① What is Risk? Explain the reactive and proactive risk strategies with appropriate examples:

→ Risk:

The risk denotes the uncertainty that may occur in the choices due to past actions and risk is something which causes heavy loses.

Reactive risk strategy:

- Reactive risk management is risk management strategy in which when project gets into trouble then only corrective action is taken. But when such risks cannot be managed and new risks come up one after the other, the sw team flies into in an attempt to correct problems rapidly. These activities are called firefighting activities.
- Resources are used to manage such risks. And if still the risks do not get managed then the project is in danger
- In this strategies no preventive care is taken about the risks. They are handled only on their occurrence
- This is an older approach of risk management

Proactive risk strategy :-

- Proactive risk management strategy begins before the technical activity considering the probable risk
- In this strategy potential risk are identified first then their probability and impact is analyzed. Such risks are then specified according to priorities (ie. high priority risk should be managed first)

finally the s/w team prepares plan for managing these risks.

- The objective of this strategy is to avoid risks (prevention is better than cure!) But it is not possible to avoid all risks, hence team prepares the risk management plan in such a manner that risk controlling can done efficiently.
- This is an intelligent strategy for risk management and now a day it is used by most of IT industries.

* Tools for Automated Testing and features.

⇒ Automated s/w testing is a crucial aspect of s/w engineering, ensuring quality and reliability. Here are some tools commonly used for automated s/w testing:

(i) Selenium: A widely used tool for automating web browsers. It supports multiple programming languages and browsers.

(ii) JUnit / TestNG: These are popular Java testing frameworks used for unit testing

(iii) Pytest: A testing framework for Python that makes it easy to write simple tests.

(iv) Cucumber: A tool for behavior-driven development (BDD), allowing tests to be written in plain language.

Features :

(i) Version control
(ii) Code Review
(iii) Issue Tracking
(iv) Documentation Generation
(v) static code Analysis
(vi) Collaboration Tools.

* RMMM {Risk mitigation, monitoring, management}

⇒ RMMM stands for risk mitigation, monitoring and management.

Risk mitigation :

Risk mitigation means preventing the risks to occur (risk avoidance). following are the steps to be taken for mitigation

- communicate with the concerned staff to find of probable risk.
- Find out and eliminate all those causes that can create risk before the project starts.
- Develop a policy in an organization which will help to continue the project even though some staff leaves the organization
- conduct timely reviews in order to speed up the work.

Risk monitoring :

The project manager should monitor certain mitigation steps. for eg, if the current development activity is monitored continuosly then everybody in the team will get acquainted The objectives of risk monitoring is;

1) To check whether the predicted risks really occur or not.
2) To ensure the steps defined to avoid the risk are applied properly or not.
3) To gather the information which can be useful for analyzing the risk.

Risk Management :

- Project manager performs this task when risk becomes a reality.
- If project manager is successful in applying the project mitigation effectively then it becomes very much easy to manage the risks.
- For example: consider a sunario that many people are leaving the organization that if sufficient additional staff is availablu, if current development activity is known to everybody in the team, if latest and systematic documentation is availablu then any "new comer' can easily understand current development activity. This will ultimately hup in continuing the work without any interval.

* What is risk identification? what are different categories?

⇒ - Risk identification can be defined as the efforts taken to specify threats to the project plan. Risks identification can be done by identifying the known and predictable risks.

- It involves identifying potential threats or challanges that could affect the success of a sw project. Risks can be categorized in various ways, including generic risks applicable to most software projects and project specifics risks unique to a particular project.

- The risk identification is based on two approaches:

(i) Generic Risk:

They are potential threat to every sw project. This risk includes Technical risk, Schedule Risk, Resource risk, Requirement Risks, Quality risks, etc.

(ii) Product specific Risk:

The project plan and sw statement of scope are examined to find out threats due to special characteristics. The checklist list can be created of risk and then focus is subset of known and predictable risks in following sub categories:- product size & Complexity, Business impact, Staff size and experience, etc.

