **From a customer's view point:** quality is, "how well does the product meet my needs?"
- **From a producer' perspective:** quality refers to, "how well does the product function comply within what I said it would do?"

- A product that was developed to meet the original needs may no longer be adequate in the context of the changes.

i.e. If the original system was designed to handle an incoming load of 100 transactions per second but the volumes increase and the load increases to say, 500 transactions per second, it is likely that in the customer's opinion the product does not meet his current needs.

- **Quality** is about transforming as many of the implied requirements of the customers into stated requirements and meeting all the stated requirements.
- It is about minimization the implied requirements and minimizing the gap of unfulfilled requirements. Any such gap is called a defect.
- There is an alternative definition of quality called the quality of design.
  i.e. A Mercedes car is more expensive and perhaps more glamorous to own than say, a Suzuki or a Hyundai. Does it mean that Mercedes is of a superior quality? All that it means is that the Mercedes was designed for a different market, a different budget range and a different purpose. Stated in other words, the quality of design for Mercedes is different from that of Hyundai/Suzuki.

Importance of Quality in S/W:

i. **S/w is pervasive (Present or influence):**
   - S/w has become an integral part of almost everything that we use in our day to day life.
   - Any defect anywhere in any of the systems is more than likely to touch us, affect us and even paralyse one or more aspects of our lives.

ii. **S/w is increasingly becoming mission critical:**
   - A lot of the nuclear and power plants around the world, as well as the life saving devices in the hospitals & those people carry have a significant amount of embedded s/w.

- Any defect in these s/w programs is bound to have a life threatening impact.

iii. Expectations are increasing:

- Those the days are gone when people used to be resigned to the fact that s/w is bound to have defects & bugs.
- Customers are demanding that s/w be bug free & defect free, just as much as they expect automobiles not to stall on traffic lights.

iv. Scope for errors is high:

- S/w industry is always characterised by people working under intense pressure & trying to meet deadlines which are not always rational or realistic.
- such high pressure situations, the scope for errors becomes very high.
- With the pervasiveness of s/w, there is really no second chance to correct a problem.

v. Qualified and willing people are scarce:

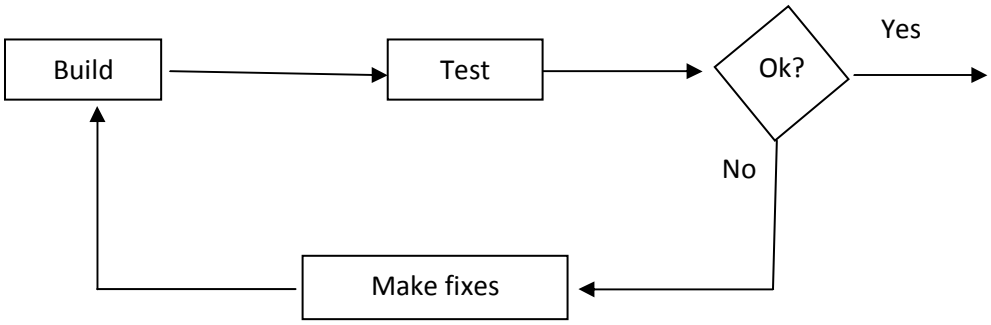
- The s/w industry is characterised by high attrition, defects become a bad legacy.
- Trying to unearth and fix the problem put in by somebody else is not an easy task and often times, not relished by s/w engineers who always took for "interesting" design work instead of "maintenance work".

Quality control and quality assurance:

- It is the presence of defects that causes non-compliance of the actual product functionality with the customers need.
- To minimize defects there are 2 approaches:
  1. Quality Control
  2. Quality Assurance

1. Quality Control (Qc):

- It refers to testing a product after a given phase to find out if it has any defects and in case it does have defects, to employ corrective measures to remedy the same.



- Done after the product is built. Hence it is usually reactive.
- Expensive and sometimes impossible. i.e. for life saving devices or mass produced devices, it may not be possible to find a problem after it is discovered.

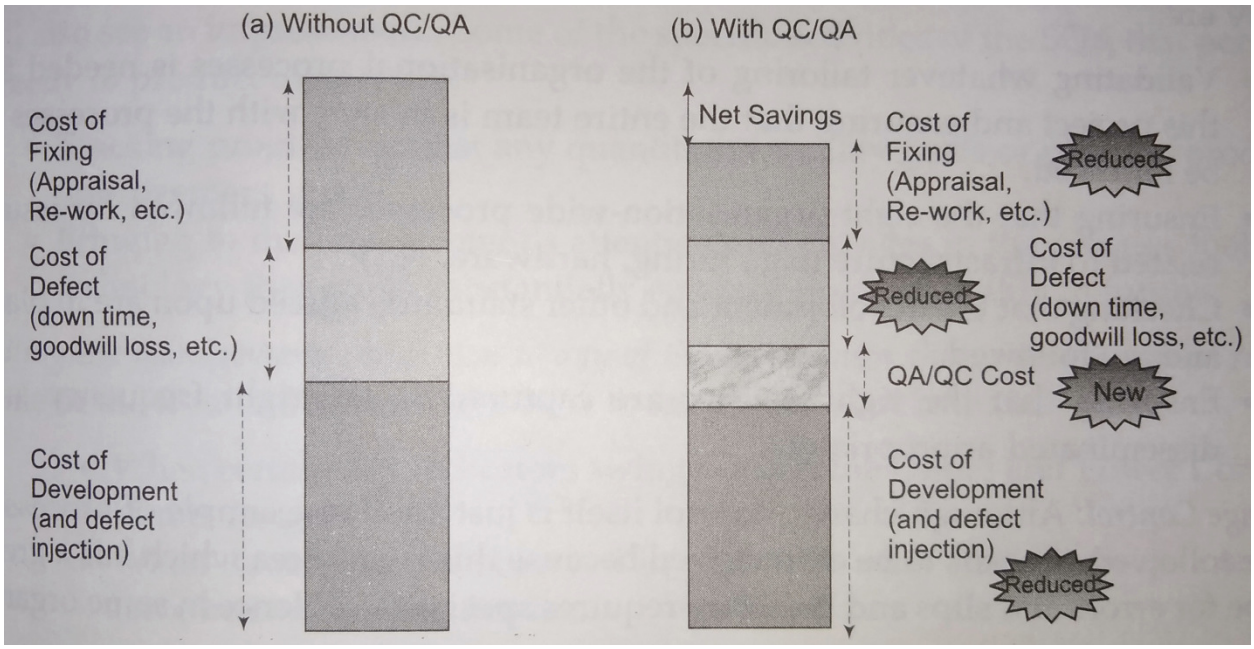
- Oriented towards defect detection rather than defect prevention.
- Ex ample of quality control: Unit testing done on each module after it is developed by the time such testing is done, considerable design and development effort has been expended.

2. Quality Assurance(QA):

- It tries to go one step further.
- Instead of concentrating on post facto defect detection and correction, it focuses on the prevention of defects from the very start.
- Meant to be pro-active, i.e defect prevention oriented rather than defect detection oriented.
- Intended to catch the defects as close to the point of injection as possible rather than let the defects trickle down to subsequent levels.
- Applies to the process rather a specific end-product, thus this can be expected to have a wide ranging impact &benefits.
- Happens across the board and not "in line"(that would explain why you do not find a flow diagram for quality assurance as you do for quality control).
- Some of the tools used for quality assurance include reviews, inspection & audits.
- In general defect prevention is always better than defect detection.

