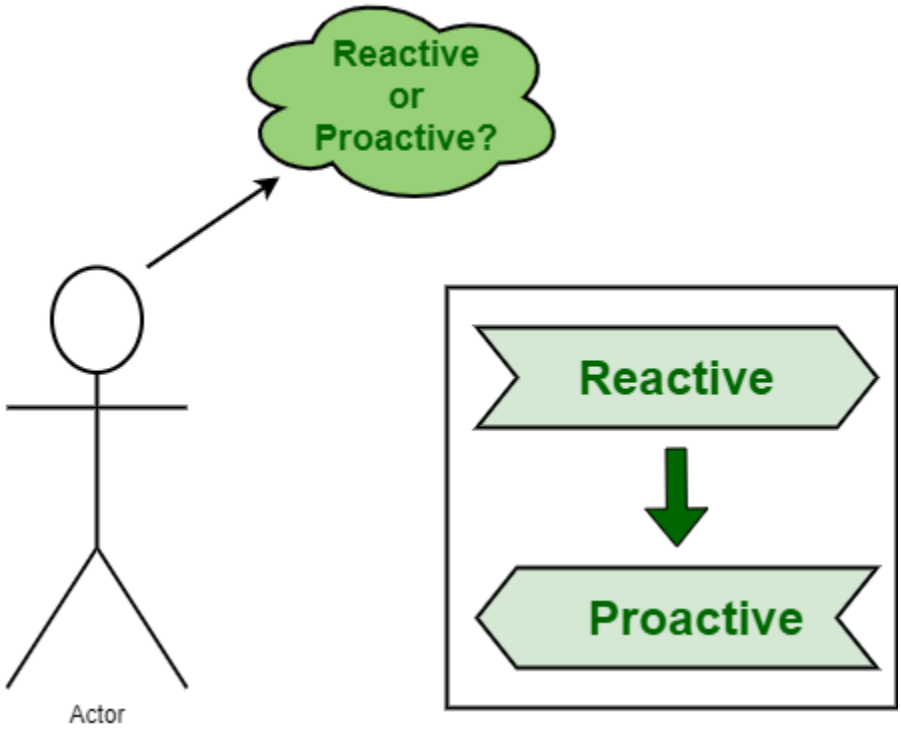
UNIT-V

Reactive and Proactive Risk Strategies:

- Root Cause Analysis (RCA) is one of the best methods to identify main cause or root cause of problems or events in very systematic way or process. RCA is based on the idea that for effective management, we need to find out way to prevent arising or occurring problems.
Each one needs to understand that if they want to solve or eliminate any problem, it is essential to go to the root cause of the problem and then eliminate problems so that they can reduce or control the reoccurrence of the problem. For organizations that want to improve and grow continuously, it is very essential to identify the root cause although it is tough to do so, it is essential. RCA can also be used to modify or change core processes and issues in such way that prevents future problems.

Reactive and Proactive RCA :
The main question that arises is whether RCA is reactive or proactive? Some people think that RCA is only required to solve problems or failures that have already occurred. But, it's not true. One should know that RCA can be both i.e. reactive and proactive as given below –



1. Reactive RCA :
The main question that arises in reactive RCA is "What went wrong?". Before investigating or identifying the root cause of failure or defect, failure needs to be in place or should be occurred

already. One can only identify the root cause and perform the analysis only when problem or failure had occurred that causes malfunctioning in the system. Reactive RCA is a root cause analysis that is performed after the occurrence of failure or defect.
It is simply done to control, implemented to reduce the impact and severity of defect that has occurred. It is also known as reactive risk management. It reacts quickly as soon as problem occurs by simply treating symptoms. RCA is generally reactive but it has the potential to be proactive. RCA is reactive at initial and it can only be proactive if one addresses and identifies small things too that can cause problem as well as exposes hidden causes of the problem.