* Explain risk projection and risk refinement in detail.

⇒ RISK PROJECTION:

- The risk projection is also called risk estimation.
- There are two ways by which risk can be related.
  (i) Probability that risk is real
  (ii) consequences with that risk.

- The project planner, technical staff peeforms, following steps to peeform following steps for risk projection:
  (i) Building the risk table
  (ii) Enlist the consequences of risks.
  (iii) Estimate the impact of those risk on the project & product
  (iv) maintaining overall accuracy of risk projection.

RISK REFINEMENT:

- It is a process of specifying the risk in more detail. The risk refinement can be represented using CTC format suggested by PP Gluch. The CTC stands for condition-transition-~~stubseq~~ consequence.

- The condition is first stated and then based on this condition sub-condition can be derived. Then determine the effects of these conditions in order to refine the risk.

- The consequences cannot be eliminate but ~~the~~ can be refine which helps in easy analysis.

* Define software risk in detail. What are different types of software risk?

→ Risk :

- The risk denotes the uncertainty that may occur in the choices due to past actions and risk is something which causes heavy loses.
- Risk management refers to process of making decisions based on evaluations of the factors that threats to the business.

Different types of software risk :-

(i) Project risk :

Project risk arise in the s/w development process then they basically affect schedule, budget, staffing, resources and requirements. When project risk become severe then the total cost of project ~~get~~ gets increased.

(ii) Technical risk :

These risks affect quality and timelines of project. If technical risks become reality then potential design implementation, interface, verification and maintenance problems gets created. It occurs when problem becomes harder to solve.

3. **Business Risk** :

When feasibility of s/w product is in suspect then business risks occurs. Business risk can be further categorized as :

(i) Market risk - When quality s/w product is built but if there is no customer for this product then it is called market risk.

(ii) Strategic risk - product is built but doesn't follow business policies

(iii) Sales risk - product is built but how to sell is not clear

(iv) Management risk : ~~organization~~ senior management loses the organization

(v) Budget risk : Losing overall budget of project.

PAGE NO.
DATE:

Q. What is SCM? Write short note on SCM Elements.

→

- Software Configuration Management is a critical aspect that focuses on controlling the changes in software to maintain its integrity & traceability throughout the development lifecycle.

- During the development of software change must be managed & controlled in order to improve quality & reduce error.

- The software configuration management is concerned with managing evolving software systems.

- The origin of changes that are required for software are -

1. New business or market position cause changes in the requirement.

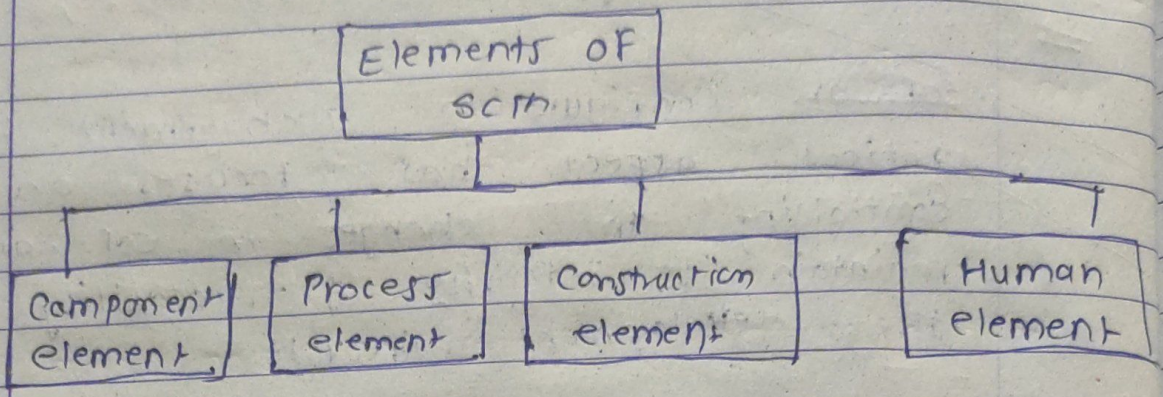
2. New stakeholder may require some changes in existing requirements.