Cost and Benefits of Quality:

- In order to measure the cost & benefits there are 2 scenarios:
  1. Firstly without QA and QC.
  2. Secondly with QA and/or QC.
- The defects are detected by the consumers.
  - This results in some amount of down time for the consumer and possibly causes a loss of goodwill (the impact of which cannot always be easily quantified).
  - These costs are called cost defect.
- The defects then get reported to the people who actually injected them.
- These people then have to do an analysis of the root cause of the defect(called appraisal cost) and make the corrections.
- The last set of costs to do with fixing and re-packaging are also called re-work cost.
- Total costs are the sum of the initial development costs, defect cost, appraisal and the rework cost.



- **Total Cost = Initial Development Cost + Defect Cost + Rework Cost + Appraisal Cost**

S/w Quality Analyst's Function:

- The role of a s/w quality analyst (SQA) can be viewed as that of a "Conscience keeper" of the project.
- The SQA is someone with a character and a mandate to alert the top management on any possible quality consistency slippage (A reduction in the rate , amount or loss popularity)
- The main function of the SQA can be classified into the following 5 major areas:
  1. Requirement fidelity
  2. Process compliance
  3. Change control
  4. Minimising the gap between defect injection & detection
  5. Product quality

1. Requirements Fidelity:

- It is very important to capture the right requirements & these should be reflected in the final product.
- This would include:
  - Ensuring that the requirements are signed off by people vested with the appropriate authority.
  - Assisting in converting as many of the implied requirements into stated requirements
  - Carrying out checks to ensure that the design & development incorporate these requirements.

2. Process Compliance :

- The set should ensure that all the development and organisational processes are compiled with some of the activities that fall under this category are:
  - Validating whatever tailoring of the organizational processes is needed for this project & ensuring that the entire team is in sync with the processes to be followed.
  - Insuring that the right organisation wide processes are followed for issues related to infrastructure.
  - Checking that the development & other standards agreed upon are in placed & are followed.
  - Ensuring that the right metrics are captured at the right frequency and disseminated appropriately.

3. Change Control:

- It needs to be emphasized because this is one area which has a great scope for errors and slips & therefore requires special care.
- In some organisation a special role is carried out that of the project configuration controller (PCC) to oversee the change control related activities of the organization.
- The role of the SQA involves:
  - Insuring that the change control requests & records are in place and that all the necessary information needed by the reviewing /approving authority is in place.
  - Maintaining appropriate audit trails when an approval comes back.
  - Validating the re-create ability &currency of the configuration.

4. Minimising The Gap between the Points of Injection & Detection:

- If defects can't be prevented, at an early detection should be the next best target.
- It is advantageous to catch and rectify the defects as close to the point of injection as possible.
- Some of the responsibilities are:
  - Confirming that the completion criteria for each phase are defined.
  - Ensuring that documentary evidence exists for such phase-wise completion criteria that have been met.

5. Product Quality:

- The role of SQA involves:
  - Tracking progress against any quantitative quality goals or process or product improvement goals.
  - Bringing to the management's attention any change in the process tools or technology that could substantially enhance product quality in future.
- Some of the factors that cause the SQA to raise a red alert include:
  - When certain key indicators swing outside the upper & lower control limits ranges.

- When deviations are observed in processes even after the project teams have been alerted to this fact.

✸ Some popular misconceptions about the SQA Role:

➢ **SQA = Testing:**

- Testing is done after a product is built.
- The quality of a product can't be enhanced simply by increasing testing.
- Quality has to be planned right from the initial stage of product conception and design.
- Quality assurance is more than testing and SQA's activities which are quality assurance oriented can't be equated to testing.

➢ **SQA = Fault Finding:**

- The SQA should not be seen as someone who simply finds faults and nitpicks all the time.
- Then the effectiveness in the set & ability to get a quality control across the organisation will suffer badly.
- The suggestion of the SQA should be seen as constructive and value-adding.
- The SQA should also highlight and propagate the best practices he observes in a project.

➢ **SQA = Management's Spy**

- The SQA should not be seen as the "Management's spy" whose only job is to keep complaining to the management.

➢ **SQA is the only person responsible for Quality:**

- Every one in the team is responsible for quality.
- The SQA can indicate possible areas where problems exist and improvement needs to be made.
- It is the responsibility of the entire team is act on the recommendations of the SQA.

➢ **SQA is responsible for fixing all problem:**

- The SQA is only a messenger or a conduit that gives an update on the state of the project, product & process quality issues.
- He is not directly responsible for fixing all the problems he has reported.

S/W Quality Assurance Tools:

❖ Review & Inspections:

✸ Formal reviews and inspections are among the most effective tools to weed out the defects from as close as possible to the point of injection.

✸ There are some distinction made in the literature between reviews, inspections & walkthrough.

✸ **The salient features of a formal review are:**

1. A review is formal i.e. there are formal, pre-defined roles by each participant and broadly follows the process outline below.

2. For any work product to be reviewed, a review team is formed. The review team consists of following person:
   a. The author of the work product.
   b. A chairperson (usually other than the author) also called the moderator.
   c. A scribe.
   d. One or more reviewers who are qualified to review the work product & have no bias in terms of reporting structure of hierarchy.
3. The project manager/process manager forms the above review team.
4. The review team has an opening meeting. During this meeting the author of the work product presents a general overview of the work product i.e what is its objective & any other relevant details like implementation details, development standards etc. He also ensures that such review team member has access to the work product (either a soft copy or a hard copy) & any relevant checklists or criteria against which to evaluate the work produced.
5. The moderator decides when the actual review meeting is going to be held. In between the opening meeting and the actual review meeting, the review team members are expected to diligently go through the work product & review it using any application standards & guidelines as appropriate.
6. The review team meets on the stipulated date & goes through the work product sequentially. The moderator controls the meeting. the scribe records the proceeding of the meeting.
7. Each defect found in the work product is also recorded & classified into 2 categories.
   a. The first is major vs minor, with major defects obviously being more serious.
   b. The 2nd category is whether it is a systemic defect or a miss execution defect.
8. After the meeting the author a performs root cause analysis for the defects and fixes problems.
9. The moderator ensures that all the open defects are fixed. In case any process changes are needed, he conveys them to the process quality group.

✸ In case any process changes are needed, he conveys them to the process quality group.

To get the best results from the review process, the following guidelines have been found effective:

- The review team members clearly understand & faithfully enact their roles.
- All the team members come fully prepared for the meetings.
- The review is only of the product and not of the author.
- The process of identifying the problems should be the prerogative of the review team. Fixing the identified problems is the author's responsibility & domain & should not be a part of the review meeting.

- The defects found in the review process are not to be used in the performance appraisal of the author.
- To this end, the author's management chain should not a part of the review team.
- The moderator keeps the discussions focused on the work product and does not let any unnecessary small talk in the way.

❖ Audits:

- A well conducted audit is a very effective software quality assurance tool.
- Audit follows the broad process as below:

1. The senior management/the process quality team schedules periodic audits & communicate the schedules to the entire organisation.

2. The audit can be conducted either by an external body or a team of trained internal auditors.

3. The audit is always performed using the existing documentation of the processes of the organisation.
   - In-addition, if the audit is for certification against any public standard like ISO-9001, such standards also become yardsticks against which the audit is performed.

4. Each audit is assigned to lead auditors. The lead auditor is responsible for the following:
   - Selecting the members of the audit team-either trained internal auditors or external auditors.
   - Maintaining & communicating the schedules & responsibilities to all concerned.
   - Communicating the findings of the audit to the top management.

5. The lead auditor selects a sample of projects that are to be audited. The sample would be effective if:
   - It covers most of the representative projects in the organisation.
   - It covers some of those projects that have had problems in the past.
   - It covers projects in different phases of life cycle.

6. The lead auditor assigns a team of auditors to perform the audit of each project.
   - To be effective, the auditors for a given project should neither belong to the project team nor should they be in the direct management chain of the project being audited.

7. The audit team audits the project team for compliance against the documented processes & any applicable standards.
   - During the audit the audit ensures that all the relevant processes are practiced properly & that all relevant quality records are available.