Advantages :
- Helps one to prioritize tasks according to its severity and then resolve it.
- Increases teamwork and their knowledge.

Disadvantages :
- Sometimes, resolving equipment after failure can be more costly than preventing failure from an occurrence.
- Failed equipment can cause greater damage to system and interrupts production activities.

2. Proactive RCA :
The main question that arises in proactive RCA is "What could go wrong?". RCA can also be used proactively to mitigate failure or risk. The main importance of RCA can be seen when it is applied to events that have not occurred yet. Proactive RCA is a root cause analysis that is performed before any occurrence of failure or defect. It is simply done to control, implemented to prevent defect from its occurrence. As both reactive and proactive RCAs are is important, one should move from reactive to proactive RCA.
It is better to prevent issues from its occurrence rather than correcting it after its occurrence. In simple words, Prevention is better than correction. Here, prevention action is considered as proactive and corrective action is considered as reactive. It is also known as proactive risk management. It identifies the root cause of problem to eliminate it from reoccurring. With help of proactive RCA, we can identify the main root cause that leads to the occurrence of problem or failure, or defect. After knowing this, we can take various measures and implement actions to prevent these causes from the occurrence.

Advantages :
- Future chances of failure occurrence can be minimized.
- Reduce overall cost required to resolve failure by simply preventing failure from an occurrence.
- Increases overall productivity by minimizing chances of interruption due to failure.

Disadvantages :
- Sometimes, preventing equipment from failure can be more costly than resolving failure after occurrence.
- Many resources and tools required to prevent failure from an occurrence that can affect the overall cost.
- Requires highly skilled technicians to perform maintenance tasks.

Software Risks:

Software risk analysis in software development is a systematic process that involves identifying and evaluating any problem that might happen during the creation, implementation, and maintaining of software systems. It can guarantee that projects are finished on schedule, within budget, and with the appropriate quality. It is a crucial component of software development.

What is Software Risk Analysis in Software Development?

Software risk analysis in Software Development involves identifying which application risks should be tested first. Risk is the possible loss or harm that an organization might face. Risk can include issues like project management, technical challenges, resource constraints, changes in requirements, and more Finding every possible risk and estimating are the two goals of risk analysis. Think about the potential consequences of testing your software and how it could impact your software when creating a test plan. Risk detection during the production phase might be costly. Therefore, risk analysis in testing is the best way to figure out what goes wrong before going into production.

Why perform software risk analysis?

Using different technologies, software developers add new features in Software Development. Software system vulnerabilities grow in combination with technology. Software goods are therefore more vulnerable to malfunctioning or performing poorly.

Many factors, including timetable delays, inaccurate cost projections, a lack of resources, and security hazards, contribute to the risks associated with software in Software Development.

Certain risks are unavoidable, some of them are as follows:

- The amount of time you set out to test.
- Flaw leaks can happen in complicated or large-scale applications.
- The client has an immediate requirement to finish the job.
- The specifications are inadequate.

Therefore, it's critical to identify, priorities, and reduce risk or take proactive preventative action during the software development process, as opposed to monitoring risk possibilities.

Possible Scenarios of Risk Occurrence

Here are Some Possible Scenario of Software Risk

Unknown Unknowns

These risks are unknown to the organization and are generally technology related risk due to this these risks are not anticipated. Organizations might face unexpected challenges, delays, or failures due to these unexpected risks. Lack of experience with a particular tool or technology can lead to difficulties in implementation.

Example

Suppose an organization is using cloud service from third-party vendors, due to some issues third party vendor unable to provide its service. In this situation organization have to face an unexpected delay.

Known Knowns

These are risks that are well-understood and documented by the team. Since these risks are identified early, teams can plan for mitigation strategies. The impact of known knowns is usually more manageable compared to unknown risks.

Example

The shortage of developers is a known risk that can cause delays in software development.

