5) Software Project Management

Page No. 79
Date

Introduction to Software Project Management and its need :

Software Project management is a necessary activity. It involves :

- Planning
- Monitoring and control of :
  1) People
  2) Process
  3) Events that occur during development of software.

The process of software management includes thousands of activities which are linked with each other. Without project management approach or technique it is impossible to build or develop a sucessful and quality software product.

Project management is needed because software development is always subject to budget and schedule constraints that are set by the organisation developing the software.

Management Spectrum :-

Effective project management focuses of four P's :

① People ② Product ③ Process ④ Project

The order is not arbitrary.

The people:-

The cultivation of motivated, highly skilled software people is very important.

People Capability Maturity Model (P-CMM) defines the following key practice areas for software people:

- Staffing
- Communication & Coordination
- Work environment
- Training
- Compensation
- Competency analysis and development
- Career development
- Workgroup development
- Team Culture development and others

The product:-

Before a project can be planned:

1. Product objectives and scope should be established
2. Alternative solutions should be considered
3. Technical and management constraints should be identified.

Software developer and other stakeholders must meet to define product objectives and scope:-

1. Objectives identify the overall goals for the product without considering how these goals will be achieved.
2. Scope identifies the primary data, functions, and behaviours that characterize the product.

The process :-

A software process provides the framework from which a comprehensive plan for the software development can be established. A small number of framework activities are applicable to all software projects, regardless of their size and complexity.

A number of different task sets — tasks, milestones, work products and quality assurance points — enable the framework activities to be adapted to the characteristics of software projects and requirements of the project teams. Finally, umbrella activities overlay the process model

The project :-

To avoid project faliure :

① Must avoid a common set of warning signs.
② Understand the critical sucess products that lead to good project management
③ Develop a common sense approach for planning monitoring and controlling the project.

Why software projects fail :-

① Unrealistic deadlines established
② Changing customer requirements
③ Underestimate of effort.
④ Predictable and unpredictable risks
⑤ Technical difficulties
⑥ Miscommunication among the project members
⑦ Faliure in project management.

People :-

① Stakeholders :- Software process and project is contains stake holders who can be categorised into five types :

- Senior managers :- Define business issues that often have significant influence on project
- Project (technical) managers :- Who must plan, motivate, organise, and control the practitioners who do software work.
- Practitioners :- Who deliver the technical skill that are necessary to engineer a product
- Customers :- Who specify the requirements for the software and other stake holders who have intrest in the outcome.
- End users :- Who interact with the software once it released for production use.

② Team Leaders :-

MOI model of leadership :-

① Motivation :- The ability to encourage technical people to produce their best ability

② Organisation :- The ability to mould existing processes that will enable the initial concept to be translated into a final product

③ Ideas or Innovation :- The ability to encourage people to create and feel creative even when they must work within bounds established for a particular software product or application.

Date

Successful project leaders apply a problem solving management style i.e a software project manager should concentrate on understanding the problem to be solved, managing the flow of ideas, and letting everyone on the team know that quality is very important.

③ Software Teams:- There are many human organisational structures for software development. Organisational structure cannot be easily modified. Organisation changes are not within the software manager's scope but the organisation of people who are involved are within the scope of software manager.

Seven project factors that to be considered while planning the structure of software team
① Difficulty of the problem to be solved
② "Size" of the resultant program(s) in lines of code or function points.
③ Time that the team will stay together.
④ Degree to which program can be modularized
⑤ Required quality and reliability of the system to be built.
⑥ Rigidity of delivery date
⑦ Degree of sociability (communication) required for the project

organisational paradigms" for software engineering teams.

① Closed paradigm
② Random paradigm
③ Open paradigm
④ Synchronous paradigm

To achieve a high performance team:

① Team members must have trust in one another
② The distribution of skills must be appropriate to the problem.
③ Mavericks have to be excluded from a team, if team cohesiveness is to be maintained.

Factors that "foster a potentially toxic team environment":

① A frenzied work atmosphere
② High frustration that causes friction among team members.
③ A "fragmented or poorly coordinated" software process
④ An unclear definition of roles of the software team
⑤ "Continuous and repeated exposure to failure."

④ Agile team : Agile philosophy:

* Encourage customer satisfaction
* Early incremental delivery of software
* Small highly motivated project teams
* Informal methods
* Minimal software engineering work products
* Overall development simplicity
* Adopt many of the characteristics of sucessful software project.

Coordination and Communication issues:-
There are many reasons that software projects get into trouble:
* Scale of many development efforts is large, leading to complexity, confusion and significant difficulties in coordinating team members
* Uncertainity is common, resulting in a continuing stream of changes that ratches the project team
* Interoperatability has become a key characteristic of many system. New software must communicate with existing software and conform to predefined constraints imposed by the system or product.

To deal with them effectively, we must establish effective methods for coordinating the people who do the work. To accomplish this, mechanisms for formal and informal communication among team members and between multiple teams must be established.
Formal communication is accomplished through writing, structured meetings and other relatively non-interactive and impersonal communication channels.
Informal communication is more personal. Members of a software team share ideas on an ad hoc basis, ask for help as problem arise, and interact with one another on daily basis.