3. Due to business growth or project extension, it is essential to make changes in project.

4. Sometime due to budget or schedule.

PAGE NO.
DATE:

Elements of Configuration management



1. Component Elements:

It consists of tools that are used for file management system.
e.g. database.

2. Process Element:

It consists of actions & tasks used during change management & use of software.

3. Construction Elements:

It is a collection of tools that automate the construction of software.

4. Human Elements:

It consists of set of tools that are used by software team to implement software configuration management.

PAGE NO.
DATE:

Q. Process of scm.

→

The primary object of software configuration management process are -

1. Configuration Identification:

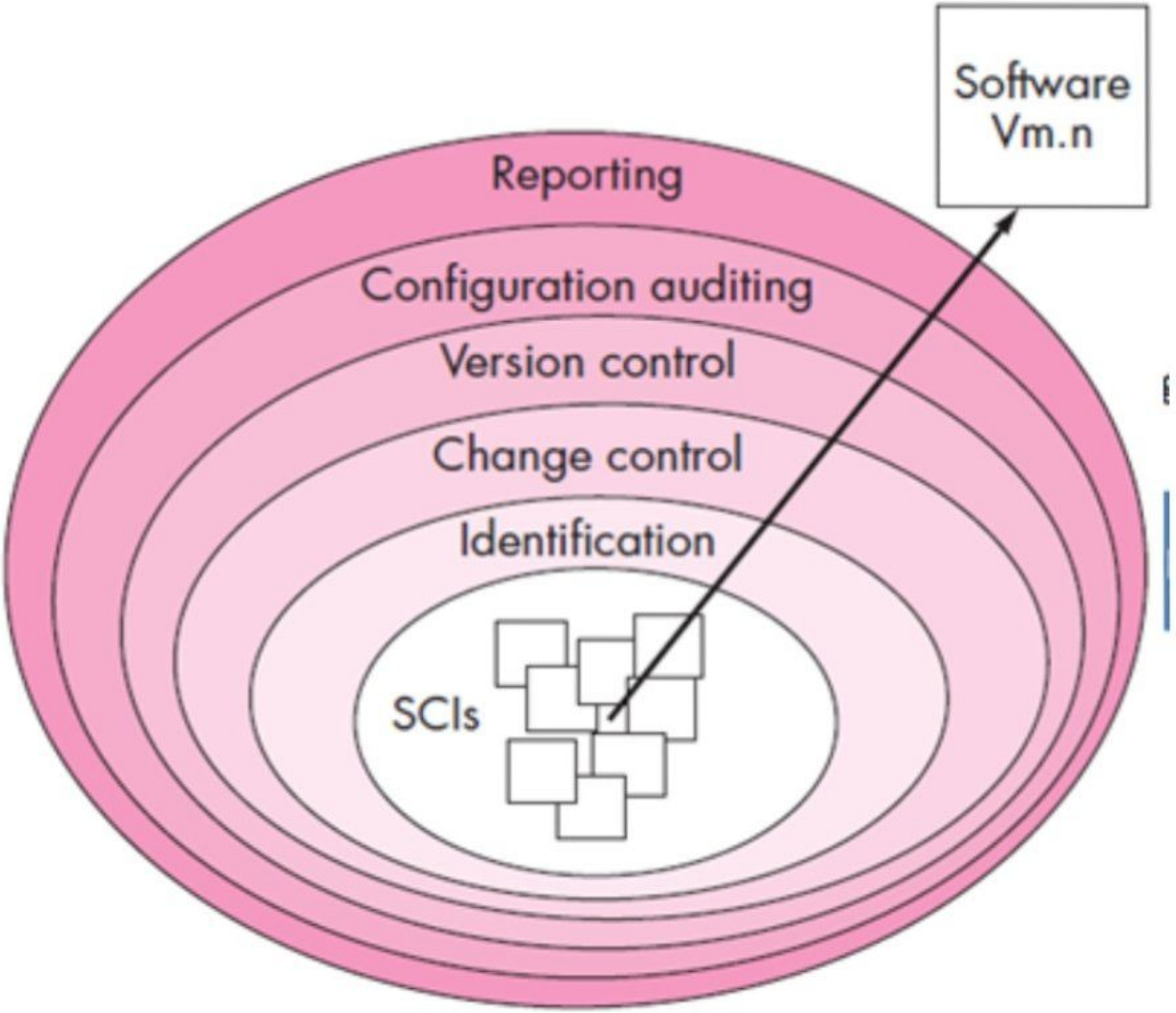
- This involves identifying & defining configuration items in software project.
- These items could include source code files, documentation, libraries etc.
- Each CI is given a unique identifier to track changes & versions.