8. The auditors document any variation between the processes and practices based on objective evidence as non-conformances.
   - The auditee's consent is usually sought before documenting the non-conformance.

9. The auditee comes with corrective actions & preventive action.
10. The lead auditor collates all the non-conformances, corrective & preventive action & conveys these to the top management.
11. The top management and the process quality group perform analysis on the collated data to identify overall process improvement possibilities.

❖ Cross-SQAs Information Sharing:

- ✸ The SQAs should communicate among themselves on a periodic basis, exchanging notes on their experiences & practices.
- ✸ It achieve the following purpose:
  - ➢ Early warning signals of organisation wide (and potentially endemic) issues can be got.
  - ➢ Best practices that take place in isolated pockets can be disseminated through information sharing.
  - ➢ Novel methods adopted by SQA's to do their job can be shared.

❖ Defect Classification & Analysis Tools:

- ✸ Two of the more popular tools in use for classification & analysis are:
  1. Pareto Analysis
  2. Fish Bone diagram

1. Pareto Analysis:

- ➢ The 80-20 (pareto) rule is probably one of the most widely observed laws of nature.
- ➢ 80% of the productivity of an organisation is contributed by 20% of its people, 80% of the value of inventory is accounted for by 20% of the items.
- ➢ On the same line, when the problems are analysed & their root cause identified, it is normally found that 80% of the problem are caused by a very specific set of root causes.
- ➢ If that specific set of root causes can be attacked & improvement brought about, it would go a long way in removing a majority of the problems.

2. Fish Bone diagrams:

- ➢ The fish bone diagram is another common tool that is used for getting to the root cause of the defect.
- ➢ Starting from the basic item being measured on the "spine", he fish bone diagram creates new branches for any cause of defects.
- ➢ Each gross level classification can be broken down into sub-branches or sub-bones which would give a further refinement of the gross cause.

Measures of SQA success:

1. Process non-conformances that escape the SQA's evaluation:

- ➢ It is the responsibility of the SQA to prevent defects or catch the defects as close as possible to the point of injection.
- ➢ Defects can be product defects or process non-conformances.

2. Number & impact of process changes suggested:

- The SQA suggestions should result in improvements in at least one of further dimensions-products quality, cost-effectiveness, customer satisfaction and employee satisfaction.

3. Impact of process quality on product quality:

- Any process or step is followed should have a material impact on the product quality.
- The success of the SQA function is measured by the impact on product quality of any process changes brought about by the findings of the SQA.

4. Applicability of process or product changes across projects:

- The impact of a change would be more widespread if there are more people who benefit by the change.
- If the SQA can changes that can have ramifications & benefits in multiple projects, they are likely to herald greater improvement than would the project-specific process changes.

5. Number & impact of best practices disseminated across the projects:

- SQA brings to the attention of the project team & the management not just the problems but also the things that are done well.
- Hence, one of the measures of the SQA's success is how well the best practices identified by him are disseminated across the organisation.

Risk Management:

❖ Risk:

- Risks are events that are usually beyond the planner's control.
- When a project plan is put together, a target is get and the objective is to strive to meet or beat the target.
- But there could be certain completely unanticipated events that could derail the progress towards the achievement of thus target.
  - e.g.: Acts of god- like earth quake, floods and changes in macro economic or political scenarios, sudden departure of key personnel etc.
- Risks if not accounted for, can guide or control the course of activity away from the desired or expected or planned outcome.
- Risks are events that are worth anticipating and having alternative plans. All through an infinite number of risk & scenarios may be possible it would be practical to focus on all the risk factors.

❖ Characteristics Of Risk Management:

- There are 2 characteristics of risk management:

I. It is Proactive:

- We anticipate what could go wrong and plan accordingly.

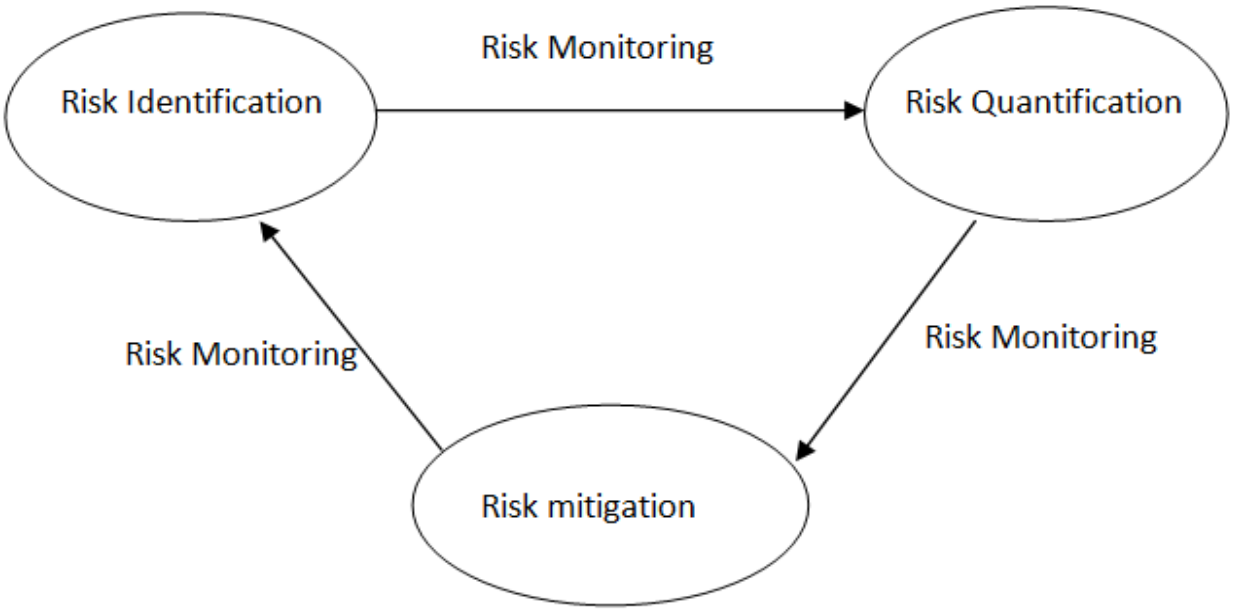
- If and when what we had anticipated happens, we are better prepared & have an alternate strategy for arriving at our goal.

II. It strives to reduce the impact of uncertainty:

- "Plan B" (or alternative strategy) is designed to minimise the deviation from the expected outcome.
- Thus, notwithstanding the risks, the predictability & consistency of the final result is not compromise very much.

❖ Risk Management Cycle:

- Risk management is an umbrella activity that takes place throughout a project life cycle.
- Risk management comprises of 3 phases:
  1. Risk identification
  2. Risk qualification
  3. Risk mitigation



1. Risk Identification:

- Risk identification is the process of identifying those risks that a project manager needs to guard against.
- There are 6 different ways of identifying the potential risks:
  - **i.** Examining organisational history
  - **ii.** Preparing checklists
  - **iii.** Information buying
  - **iv.** Framework based risk categorisation
  - **v.** Simulation
  - **vi.** Decision trees

i. Examining Organisation History:

- By looking through the track record of an organisation , one may identify the certain patterns of risks.

e.g. When one gets a new project from a customer, a review of the previous projects done with him may reveal patterns like the customer changing requirements very often.

ii. Preparing Checklist:

- Checklists are a great way to ensure that we are not missing out on any obvious inputs to a project.
- It is worth having a checklist that one can consciously validate the results against.
- It is important to note that checklists should be used as guidelines rather than dogmatic commands.
- Checklists evolve as time progresses and encapsulate organisational learning from the past experiences.
- Working with an outdated or an inappropriate checklist of risks is in itself a great risk.

iii. Information Buying:

- When we encounter risks in our projects we tend to think that our situation is very unique.
- While identifying the risks for a project , it is certainly worth looking at the industry treads and using the available literature & information judiciously.
- Related to utilising information in the literature is the concept of information buying.
- Everyone knows about the spiralling salaries for the s/w jobs and that if as an organisation, one does not keep peace with the salaries in the market one cannot attack or retain top talent.
- This would probably be the number one risk for any organisation. In order to identify their position, most organisations participate in compensation surveys conducted by natural / third party agencies.
- By sharing and aggregating information, each organisation is able to buy information as to where stands in the market with respect to salary and compensation and thus minimises the risk of not being able to attract or retain top talent.

iv. Framework Based Risk Categorisation:

- Providing a framework or broad categories of risks could establish a channel of thought processes to enable identification of general risks and project-specific risks.
- It can be classified as follows:
    1. Technology risks
    2. People related risks
    3. Political risks
    4. Market risks

v. Simulations to Identify Performance Risks:

- Statistics based simulation technique are useful in situations where the performance of the system is a major risk that has to be anticipated and planned for.

vi. Decision tree:

- Risk often happen in series or in a cascading fashion.
- Decision trees present pictorial way to represent risks.
- They are used in conjunction with probabilities assigned to various anticipation events to qualify risks.
- Since decision trees are useful not just for identifying but also for quantifying risks.

2. Risk Quantification:

- There are 2 dimensions for quantifying risks-
  - **i.** Probability risk
  - **ii.** Impact risk

i. Probability of the risk:

- The higher the probability of a risk , the more aware we should be of the risk and plan an alternate strategy.
- We should be able to assess the probability by using historical data or simulations.

ii. Impact of the risk:

- If the risks turn into a reality what would be the impact?
- The impact may be in terms of lost hours of effort or the need to buy extra hardware or the loss of customer goodwill.
- It is customary to quantify the impact by the monetary value i.e. in dollar terms.
- The net effect of the risk is measured as risk exposure which is defined as the product of the probability of the risk and the dollar-impact if the risk becomes a reality.

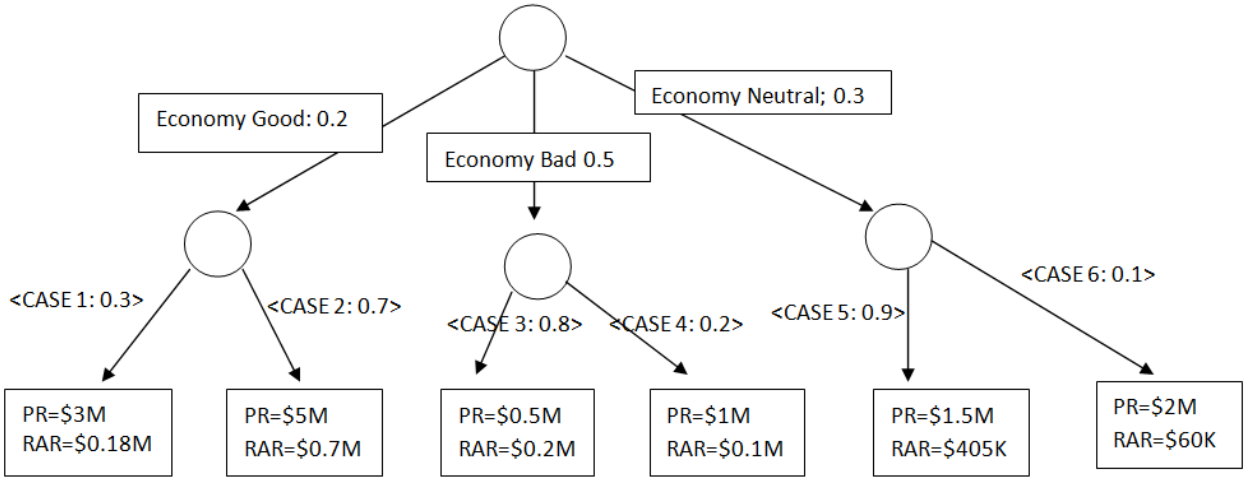
Risk Exposure= Probability of Risk * Dollar Impact of Risk

Tandem:

- Two people or pieces of equipment that work together to achieve a result.

e.g.: Suppose we are developing a s/w product whose volume depends upon the state of the economy. We hired consultants to advice us and they come back with the following predications on the state of the economy in the next year.

- It would be really good and booming with a probability of 0.2.
- It would be really bad with a probability of 0.5.
- It would be in a neutral state with a probability of 0.3.



PR=Project Revenue **RAR= Risk Adjusted Revenue**

Risk Monitoring:

- It would be advantageous to predict its arrival by watching the possible symptoms.
- Thus, associated with each risk we should have some indicators to portend the actual manifestation of the risk.
- We have the mechanisms in place to detect such symptoms and trigger corrective strategies also called the mitigation strategies.
- This process of risk monitoring is not a discrete, one time activity but should ideally be a continuous process.

3. Risk Mitigation:

- Mitigation is a possible means of minimising or even avoiding the impact of a risk.
- Risk mitigation usually comprises of identifying alternative strategies in the event of a risk becoming an actuality.

Metrics in Risk Management:

1. How often do the risks that were actually predicted turn into reality?

- Perhaps, if the risks do not turn into reality, it may indicate that these are red herrings rather than real risks.
- A root cause could be checklists used to predict the risks are not kept current.
- It is also important that this metric does not steer an organisation into complacency where planning is concerned.

2. Was the impact of the risks as severe as originally made out to be?

- This again points to the ability of the project management team to asses the impact of risks.

3. How many risks keep coming up repeatedly?

- This point to the inability of the management to learn from previous mistakes.

4. **How do the actual risks faced in a project differ from the anticipated risks (regardless of the impact)?**
    - This gives an indication of how good the anticipatory capabilities of the organisation are.
    - If hitherto unidentified risks are actually encountered then it is important to feed this back into organizational learning so that future projects can make use of this of knowledge and plan accordingly.

Module-III

Project Management Process & Activities:

✸ Umbrella Activity:

- Umbrella activity takes place through the life of a project.
  e.g.: Configuration management and metrics are carried out to find out whether the project is in the requirements phase or the implementation phase.

✸ In-stream Activity:

- Instream Activity take place sequentially & only at some specific times during the project life cycle.
  e.g.: Project initialization is done only at the beginning of the project. Project closure that will done at the end of the project.

The Activity During Project Initialization:

✸ Project initialization is an activity that marks the formal start of a project.

- At this stage the senior manager also identify a project manager who is responsible for achieving high level goal.
- In this section, we go into the details of some of the important activities carried out during project initialization phase:
  1. Management team building
  2. Scope & very high level work division agreement
  3. Decision of management reporting
  4. Involvement of infrastructure or support group
  5. Team formation
  6. Project kick off meeting

1. Management Team Building:

✸ The identified local managers have a face to face meeting for the 1st few days.

- This meeting should take place at a location where a large part of team is currently residing.
- Such a meeting serves for the following purposes:
  - **i.** It enables the team members to understand one another & to build a cohesive management team.
  - **ii.** Minimize the impact of cultural & language barriers.
  - **iii.** It is a sign of commitment.

2. Scope & High Level Division Element:

✸ When the local manager meets during project initialization, one of the goals to be achieved is that they should clearly understand the overall goal of the project, role & responsibilities of each manger & his team.

- The purpose of scope review during the project initialization include the following:
  - **a.** Verify that the scope is clearly documented & understand by all the members.
  - **b.** Identify any impossible constraint & altering the management on these.
  - **c.** Doing some initial risk analysis & mitigation planning.
- After the scope of the work to be done is review work allocation across the different sides. Each agreed upon at a very high level.
- The high level work allocation serves as the basis on which the work break down structure evolves.