Product:- It is very necessary for us to examine the product and the problem at the very beginning of the project. At a maximum, the scope of product must be established and bounded.

Software scope:- The first software project management activity is the determination of software scope. Scope is defined by answering the following questions:-

* Context:-
  Q) How does the software to be built fit into larger system, product or business context?
  Q) ~~Information~~ What constraints are imposed as a result of the context?
* Information Objective:-
  Q) What customer-visible data objects are produced as output from the software?
  Q) What data objects are required for input?
* Function and performance:-
  Q) What function does a software perform to transform input data into output?
  Q) ~~Are any~~ Are any special performance characteristics to be addressed?

The software project scope must be unambiguous and understandable at the management and technical levels.

A statement of software scope must be bounded

Problem decomposition :-

Problem decomposition, sometimes called partitioning or problem elaboration, is an activity that sits at the core of software requirements analysis. Decomposition is applied on two major areas :-

① The functionality and content (information) that must be delivered.

② The process that will be used to deliver it.

A complex problem is partitioned into smaller problems that are more manageable. This is the strategy that applies as project planning begins.

Process :-

We should select the appropriate process model that is appropriate for :

① The customers who have requested the product and the people who will do the work.

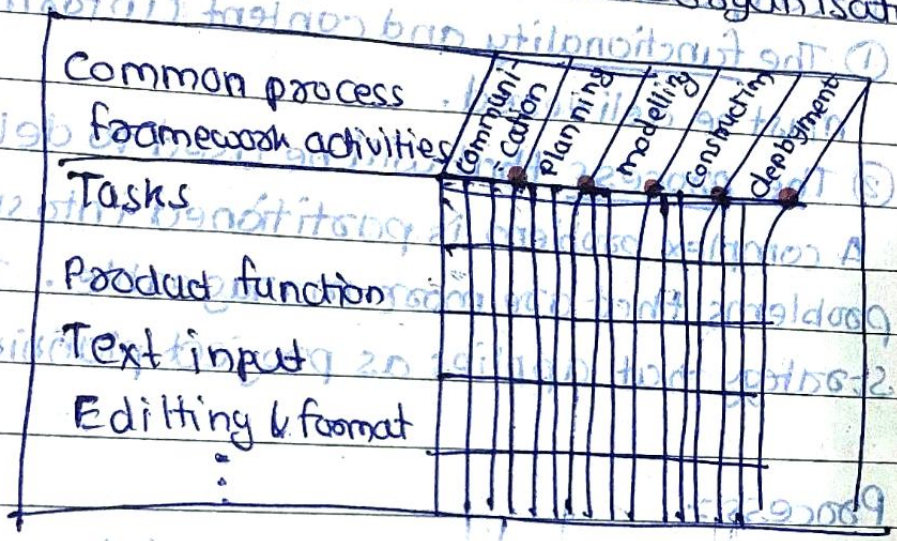
② The characteristics of the product itself

③ The project environment in which software team works.

When a process model has been selected, the team then defines a preliminary project plan based on the set of process framework activities.

Melding the Product and the process :-

Project planning begins with the melding of the products and the process. Each function to be engineered by our team must pass through the set of framework activities that have been defined by your software organisation.



Assume that the organisation has adopted the generic framework activities - communication, planning, modelling, construction & deployment. The team members who work on the product function will apply each of the framework activities to it.

Here, job of project manager is to estimate:

1. Resource requirements for each cell.
2. Start and end dates for tasks associated with each cell.
3. Work products to be produced as a consequence of each task.

Date

Process Decomposition :-

Software team have flexibility in choosing the software process model as per project.

* Small project that is familiar to the past might be accomplished using linear sequential approach.
* If the deadline is so tight that full functionality cannot reasonably be delivered, an incremental strategy must be best.
* Projects with other characteristics will lead to selection of other process models.
* A project decomposition commences when the project manager asks, "How do we accomplish this framework activity?"

Ex:- A small relatively simple project might require the following work tasks for communication activities:-

* Develop a list of clarification issues
* Meet with stakeholders to address clarification issues
* Jointly develop a statement of scope
* Review a statement of scope with all concerned
* Modify a statement of scope as required.

Consider a more complex project, which has a broader scope and more significant business impact:-

* Review the customer request
* Plan and schedule a formal, facilitated meeting with all stakeholders.
* Conduct research to specify the proposed solution and existing approaches.
* Prepare a "working document" and an agenda for the formal meeting.
* Conduct the meeting.
* Jointly develop mini-specs that reflect data, functional and behavioural features of the software.
* Review each mini-spec or use case for correctness, consistency and lack of ambiguity
* Assemble the mini-specs into a scoping document
* Review the scoping document or collection of use cases with all concerned.
* Modify the scoping document as use cases as required.

Project :- In order to manage a successful software project, we have to understand what can go wrong so the problems can be avoided.

10 signs indicating projects is in jeopardy :-

① Software people don't understand the customer need.
② The product scope is poorly defined.