Known Unknowns

In this case, the organization is aware of potential risks, but the certainty of their occurrence is uncertain. Organization should get ready to deal with these risks if they happen. Ways to deal with them might include making communication better, making sure everyone understands what's needed, or creating guidelines for how to manage possible misunderstandings.

Example

The team may be aware of the risk of miscommunication with the client, but whether it will actually happen is unknown.

Types of Software Risk

Given below table shows the type of risk and their impact with example:

Type of Risk	Description	Impact	Examples
Technical risks	Risks arising from technical challenges or limitations in the software development process.	Technical risks can lead to delays, cost overruns, and even software failure if not properly managed.	• Incomplete or inaccurate requirements • Unforeseen technical complexities • Integration issues with third-party systems • Inadequate testing and quality assurance
Security risks	Risks related to vulnerabilities in the software that could allow unauthorized access or data breaches.	Security risks can lead to financial losses, reputational damage, and legal liabilities.	• Insecure coding practices • Lack of proper access controls • Vulnerabilities in third-party libraries • Insufficient data security measures
Scalability risks	Risks associated with the software's ability to handle increasing workloads or user demands.	Scalability risks can lead to performance bottlenecks, outages, and lost revenue.	• Inadequate infrastructure capacity • Inefficient algorithms or data structures • Lack of scalability testing • Poorly designed architecture

Type of Risk	Description	Impact	Examples
Performance risks	Risks related to the software's ability to meet performance expectations in terms of speed, responsiveness, and resource utilization.	Performance risks can lead to user dissatisfaction, lost productivity, and competitive disadvantage.	• Inefficient algorithms or data structures • Excessive memory or CPU usage • Poor database performance • Network latency issues
Budgetary risks	Risks associated with exceeding the project's budget or financial constraints.	Budgetary risks can lead to financial strain, project delays, and even cancellation.	• Unrealistic cost estimates • Scope creep or changes in requirements • Unforeseen expenses, such as third-party licenses or hardware upgrades • Inefficient resource utilization
Contractual & legal risks	Risks arising from legal or contractual obligations that are not properly understood or managed.	Contractual and legal risks can lead to disputes, delays, and even legal action.	• Unclear or ambiguous contract terms • Failure to comply with intellectual property laws • Data privacy violations • Lack of proper documentation and record-keeping
Operational risks	Risks associated with the ongoing operation and maintenance of the software system.	Operational risks can lead to downtime, outages, and data loss.	• Inadequate monitoring and alerting systems • Lack of proper disaster recovery plans

Type of Risk	Description	Impact	Examples
			• Insufficient training for operational staff • Poor change management practices
Schedule risks	Risks related to delays in the software development process or missed deadlines.	Schedule risks can lead to increased costs, pressure on resources, and missed market opportunities.	• Unrealistic timelines or milestones • Underestimation of task complexity • Resource dependencies or conflicts • Unforeseen events or delays

How to perform software risk analysis in Software Development

In order to conduct risk analysis in software development, first you have to evaluate the source code in detail to understand its component. This evaluation is done to address components of code and map their interactions. With the help of the map, transaction can be detected and assessed. The map is subjected to structural and architectural guidelines in order to recognize and understand the primary software defects. Following are the steps to perform software risk analysis.

Risks Identifiation:

- Identifying risk is one of most important or essential and initial steps in risk management process. By chance, if failure occurs in identifying any specific or particular risk, then all other steps that are involved in risk management will not be implemented for that particular risk. For identifying risk, project team should review scope of program, estimate cost, schedule, technical maturity, parameters of key performance, etc. To manage risk, project team or organization are needed to know about what risks it faces, and then to evaluate them. Generally, identification of risk is an iterative process. It basically includes generating or creating comprehensive list of threats and opportunities that are based on events that can enhance, prevent, degrade, accelerate, or might delay successful achievement of objectives. In simple words, if you don't find or identify risk, you won't be able to manage it.
  The organizer of project needs to expect some of the risk in the project as early as possible so that the performance of risk may be reduced. This could be only possible by making effective risk management planning.