2. Change control:

- Change control ensures that any changes made to the software are properly managed & controlled
- This includes submitting change ~~results~~ requests, evaluating their impact, approving or rejecting them & implementing approved changes.

3. Version control:

- Also known as revision control or source control, version control manages ~~the~~ changes to documents, programs & other information stored as computer files. It tracks changes over time, allowing developers to revert to previous versions.


Software
Vm.n
Reporting
Configuration auditing
Version control
Change control
Identification
SCIs

PAGE NO.
DATE:

4. Configuration Authentication:

- This aspect ensures that the integrity & authenticity of configuration items are maintained.

Q short note on change control mechanism in scm

⇒

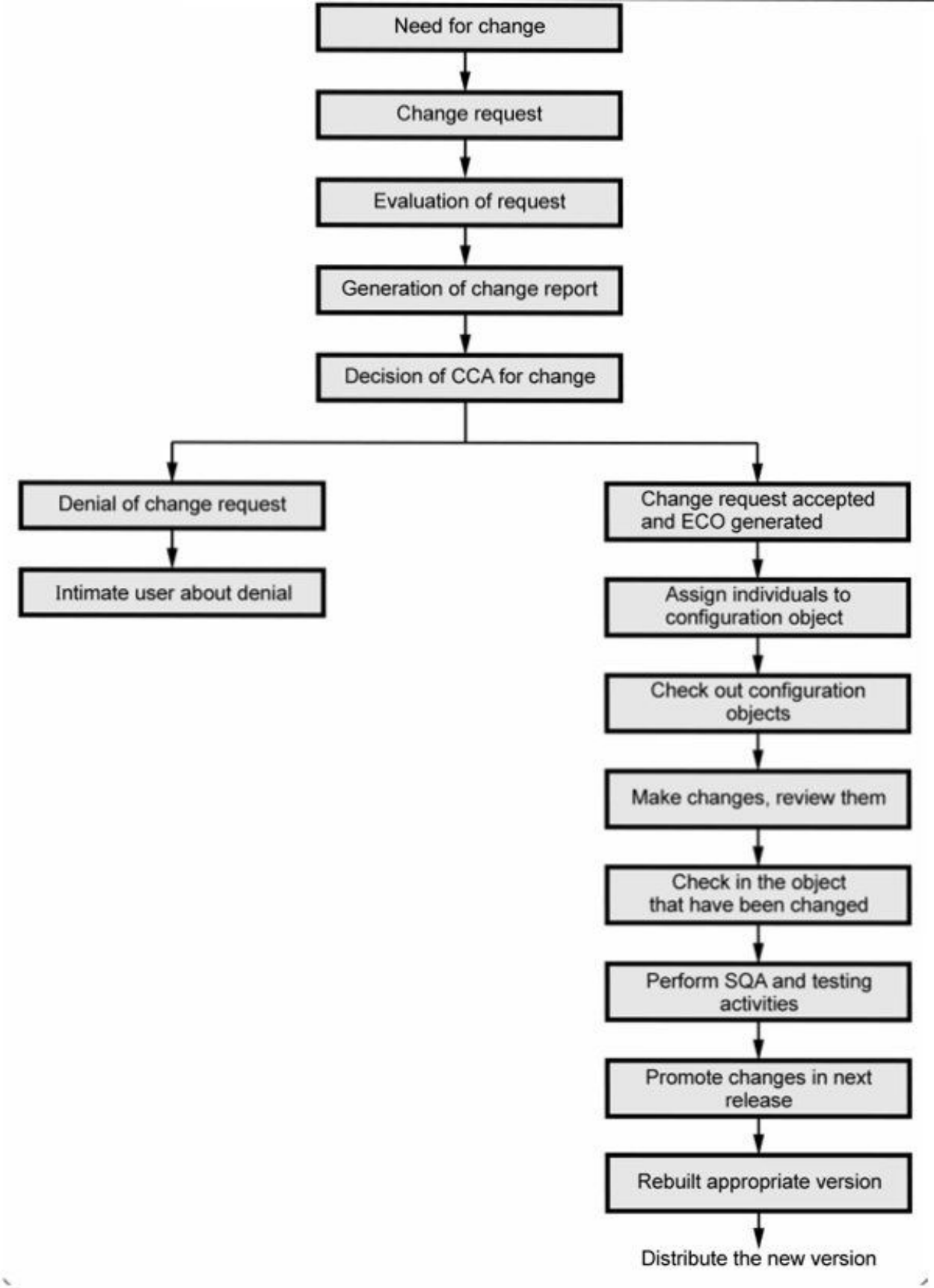


Fig. 5.11.1 Change control process

Step 1: First of all there arises a need for change.

Step 2: User ~~requests~~ submits the change result.

Step 3: Developer evaluate this request & predict ~~sid~~ potential side effects, & overall impact on system functions & cost of project.

Step 4: A change report is generated & presented to change control Authority (CCA).

Step 5: The CCA is a person or group of people who makes final decision.

Step 6: An ECO (Engineering change order) is generated when get approved. In ECO the change is described, the restrictions & criteria for review & audit are mentioned.

PAGE NO.
DATE:

step 7: The object that needs to be changed is checked out of project database

step 8: The changes are made & appropriate SQA activities are applied.

step 9: The changed object is then checked in to the database & appropriate version control is made to create new version.

Then it is deployed in newer version

* Explain the repository feature with respect to software configuration managment.