3. Decision of Management Reporting:

- The project initialization or initiation is the time when some of the expecting setting for status reporting.
- Status reporting means:
  - **a.** The progress & action
  - **b.** The action item to completion
  - **c.** Time period or resolution
  - **d.** Recipient of the report
  - **e.** Level of the details, content & formal of status report.
  - **f.** Communication between various group conference calls.

4. Involvement of Infrastructure or Support Group:

- A project group doesn't operate in isolation for success it require the active participate, support & co-operation of other infrastructure group.
  - **The human research group:** responsible for hiring, training & working out policy for retention.
  - **Facility management group:** responsible for allocating & maintaining common facility like office space.
  - **Administration group:** responsible for travel.
  - **H/w infrastructure group:** manages all the machine, networks & s/w.
  - **Finance:** group for budgeting, case flow management.
  - **Corporate Quality group:** for aligning / adapting the organisational processes.
  - **Senior management:** for providing the overall support and commitment needed.
- During the project initialization phase. It is important that the expectations from each of the above group are clearly conveyed. So that they can plan & optimized their workload & procedure accordingly.

5. Team Formation:

- The final activity of project initiation is the formation of the complete team.

6. Project Kick Off Meeting:

- The project kick-off meeting is formally attended by all the team members so that everyone has a common understanding of what is expected from them.

- It can be viewed as a sign off meeting.
- The minutes of this meeting are recorded & circulated to all the team members this is known as kick off minute meeting.

Outputs, Quality Records & Completion Criteria for the Project Initiation Phase:

- Project initiation marks beginning of the project.
- At the beginning of the project. It is a very high level understanding of requirement of project & demand.
- Some of the characteristics of successful project initiation are:
  1. The requirements are understood very well.
  2. The communication between the management team members & the engineers working on the project are very good.
  3. Communications between various groups or teams that are geographically distributed are very good.
  4. The requirement specifications are accurate.

O/P Quality Record & Completion Criteria:

1. A statement of understanding of what the requirements & constants of the project are.
2. A statement of understanding of the requirements from the infrastructure or support group.
3. Indentified set of people to work on the project.

Project Planning & Tracking:

- Project planning consists of the 5 interrelated piece of things:
  1. What part?
  2. What cost?
  3. When part schedule?
  4. How part processes?
  5. By whom part?

1. **The What part(Contents):**

- For a product what will translate into what feature to deliver on what platform & in what environment.
- The what refers to what is the service we going to provide.(no of people available online or offline/ on site or off site)

2. **The What Cost (Money, Quality, Performance):**

- The what cost part of project planning is about understanding the constraint clearly & accounting for them in the project plan.

3. **The When part (Schedule):**

- When part of the project planning is about timeline of when we will deliver what part of the project or the product.

4. The How part (Process):

- The project plan should specify what processes are to be followed. How they are different from the processes. Currently being used in the organisation. How to reuse the existing process.

5. The By Whom part:

- By whom part of the project plan layout the roles & responsibilities of various members & what is expected of each one of them.
- Project planning is not really a onetime activity. The project manager is always in the planning mode. He set up an initial plan based on the above 5 dimensions. This is an umbrella activity instead of instream activity.

What Part of a Project Plan:

- When starting a project, one stat week some statements of what are the requirements.
- One also starts with a set of requirements. The requirements can be something like the system should be user friendly.
- When the project is planned properly a haziness that surrounds the implied requirements clears. This is done by using the work breakdown structure.

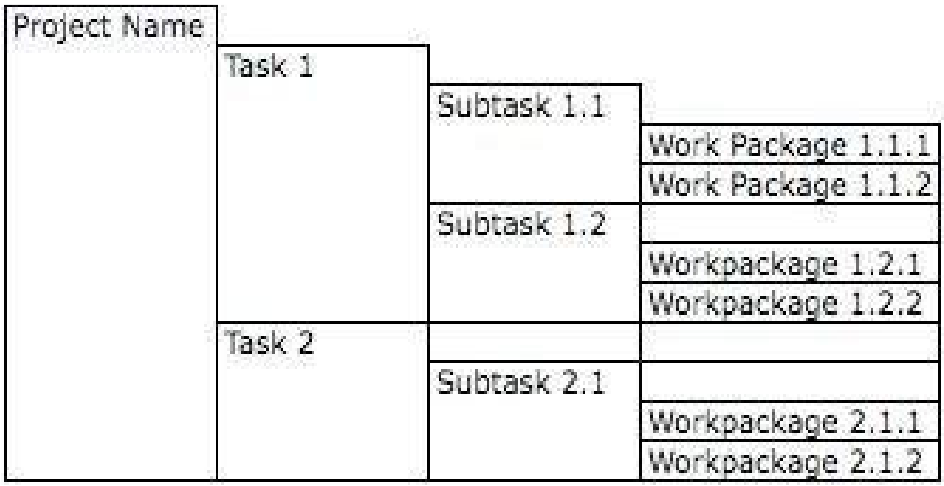
Work Breakdown Structure (WBS):

- Work breakdown structure is the decomposition (dividing) of the project into smaller & manageable (work breakdown unit)/ WBS unit.
- Usually, the project managers use this method for simplifying the project execution. In WBS, much larger tasks are broken down to manageable chunks of work. These chunks can be easily supervised and estimated.
- WBS is not restricted to a specific field when it comes to application. This methodology can be used for any type of project management.
- WBS unit should satisfy the following criteria:
  1. Each WBS unit has a clear outcome.
  2. The outcome has a direct relationship to achieving the overall project goal.
  3. Each WBS unit has a single point of accountability.
  4. Each WBS unit is something that can be tracked & monitored as an unit.
  5. Each WBS unit has a well defined interface to other WBS units.

Construction of a WBS:

- Identifying the main deliverables of a project is the starting point for deriving a work breakdown structure.
- This important step is usually done by the project managers and the subject matter experts (SMEs) involved in the project. Once this step is completed, the subject matter experts start breaking down the high-level tasks into smaller chunks of work.

- In the process of breaking down the tasks, one can break them down into different levels of detail. One can detail a high-level task into ten sub-tasks while another can detail the same high-level task into 20 sub-tasks.
- Therefore, there is no hard and fast rule on how you should breakdown a task in WBS. Rather, the level of breakdown is a matter of the project type and the management style followed for the project.
- In general, there are a few "rules" used for determining the smallest task chunk. In "two weeks" rule, nothing is broken down smaller than two weeks worth of work.
- This means, the smallest task of the WBS is at least two-week long. 8/80 is another rule used when creating a WBS. This rule implies that no task should be smaller than 8 hours of work and should not be larger than 80 hours of work.
- One can use many forms to display their WBS. Some use tree structure to illustrate the WBS, while others use lists and tables. Outlining is one of the easiest ways of representing a WBS.
- Following example is an outlined WBS:

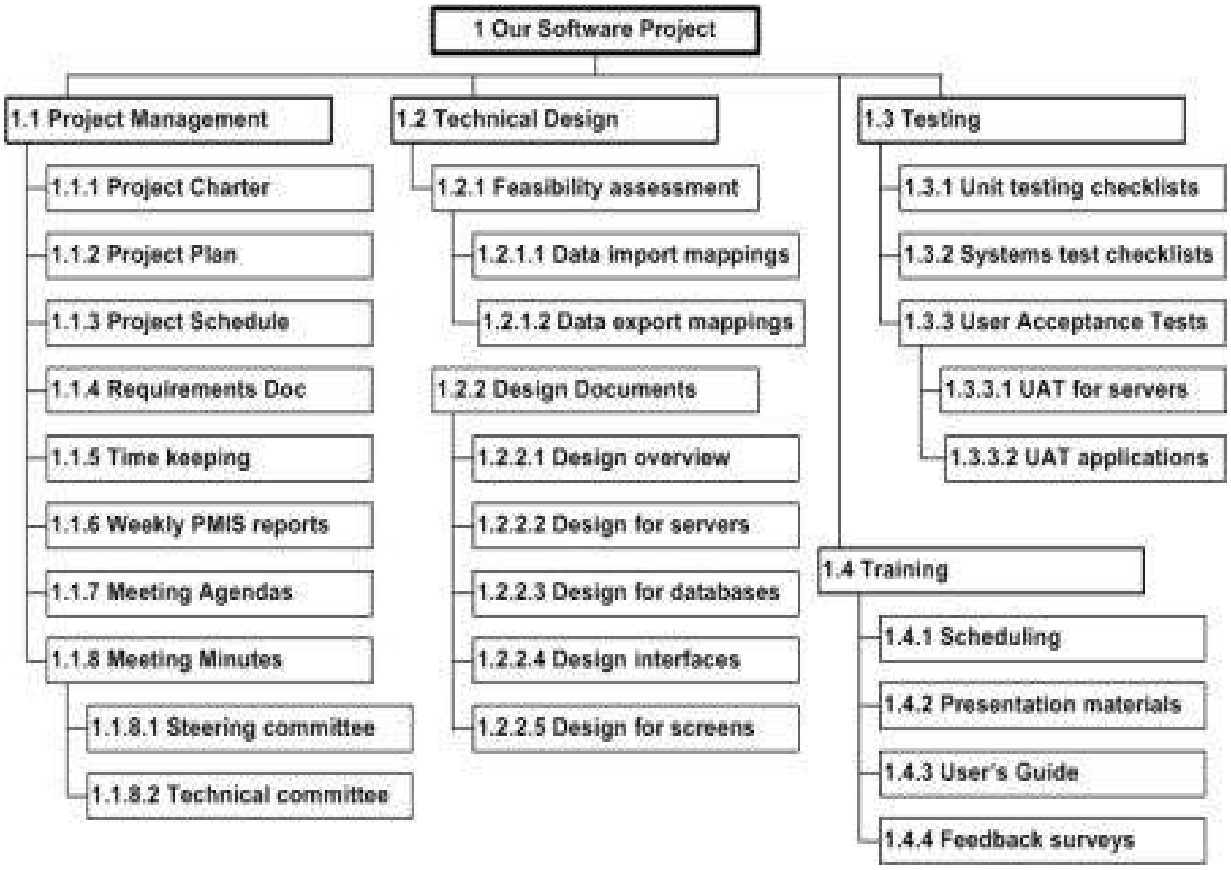


- There are many design goals for WBS. Some important goals are as follows:
  1. Giving visibility to important work efforts.
  2. Giving visibility to risky work efforts.
  3. Illustrate the correlation between the activities and deliverables.
  4. Show clear ownership by task leaders.

WBS Diagram:

- In a WBS diagram, the project scope is graphically expressed. Usually the diagram starts with a graphic object or a box at the top, which represents the entire project. Then, there are sub-components under the box.
- These boxes represent the deliverables of the project. Under each deliverable, there are sub-elements listed. These sub-elements are the activities that should be performed in order to achieve the deliverables.

- Although most of the WBS diagrams are designed based on the deliveries, some WBS are created based on the project phases.
- Usually, information technology projects are perfectly fit into WBS model.
- Therefore, almost all information technology projects make use of WBS.
- In addition to the general use of WBS, there is specific objective for deriving a WBS as well.
- WBS is the input for Gantt charts, a tool that is used for project management purpose.
- Gantt chart is used for tracking the progression of the tasks derived by WBS.
- Following is a sample WBS diagram:



Conclusion:

- The efficiency of a work breakdown structure can determine the success of a project.
- The WBS provides the foundation for all project management work, including, planning, cost and effort estimation, resource allocation, and scheduling.
- Therefore, one should take creating WBS as a critical step in the process of project management.

Activity specific to project tracking:

- Project tracking ensures that the project is progressing or progress as per the plan.
- If there is any difference to the plan the appropriate mid correction will be done.
- The specific the action or activity that increase the effectiveness of project tracking include:
    1. Status reporting
    2. Communication
    3. SPMP(s/w project Management plan) update

1. Status Reporting:

- A Status report that tracks the progress of the project is given on a periodic basic (daily/weekly etc.)a against a determine plan.
- The key values or e-issues for effective status reporting are:
    i. What should be the level of detail in status report?
    ii. Who should get the status report?
    iii. What should be done with the status report?
    iv. What should be the periodicity of status report?

i. What should be the level of detail in status report:

- The status report is generally likely to be read & used by high level management.
- Thus the information to be given in the report should have some degree of abstraction.
- A very simple format of status report shown below.

A Simple & Effective Status Report Format:

STATUS REPORT FOR THE PERIOD <FROM DATA> TO <TO DATE>
REPORTED BY: <REPORTING PERSON>
RED ALERT
ACTIVITIES PLANNED AND ACHIEVED FOR THIS PERIOD
ACTIVITIES PLANNED FOR THIS PERIOD BUT NOT ACHIEVED ,WITH REASON
ACTIVITIES NOT PLANNED FOR THISPERIOD BUT CARRIED OUT
ACTVITIES PLANNED FOR THIS NEXT PERIOD.

ii. Who should get the status report:

- It is important to ensure that the information present in the status report is not an ambiguous nature & i.e reaches the right person without any delay.
- The status report generally goes to the direct manager.

iii. What should be done with the status report:

- The project manager look at the Status report, observe the deviation from the plan & does any immediate service.

iv. What should be the periodicity of status report:

- The periodicity of the status report varies with the duration of the project.
- In general there are 2 factors that drive the frequency of status reporting:
  1. Project Duration
  2. Project Critically

2. Communication:

- Different types of communication are:-
  - **i.** One to One.
  - **ii.** Meeting/group meeting.
  - **iii.** Periodic conferencing among the team in various locations.
  - **iv.** Non work related outing.

i. One to One:

- The project manager has to face any team member, addressed in a timely & certifactory manner.
- The project manager should plan to have weekly one to one with each of team member.

ii. Group meeting:

- In project team there is a lot of inter dependency between work of various team members.
- There are potential conflicts their interest.
- A periodic meeting of the group has been held to get all the members at the same time.

iii. Periodic conferencing among the team in various locations:

- One of the biggest challenges of geographically distributed Project team is that people don't get to see each other physically very often.
- By using video conferencing the effectiveness of communication within the team increases.

iv. Non work related outing:

- Another very powerful way to accomplish the bonding is to have non-work related outing like picnics or offsite.
- A project manager should allocate some budget for this.
- This provides sufficient time and opportunity for interaction; it would have served the purpose.
- By organizing group games like cricket or volleyball it can increase the bonding and co-operation between the team members.

Project Closure:

- Closure refers to conclusion of a project or some logical part of the project.

Effective Closure Process:

- For a closure process to be effective:
  1. The meeting should have a point of focus & this focus should be on-learning.
  2. Each member should come prepared.
  3. Each member should get an opportunity to voice his view.