③ Changes are managed poorly.
④ The chosen technology changes.
⑤ Business need change
⑥ Deadlines are unrealistic
⑦ Users are resistant
⑧ Sponsorship is lost
⑨ The project team lacks people with appropriate skills
⑩ Managers & practioners avoid best practices and lessons learned

Five part common-sense approach to software projects :-

① Start on the right foot :- This is accomplished by working hard to understand the problem and then setting realistic objectives and expectations for everyone who will be involved in the project.

② Maintain momentum :- To maintain momentum, the project manager must provide incentives to keep turnover of personnel to almost minimum, the team should emphasize quality in every task it performs, and senior management should do everything possible

③ Track progress :- For a software project, progress is tracked as work products are produced and approved as a part of quality assurance activity

④ Make smart decisions :- The decisions of the project manager and software team should be to "keep it simple."

⑤ Conduct a post mortem analysis :- Establish a consistent mechanism for extracting lessons learned for each project.

Project scheduling :-

Project scheduling provides details like start and end date of the project, milestones and tasks for the project. It specifies the resources like people, equipment etc. required to complete project and dependencies of task on project on each other. An appropriate project schedule prepared according to project plan not only aims to complete the project on time but help to avoid additional costs occurs when project is delayed.

Scheduling for software development projects can be viewed from two perspectives :

① An end date for release of a computer-based system that has already been established.

② Rough chronological bounds have been discussed but that the end date is set by the software engineering organisation.

Project scheduling activities :-

① Identify activity :- Identify the specific activities that must be performed to produce the various project deliverable.

② Identify activity dependencies :- Resources are allocated and estimating the number of work periods which will be needed to complete individual activities.

③ Allocate resources :- Resources are allocated and estimating the number of work periods which will be needed to complete individual activities

Date

④ Create Project charts :- Project activity charts are created to analysing activity sequences, activity durations, and resources requirements to create the project schedule

⑤ Allocate people to activities :- According to different activities people are attended to that activities.

Factors that delay project schedule :-

① An unrealistic deadline established by someone outside the software team and forced on managers and practitioners.

② Changing customer requirements that are not reflected in schedule changes.

③ An honest underestimate of the amount of effort

④ Predictable or unpredictable risks that were not considered when then project commenced.

⑤ Technical difficulties that could not have been foreseen in advance.

⑥ Human difficulties that could not have been foreseen in advance.

⑦ Miscommunication among project staff.

⑧ A faliure by project management to recognise

Principles of Project scheduling :-

① Compartmentalization :- The project should be compartmentalized into number of manageable activities and tasks.

② Interdependency :- The interdepency of each compartmentalized activity must be determined

③ Time allocation :- Each task ~~shou~~ to be scheduled must be allocated some number of work units.

④ Effort validation :-

⑤ Defined responsibilities :-

⑥ Defined outcomes :-

⑦ Defined milestones :-

Project Scheduling Techniques :-

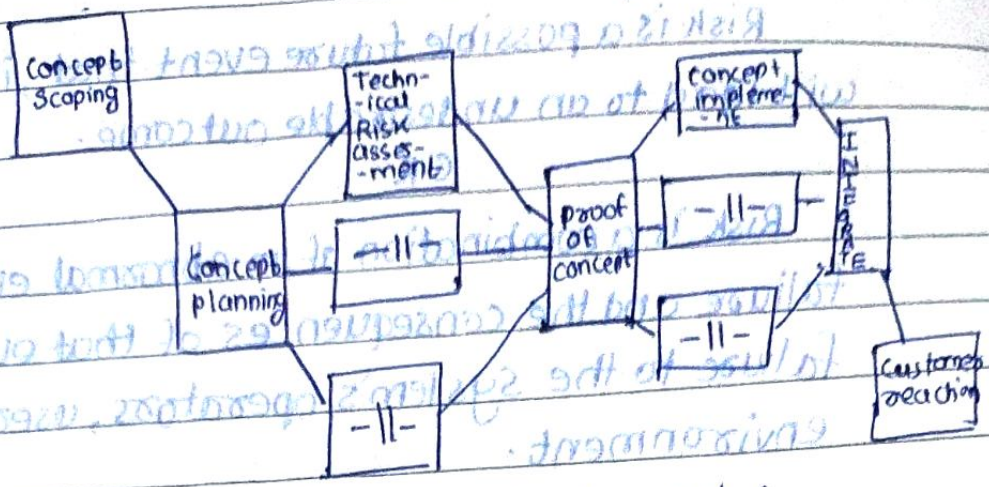
Gantt charts / Timeline charts :

When creating a software project schedule, we begin with a set of tasks (the work breakdown structure). If automated tools are used, the work breakdown is input as a task network or outline. Effort, duration and start date are then input for each task. In addition, tasks may be assigned to specific individuals.

As a consequence of this input, time-line chart also called as Gantt chart, is generated.

Concept of Task Network :-

The task network, also called an activity network, is a graphic representation of the task flow for a project. It is sometimes used as a mechanism through with task sequence and dependencies are input to an automated project scheduling tool.



Task N/W for concept development :

Figure shows schematic task network for a concept development project.