* SCM :-

- Software configuration Managment is defined as a process to systematically manage orginize and control the changes in the document's code and other entities during the software Development Life cycle.
- It keep track of change control and version control of all the files/entities that make up a software product.

* features :-

1. version control
   - the repository tracks changes to files over time enabling team's to maintain a history of madificationS
   - This includes who made changes, when they were made, what was change. It is used for enderstanding evolution of software.

2. collaboration.
   - By providing a shared space where team members can contribute code a repository facilitates collaboration
   - Muultiple developer's can work on different parts of the software simultaneously and SCM tool help merge changes. resolve confilicts and ensure code consistency.

3. ~~Brain~~ Backup and Recovery :-
As a central store for code and related artifacts the repository serves as a backup In case of data loss or corruption on individul machines

4. Access control and security.
- Repositories typically offer mechanisms for controlling access to the stored artifacts. This includes user authentication, authorization and permission to ensure thart only authorized personnel can make changes or view sensitive parts of the project.

5. Audit and complaince :-
- maintaining detailed record of changes repositories support audit trails and complaiance with various regulatory requirements.

6. Branching and Merging :-
- Repositories support braching which allows developer's to create separate line of development within the same project. This is useful for working the same project. this is useful for working on new features bug fixes or experiments without affecting the main codebase.