Issue That Get Discussed During Closure Process:

1. **What were the goals that we set out to achieve? :**
   - This is the 1st thing. The 1st deliverable of any project is ensuring customer satisfaction by meeting these entire requirements.
2. **How effective were the in process metric? :**
   - The in process metric truly reflects the end goal.
3. **If there were instances of over achievement or under achievement, what are the root causes? :**
   - It is important to do analysis for both underachievement & over achievement.
4. **Was our estimate of size in the right ball park? :**
   - If the size of the final work product was found to be different from originally estimated size.
   - Then it probably implies that an inappropriate size estimation method was used.
   - Hence, the root cause of size variation must be find out and the estimation procedures corrected for future use.
5. **Was our estimation of effort correct? :**
   - If the effort is actually expanded on the project is substantially disproportionate to the estimate effort then we should examine the means by which the efforts were originally estimated.
6. **What did we accidentally do write that we could legitimise? :**
   - When we execute a project, there are times when we try out new things accidentally (consciously or unconsciously). Some of the experiments or things done accidentally turned out to be great success.
7. **What did we gain from the system or environment? :**
   - Every project group gain some extend by the existing environment or system.
8. **What are the mistakes that we need not have made as they were under our control? :**
   - All of us do some mistake.
   - Some mistakes are under our control & some are not.
   - It is important to identify the once that are under our control & ensures that don't do such type of mistake.

9. What were the factors in the environment that we would like to change? :

- There may be need to change the organization Structure, operating procedure or infra structure. Each of these involves cost & may have some benefits.

10. Was our estimation of the H/W requirement correct? :

- Sometimes people working on s/w project are much higher than the actual requirements?
- During closure time some analysis should be done to check the size of the H/W is appropriate.

11. Would the use of any tools have made our life easier? :

- During project closure any experience on the selection & use appropriate tools.

12. Was the use of communication channel appropriate? :

- It is important to have a healthy balance between flexibility & Structured communication & processes.

13. What was the employee satisfaction is carried out this project? :

- The organizational goals & individuals satisfaction should be aligned for the success.

14. How many of the originally anticipated risk become realities? :

- This information is needed so those future projects performed are more realistic risk analysis.

15. Was the distribution of work done fairly among the geographically distributed team? :

- The distribution of work done properly among the geographical distributed team.

16. What changes did we have to make the organizational process to make them applicable to this project? :

- This point means which new project or processes were added for this project.

17. Did we get & use customer feedback appropriately? :

- In this point we discuss the channel through which the customers give their feedback.

Metrices for Project Closure:

Metrices	Significant
1. Size variance	How much did the estimate size of the work Products vary from its actual size.
2. Effort variance	How much did the estimated effort on the project Vary from actual effort.
3. Schedule variance	How much did the estimated schedule on the project Vary From the actual schedule.
4. Number of changes made to the project	Process changes could be any of the following: (i)Change in the method of estimation of size or effort. (ii)Change in the organizational structure. (iii)Change in the operational procedure. (iv)Change in the work load distribution.
5. S/W tool enhancement	If new tooling is supported/suggested changes are that this Would lead to higher productivity.
6. Employee Satisfaction	It can be estimated by formal & informal survey.

Module-IV

Engineering Activities in Project Life Cycle:

S/W Requirement Gathering:

1. I/p & start criteria.
2. Dimension step.
3. O/p & quality method.
4. Skilled set.
5. Challenges.

1. I/P & Start Criteria:

- In this steps the identification of components of s/w that are needed to develop a project.
- Understanding the detail of s/w component known as s/w requirement gathering.
- During requirement gathering phase the s/w development organization understand what is needed for the product to satisfy the need of the customer.

2. Dimension Step:

- There are 4 dimensions that are need for requirement gathering
  i. Responsibilities
  ii. Current system requirement
  iii. Target
  iv. Ongoing needs

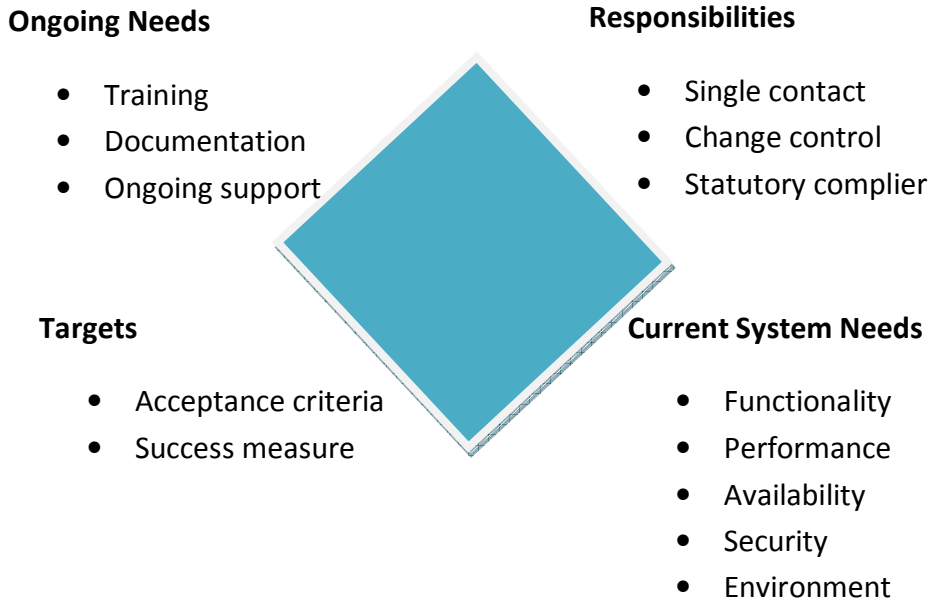


Fig: Dimension of Requirement

i. Responsibility:

- The Responsibility are taken from 4 points of view:

a) Single contact
b) Change control
c) Statutory compliance
d) Organizational structure

ii. Current System Needs:

- The 2nd dimension of requirements management is the current system needs 5 points of view:

a) Functional requirement
b) Performance
c) Availability
d) Security
e) environment definition

a) Functional Requirements:

- The functionality past of the requirement address issues like what is expected from the system & how the system would satisfy the business need of the customer.

b) Performance Requirement:

- It can be measured by the following parameters:

- Data storage requirements
- Maximum no. of users expected on the system
- Excepted response time for transaction
- Throughput that the system most satisfy
- Growth rate in future for any of the above

c) Availability:

- Availability needs are about the expectation of the various components.

d) Security:

- Security determines who can access which part of the system.

e) Environmental definition:

- As the choice of the h/w or s/w platform greatly influence design and future development.

iii. Target:

1. Acceptance criteria
2. Success measure

1. Acceptance criteria:

- When a system is developed how can one conclude that it meets the need of the customer.
- Acceptance criteria are meant to serve this purpose.

2. Success Measure:

- Success measures denote the criteria under with the project can be deemed successful.

iv. Ongoing Needs:

1. Training.
2. Documentation.
3. Ongoing support.

1. Training:

- Once a project/product is developed its user needs to be trained how to use the product.
- Different levels of training are used for different people.

2. Documentation:

- Every product require documentation & to which it is needed depends on the complexity of the product.

Ex:

i. User manual.
ii. Design a internal documentation.
iii. Installation guide.
iv. Online help.

3. Ongoing support:

- Once the system is developed in the customer site there would be need for ongoing support.

3. O/P & Quality Record:

- The o/p from the requirement gathering process is requirement specification document. There are various formats for the requirement specification document.
- The main quality record that need to be captured during the requirement gathering phase include:
  a. Minutes of the various meeting held to discuss requirements.
  b. Any correspondence carried out to clarify or solve conflict in the requirements.
  c. Change request % there impact signed by the appropriate authority.
- These quality records ensure that the right requirements captured from the right person.