Ways of Project Tracking :-

Tracking can be accomplished in a number of different ways :-

1. Conducting periodic project status meetings in each team member reports progress & problems
2. Evaluating the results of all reviews conducted throughout the software engineering process.
3. Determining whether formal project milestones have been accomplished by the scheduled date.
4. Comparing the actual start date to the start date for each project tasks listed in the resource table.
5. Meeting informally with practitioners to obtain their subjective assesment of progress to date and problems to the horizon.
6. Using earned value analysis to assess progress quantatively.

Risk management :-

Risk is a possible future event that, if it occurs, will lead to an undesirable outcome.

OR

Risk is a combination of an abnormal event or faliure and the consequences of that event or faliure to the system's operators, users or environment.

Risk management is a systematic process, which focuses on identification, control and elimination of risk, which impacts the resources of the organisation and the project. Risk analysis and management are series of steps that help a software team to understand and manage uncertainty.

Seven principles of Risk management :-

1. Maintain a global perspective :-
2. Take a forward looking view :- think about all the risks that may arise in future.
3. Encourage open communication :-
4. Integrate :- Consideration of risks must be integrated into the software process.
5. Emphasize a continuous process :-
6. Develop a shared product vision :-
7. Encourage team work :-

Purpose of risk management :-

① Anticipate and ~~identy~~ identify risk
② Minimize the impact / damage / loss
③ Reduce the probability
④ Monitor risk areas for early detection
⑤ Ensure management awareness of risks.

Process is risk management :-

① Risk identification :- Identify project, product & business risks
② Risk analysis :- Access the likehood and consequence of the risk.
③ Risk planning :- Draws up plans to avoid or minimize the effects of the risk.
④ Risk monitoring :- Monitor the risks throughout the project.

Software Risks / Types :-

Risk always involves two characteristics :-

① Uncertainty :- the risk may or may not happen
② Loss :- if the risk becomes a reality, unwanted consequences or losses will occur.

Categories of risk :-

① Project risks :- It threatens project plan, the project schedule will slip & costs will increase. It identifies potential budgetary, schedule, personnel, resources, stakeholder and requirements problem and their impact on software project.

② Technical risks :- It threatens the quality and timeliness of the software to be produced. If these risks occur, the implementation may become difficult or impossible. They identify potential design, implementation, interface, verification and maintainance problems.

③ Business risks :- It threatened the viability of the software to be built. Its types are :

* Building an excellent product or system that no one really wants. (Market risk).
* Building a product that no longer fits into the overall business strategy for company. (Strategic risk).
* Building a product that the sale's force doesn't understand how to sell (sales risk).
* Losing support of senior management due to change in focus or a change in people. (Management risk).
* Losing budgetary or personal commitment (budget risks).

Another general categorisation of risks :-

① Known risks :- These are those that can be uncovered after careful evaluation of project plan, the business and technical environment.

② Predictable risks :- These are extrapolated from past experience.

③ Unpredictable risks :- These can occur but are extremely difficult to detect in advance.

Proactive and reactive strategies:-
Proactive strategies are interventions which are used on an ongoing basis in an attempt to reduce the likelihood of the risks.
Reactive strategies are interventions which are implemented once the risk occurs.
A considerably more intelligent strategy for risk management is to be proactive.

Risk assesment:-
Risk assesment involves finding out the quantity and quality of risk associated with a situation of known threat. It is the first and foremost step in a risk management process. The goal of risk assesment is to prioritize the risk so that attention and resources can be focused on more risky items.

Risk identification:- It is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, the project manager steps towards avoiding them.

Two types of risk:-
1. Generic risks:- These are a potential threat to every software project.
2. Product specific risks:- These can be identified only by those with a clear understanding of the technology, the people and the environment that is specific to the software.

One method for identifying risks is to create a risk item checklist :-

(1) Product size :- risks associated with overall software to be built or modified.

(2) Business impact :- risks associated with constraints imposed by management or market place.

(3) Stakeholder characteristics :- risks associated with sophistication of stakeholders and developers ability to communicate with stakeholders in a timely manner.

(4) Process definition :- risks associated with the degree to which the software has been defined and is followed by the development organisation.

(5) Development environment :- risks associated with the availability of quality tools to be used to build the product.

~~(6) Technology to be built :- risk associated with the availability and quality of tools to be used to build the product~~

~~(7)~~

(6) Technology to be built :- risk associated with complexity of system to be built and the "newness" of the technology that is packaged by the system.

(7) Staff size and ~~newness~~ experience :- risks associated with the overall technical and ~~project~~ experience of the software engineers who will do the work.

Date

Risk Analysis :-

Questions for accessing overall project risk :-

* Have top software and customer managers formally committed to the support of the project?
* Are end users enthusiastically committed to the project and the system/product to be built?
* Are requirements fully understood by the software engineering team and its customers?
* Have customers been fully involved in the definition of requirements?
* Do end users have realistic expectations?
* Does the software engineering team have the right mix of skills?
* Are project requirements stable?
* Does the project team have experience with the technology to be implemented?
* Is no. of people on project team are adequate to do the job?
* Do all customer/user constituencies agree on the importance of project and on the requirements of the system/product to be built?

If any one of these questions is answered negatively, mitigation, monitoring and management steps should be instituted without fail.

Risk components and drivers :-

① Performance risk :- The degree of uncertainty that the product will meet its requirements and be fit for its intended use.

② Cost risk :- The degree of uncertainty that the project budget will be maintained.

③ Support risk :- The degree of uncertainty that the resultant software will be easy to correct, adapt and enhance.

④ Schedule risk :- The degree of uncertainty that the project schedule will be maintained & that the product will be delivered on time.

Impact of each risk driver on the risk component is divided into four impact categories :-

* Negligible
* Marginal
* Critical
* Catastrophic

Risk Prioritization :-

Risk projection / prioritization attempts to rate each risk in two ways :

* Probability of occurence :- Likelihood or probability that the risk is real.
* Impact :- The consequences of the problems associated with the risk.

Four risk projection steps :-

① Establish a scale that reflects perceived likelihood of risks.

② Delineate the consequences of the risk.

③ Estimate the impact of the risk on the project and the product.

④ Access the overall accuracy of the risk projection so that there will be no misunderstandings.

Risk Mitigation, Monitoring & Management (RMMM)

An effective strategy for risk control must consider three issues:-

① Risk avoidance ② Risk monitoring

③ Risk management and contingency planning

If a software team uses a proactive approach, avoidance is always the best strategy. This is achieved by developing a plan for risk mitigation.

For ex:- High staff turnover is a project risk, it will have a high impact on project cost and schedule. To mitigate this risk, we can develop a strategy for reducing turnover:-

* Meet with staff and determine causes for turnover
* Mitigate the causes which are under our control before the project starts.
* Once the project starts, assume turnover will occur and develop techniques to ensure continuity when people leave.
* Organise project team so that information about each development activity is widely dispersed.
* Define work product standards and establish mechanisms to be sure that all documents are developed in a timely manner.
* Conduct peer reviews of all work.
* Assign back-up staff member for every critical technologist.

In case of staff turnover, the project manager can monitor following factors:

* The general attitude of team members based on project pressures.
* The degree to which the team has jelled.
* Interpersonal relationships among team members
* Potential problems with compensation & benefits
* The availability of jobs within the company and outside it.

A project manager should monitor the effectiveness of the risk mitigation steps.

Software Configuration Management (SCM):-

SCM is software configuration management also known as change management. The output of software process is information that is divided into three categories:-

* Computer programs
* Work products that describe the computer programs
* Data or content

◇ SCM is a set of activities designed to:

① Manage change by identifying that work products are likely to change.
② Establish relationships among them.
③ Defining mechanisms for managing different versions of these work products.
④ Controlling the changes imposed
⑤ Auditing and reporting on the changes made.

◇ SCM activities are developed to
① Identify change
② Control change
③ Ensure that change is properly implemented.
④ Report changes to others who may have interest.

Need of SCM :-

SCM is a set of management disciplines within the software engineering process to develop a baseline. When we build computer software, change happens; hence we need to manage it effectively.

/* Write the topics marked with ◇ */

Benefits of SCM :-

SCM benefits in four areas:

* Control
* Management
* Cost savings
* Quality

These 4 benefits are mapped to an organisation overall goals and objectives when the decisions are made to bring SCM tool in house.

Other benefits of SCM are :-

* Reduce redudant work
* Effective management of simultaneous updates
* Avoid configuration related problems.
* Facilitates team coordination
* Help in building managing tools used in builds
* It ensures that every defect has traceability back to its source.

SCM scenario:-

SCM operational scenario involves,

① Project manager, who is in charge of software project.

② Configuration manager, who is in charge of the CM procedures and policies.

③ Software engineers, who are responsible for developing and maintaining software product.

④ Customer, who uses the product.

① Project manager :- Goal is to ensure that software is developed with certain time frame. Managers monitors the progress of development and recognizes and reacts to the problems. This is done by generating and analysing reports about the status of the software and by performing reviews on the system.

② Configuration manager :- Goal is to ensure that procedures and policies for creating, changing and testing of code are followed, as well as to make information about project accesible. & To implement techniques for maintaining control over code changes.

③ Software engineers :- Goal is to work effectively without unnecessarily interfere with each other, but at the same time, they try to communicate and coordinate efficiently. They use tools that help build a consistent software product. They have their own workspace for creating, changing, testing & integrating code.

④ Customer :- Uses the product. Since the product is under CM control, the customer follows formal procedures for requesting changes and for indicating bugs in the product.

Elements of Configuration Management System :-

① Component elements :- Set of tools coupled within a file management system that enables access to and management of each software configuration item.

② Process elements :- Collection of actions and tasks that define an approach to change management for all constituencies involved in management, engineering and use of software.