4. Skill Set:

- The requirement gathering phase is characterized by high customer interaction based on this factor some of the essential skill set needed for gathering the requirement.
- The skill set needed for requirement gathering are:
  1. Ability to look at the requirements from the customer point of view.
  2. Domain expertise
  3. Technology awareness
  4. Strong inter personal skill
  5. Strong negotiation skill
  6. Ability to tolerate ambiguity
  7. Strong communication skill

5. Challenges during Requirement Gathering Phase:

i. Incomplete & changing requirement:

- The requirement gatherings at the start of a project are often incomplete.
- The question arises whether one should wait for all requirements thus the challenges lies in being able to track & control change to requirement.

ii. <u>Conflicting requirement</u>

- In a complex system involving several users, conflicting requirements should be expected & planned for.
- When these conflicts are detected either during the requirement phase or need for changes is felt.
- They have to brought of the notice of the single point of contact.

iii. <u>Straighter & requirement:</u>

- During requirement management phase the goal should be knock to be leave any of the requirement as implied.
- One should be more objective & measure of success of the project against straight requirement.

iv. Technology changing the customers perception of what constitute a requirement.

v. <u>Confusing the 'how' with the 'what':</u>

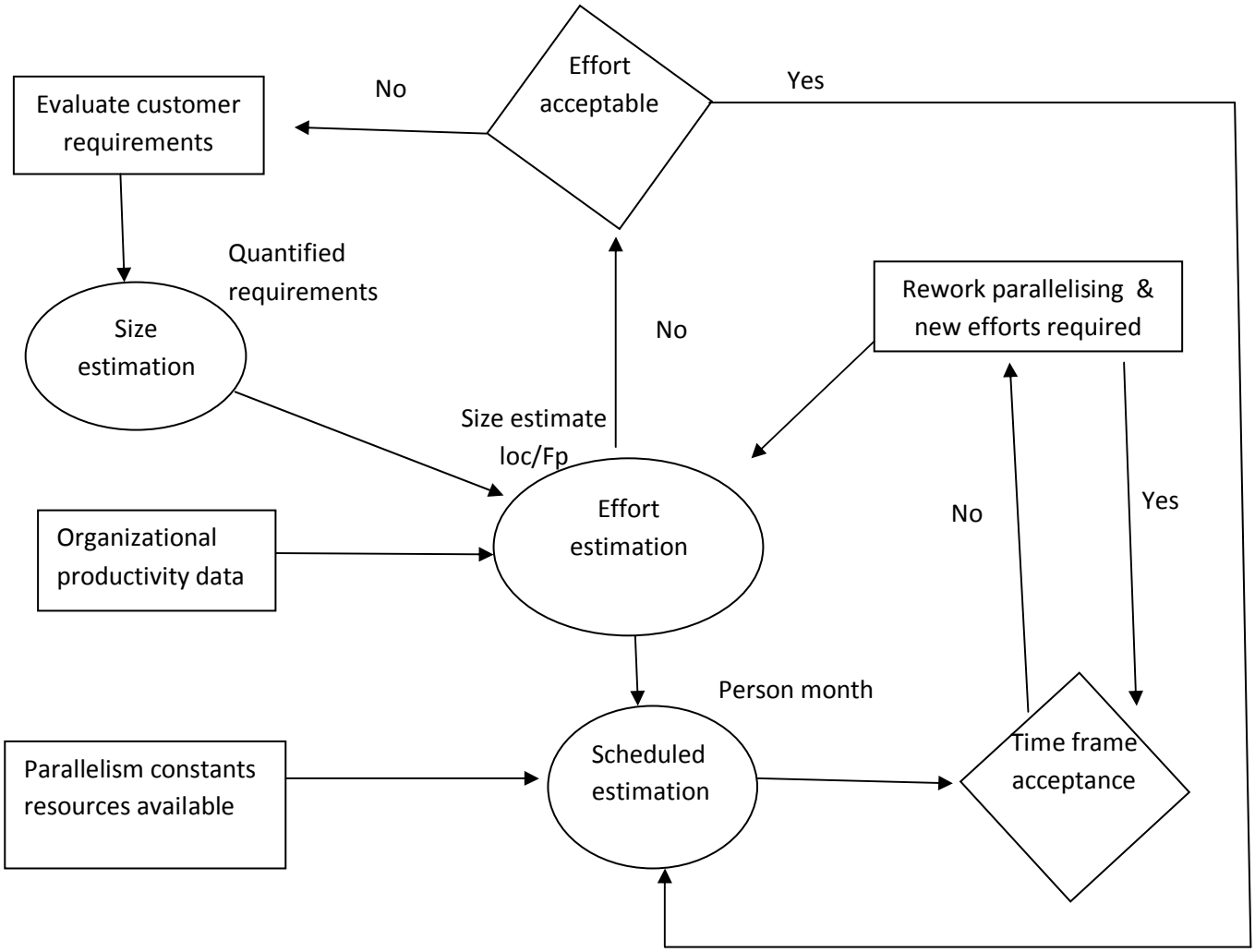
- During the requirement phase only the 'what' should be captured thus' how' gets confused with 'what'. This is something causes requirements to be captured.

vi. <u>Unwillingness the sign of requirements:</u>

- Some customer may be unfeeling to sign of requirement to allow themselves the roof further negotiation down the light.

Schedule:

- Schedule estimate is the calendar time it would take to accomplish the efforts taking into account the resource of an organization.
- The size, effort & schedule get translated into cost estimate. The cost estimation is calculated by dollar value($) cost estimation in dollar value is given by
- **Cost estimation= # of person month * dollar value per person month.**



Attribute	Size estimate	Effort estimate	Schedule estimate
What is measure?	Actual ground to be covered	Effort it would take to cover the ground	Time it would take to cover the ground
Unit of measure	Loc/Fp (Line of code/Function point)	Person month	Calendar month
Primary driver	Complexity of the product (Features to be covered)	Productivity data	Parallelism constraint resource available
Dependency	Depend on customer requirement	Depend on customer requirement organizational productivity & reuse opportunity	Schedule depends on size & effort estimate factor
Customer use	To validate the organization understanding of the requirement	To complain different vender for cost become cost is tied directly with effort	To compare different vendor for deliver date

Estimation Methodology:

- The 1st possibility is to use past experience.
  - One can talk to people who have done similar project in the past within the organization such experience can then be extract plotted. (input for the current project)
  - The main challenge in this approach is how does one define similar project know to projects are identical but they have some similarity in part.
  - A refer may of this model is the divide & conquer approach.

1. A project is decomposed into smaller & manageable components.

2. The requirements of these smaller components are studied in more details.

3. When The project is broken down into components, these components can fall into one of the following 3 categories:

a. Components those are identical to the already developed components.

Ex- Components like screen handler, error handler etc.

In this category such components are reused & don't need to be redeveloped.

b. Components that can be adopted from the already developed components with slight modification.

c. Components that are completely new & need to be built from new resources.

4. The estimation for each of the above 3 categories of components are worked out separately.

- For the components that are in the 1st category there is no impact (We already know the size & the effort)
- For the 2nd category of the components with minor modification.
- For the components that are entirely new very rough estimate are made initially.

5. Each of these estimates are given some contingency factors which are essential for applying divide & conquer approach using the historical data. Each project has some unknown that have to be accounted for.

- ✸ One should follow some general guidelines to estimate to contingency factor.

a. Apply a fire contingency factor for estimate during the early stages of evolution.

b. In a new area, apply a fire contingency factor .

c. For project that are either difficult to find or in hot demand contingency factor is high.

d. Projects that are entirely new applied a higher contingency factor.

6. An aggregate estimate of the project is then put together by adding the estimate individual components.

Formal Model for Size Estimation:

There are 2 models i.e. Loc (Line of code) & Fp (Function point)

LOC:

- In a s/w project the product is build by lines of codes in a programming language.
- The loc is a possible major for the size of the product.
- Several reasons for usage lines of code are as follows-

1. Easily understood
2. Easily measurable
3. Language dependency
4. Need for some common ground rules
5. Applicability to the entire life cycle.
6. Penalises compactness

FP:

The application features of Fp are divided into function under the heads of i/p, o/p, interfaces, and external data files & enquire.

Ex:-

- Enquires & inputs are less complex than data file. The o/p & the interface are fall between these two.
- The weighted average of function. (number of function of each type multiply by weight for that function type gives an initial estimate for their size.)