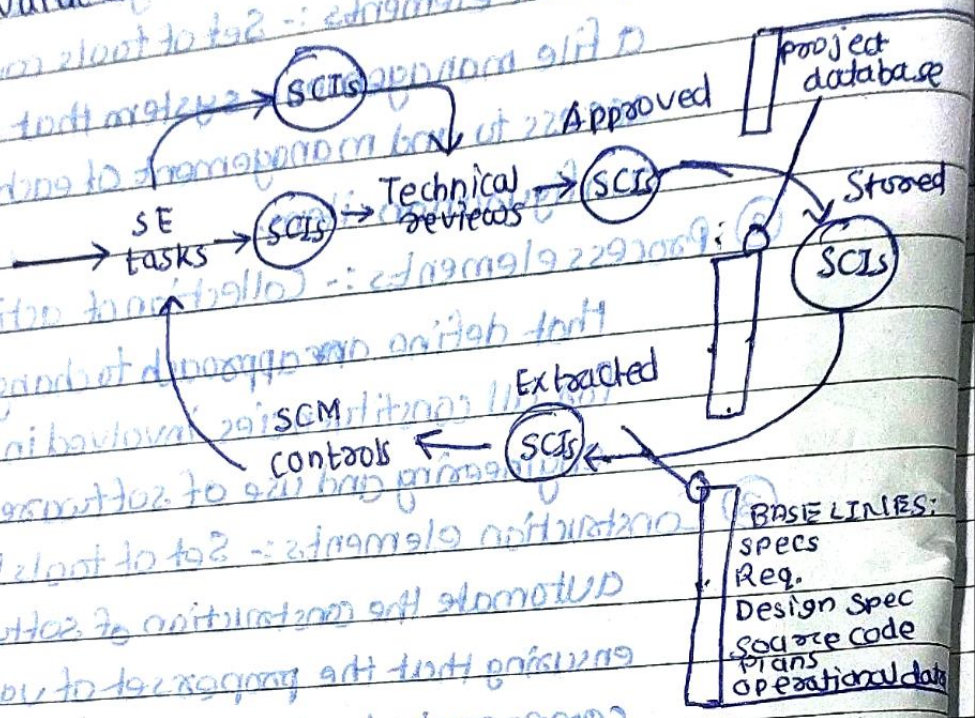
③ Construction elements :- Set of tools that automate the construction of software by ensuring that the proper set of validated components has been assembled.

④ Human elements :- Set of tools and process features used by the software team to implement effective SCM.

Baselines :- A baseline is a software configuration management concept that helps us to control change without seriously impending justifible change.

IEEE Def :- A specification or product that has been formally reviewed or agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control parameters.

Before a Software Configuration Item (SCI) becomes a baseline, changes may be made informally. However once a baseline is established, changes can be made, but a specific, formal, procedure must be applied to evaluate and verify each change.



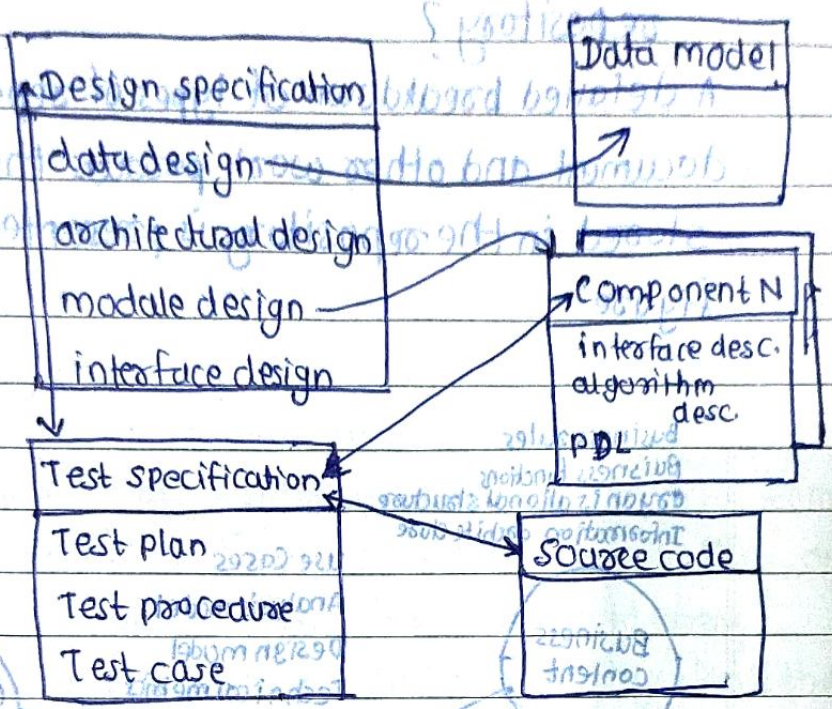
Baselines can be defined at any level of detail, the most common software baselines are shown in Figure above.

Software Configuration Items (SCI):-

Software Configuration Item is information that is created as part of the software engineering process. SCI is all or part of a work product (e.g. a document, an entire suite of test cases, or a program component).

Date

A configuration ~~item~~ object has a name, attributes and is connected to other objects by relationships. As shown in below figure, all configuration items are defined seperately. However each object is related to each other by the arrows.



A curved arrow indicates compositional relation i.e Data model and component N are part of object Design Specification.

A double headed straight arrow indicates an interrelationship.

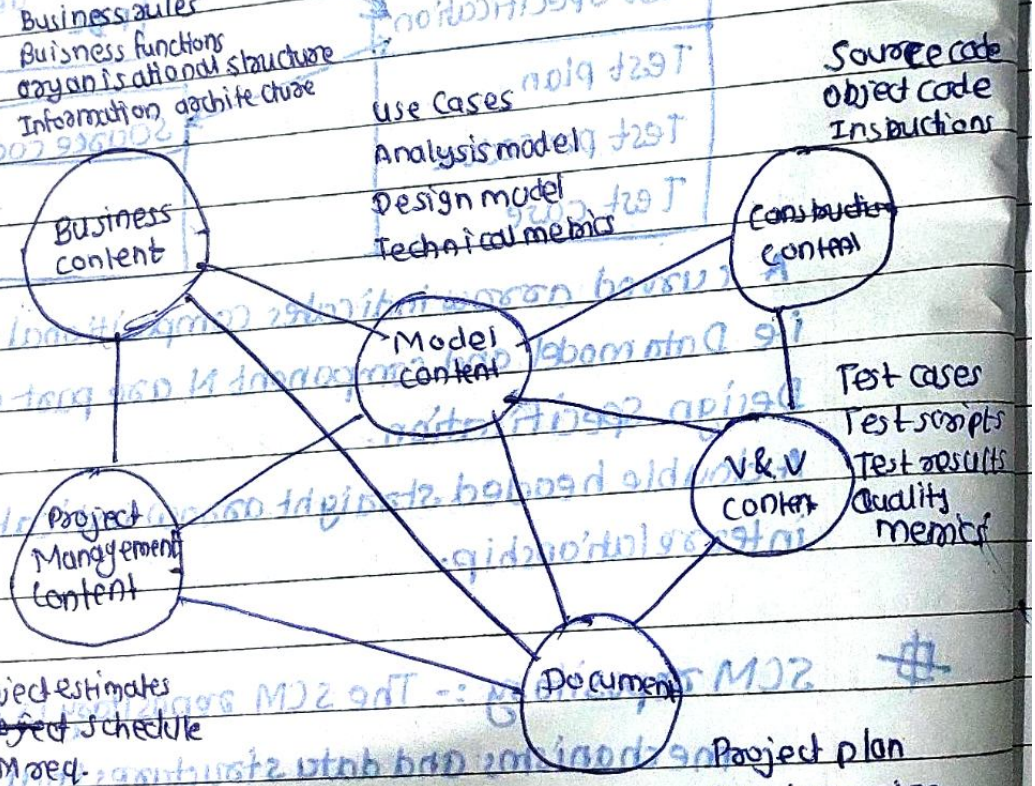
SCM repository :- The SCM repository is a set of mechanisms and data structures that allow a software team to manage change in effective manner. It provides the functions of a modern database management system by ensuring data integrity, sharing and integration.

General Features and content:

Features and content of repository are understood by looking by looking at it from two perspectives:

* What is to be stored in repository?
* What specific services are provided by the repository?

A detailed breakdown of types of representations, document and other work products that are stored in the repository is presented in below figure:



SCM features :-

To support SCM, the repository must have a tool set that provides the following features :

① Versioning :- As the project progresses, many versions of individual work products will be created. A repository must be able to save all these versions to enable effective management of product releases and to permit developers to go back to previous versions during testing & debugging.

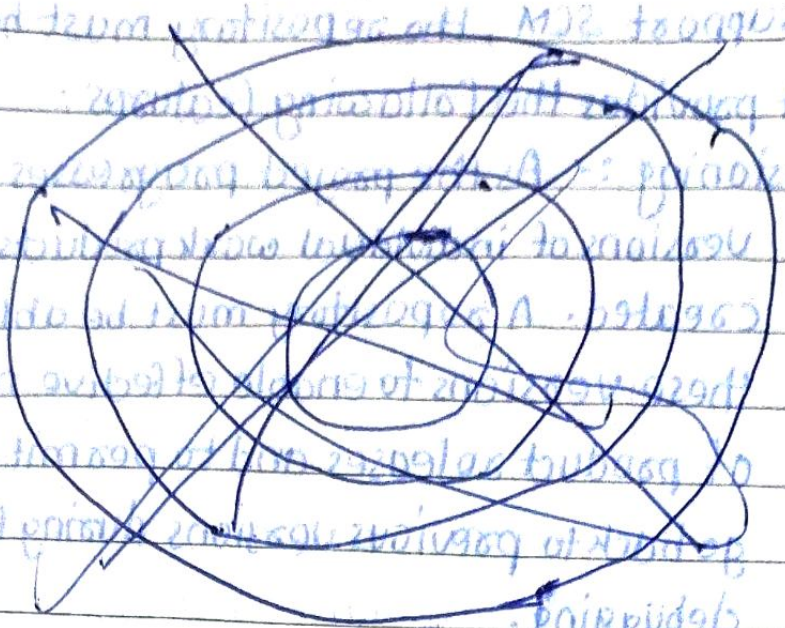
② Dependency tracking and change management :- Repository manages a wide variety of relations among data elements stored in it. Some of these relationships are merely associations, and some are dependencies, or mandatory relationships. The ability to keep track of all these relationships is important.

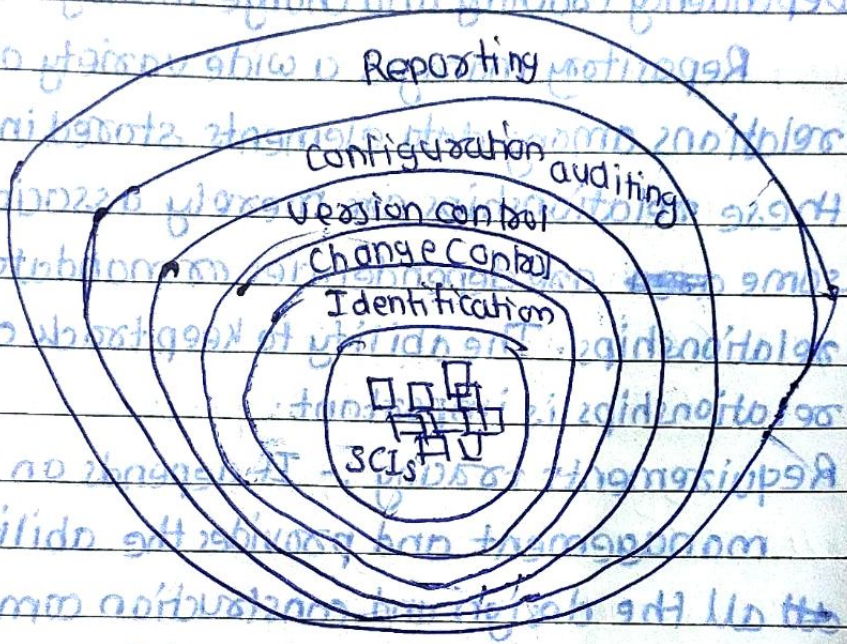
③ Requirements tracing :- It depends on link management and provides the ability to track all the design and construction components and deliverables that result from a specific requirements specification (forward tracing). In addition, it provides the ability to identify which requirement generated any given work product (backward tracing).

④ Configuration management :- It keeps track of a series of configuration representing specific project milestones or product releases.

⑤ Audit trails :- It establishes additional information about when, why and whom changes are made.

SCM process:-





SCM process defines a series of tasks that have four primary objectives:

1. To identify all items that collectively defines the software configuration.
2. To manage changes to one or more of these items
3. To facilitate the construction of different versions
4. To ensure that software quality is maintained as the configuration evolves over time.

SCM tasks:-

① Identification
② Version Control
③ Change control
④ Configuration auditing
⑤ Reporting

① Identification of objects in software Configuration

To control and manage SCIs, each should be seperately named and then organised using an object oriented approach.

Two types of objects:-

* Basic objects :- It is a unit of information that we create during analysis, design, code or test.
* Aggregate objects :- It is a collection of basic objects and other aggregate objects.

Each object has a set of distinct features :-

* Name * Description
* List of resources etc.

② Version Control :- It combines procedures and tools to manage different versions of configuration objects.

A version control system implements 4 major capabilities:-

* Repository :- A project database that stores all relevant configuration object.
* Version management :- Capability that stores all versions of configuration object

* Make facility :- that enables to collect all relevant configuration objects and construct a specific version of software.
* Issues tracking / bug tracking :- capability that enables the team to record the status of all outstanding issues associated with each configuration object.

(3) Change control :- Change control is important because small change can create big faliure in the product. For large software project, uncontrolled change creates big problems.

Important terms :-

* Change report :- The results of evaluation are presented as a change report
* Change Control authority (CCA) :- A person or a group that makes a final decision on the status and priority of change.
* Engineering Change Order (ECO) :- It is generated for each approved change

The ECO describes :-

(1) The change to be made
(2) The constraints that must be followed
(3) The criteria for review and audit.

Figure : The change control process

Need for change is recognized

↓

change request from user

↓

Developer evaluates

↓

change report is generated

↓

change authority decides

← Request is queued for action; ECO generated

→ change request is denied

↓

user is informed

↓

Assigns individuals to configuration objects

↓

check out configuration objects

↓

Make the changes

↓

Review (audit) the change

↓

"Check in" the configuration items that have been changed

↓

Establish a baseline for testing

↓

Perform quality assurance and testing activities

↓

"Promote" changes for inclusion in next release (revision)

↓

Rebuild appropriate version of software

↓

Review (audit) the change to all configuration items

↓

Include changes in new version

↓

Distribute the new version.

④ Configuration Audit :-
How can a software team ensure that the change has been properly implemented?
The answer is Twofold :-
* Technical reviews :- It focuses on technical correctness of the configuration object. A technical review should be conducted for most important changes.
* Software Configuration Audit :- It complements the technical review by accessing a configuration object for characteristics that are generally not considered during review. It is conducted seperately by quality assurance group.

⑤ Configuration Status Reporting (CSR) :-
CSR is an SCM task that answers the following questions :
* What happened? * Who did it?
* When did it happen? * Who else will be affected?
A CSR entry is always made when :-
① Each time an SCI is assigned new or updated identification
② Each time a change is approved by CCA.
③ Each time a configuration audit is conducted.
→ Output from CSR may be placed in an online database or website, so software developers or support staff can access change information.
→ CSR report is generated on regular basis and is intended to keep management and practitioners appraised of important changes