  A project may contain large variety of risk. To know the specific amount of risk, there may be chance of affecting a project. So, this is necessary to make categories into different class of risk.

There are many different types of risks which affects the software project:

1. Technology risks
2. Tools risks
3. Estimation risks
4. People risks
5. Requirement risks
6. Organizational risks

Methods for Identifying Risks : Earlier, there were no easy methods available that will surely identify all risks. But nowadays, there are some additional approaches available for identifying risks. Some of approaches for risk identification are given below:

1. Checklist Analysis – Checklist Analysis is type of technique generally used to identify or find risks and manage it. The checklist is basically developed by listing items, steps, or even tasks and is then further analyzed against criteria to just identify and determine if procedure is completed correctly or not. It is list of risk that is just found to occur regularly in development of software project. Below is the list of software development risk by Barry Boehm- modified version.

Risk	Risk Reduction Technique
Personnel Shortfalls	**Various techniques include training and career development, job-matching, teambuilding, etc.**
Unrealistic time and cost estimates	**Various techniques include incremental development, standardization of methods, recording, and analysis of the past project, etc.**
Development of wrong software functions	**Various techniques include formal specification methods, user surveys, etc.**
Development of the wrong user interface	**Various techniques include user involvement, prototyping, etc.**

2. Brainstorming – This technique provides and gives free and open approach that usually encourages each and everyone on project team to participate. It also results in greater sense of ownership of project risk, and team generally committed to managing risk for given time period of project. It is creative and unique technique to gather risks spontaneously by team members. The team members identify and determine risks in 'no wrong answer' environment. This technique also provides opportunity for team members to always develop on each other's ideas. This technique is also used to determine best possible solution to problems and issue that arises and emerge.

3. Casual Mapping – Causal mapping is method that builds or develops on reflection and review of failure factors in cause and effect of the diagrams. It is very useful for facilitating learning with an organization or system simply as method of project-post evaluation. It is also key tool for risk assessment.

4. SWOT Analysis – Strengths-Weaknesses-Opportunities-Threat (SWOT) is very technique and helpful for identifying risks within greater organization context. It is generally used as planning tool for analyzing business, its resources, and also its environment simply by looking at internal strengths and weaknesses and opportunities and threats in external environment. It is technique often used in formulation of strategy. The appropriate time and effort should be spent on thinking

seriously about weaknesses and threats of organization for SWOT analysis to more effective and successful in risk identification.

5. Flowchart Method – This method allows for dynamic process to be diagrammatically represented in paper. This method is generally used to represent activities of process graphically and sequentially to simply identify the risk.

Risk Projection:

- In Project Management, Project risk analysis is a component of effective project management, assessing, and mitigating potential threats that may impact the successful completion of a project. In order to ascertain the possibility and possible impact of risks, as well as to develop management or elimination methods, it is necessary to carefully evaluate many aspects in an iterative process.

What is Project Risk Analysis?

Project risk analysis entails creating risk response strategies specific to every danger that is detected. These plans specify the precise steps that must be done to transfer, minimize, accept, or avoid the risk. Organizations can avoid the negative effects of unplanned occurrences and sustain project momentum by proactively planning for probable contingencies.

The methodical process of locating, evaluating, and controlling the hazards that could compromise a project's successful completion is known as project risk analysis. It entails assessing risks and possible dangers to project goals, including budget, time, scope, and quality, and creating plans to successfully manage or address these hazards. Project risk analysis's main objective is to proactively detect and handle possible problems before they become serious ones to increase the possibility that the project will succeed.

How to Analyze Project Risks?

When evaluating project risks, you should take three factors into account: risk exposure, risk impact, and risk probability. Risk analysis, both qualitative and quantitative, can be used to estimate these three factors.

1. Risk Probability

- **Qualitative Analysis:** This method determines the possibility of a risk materializing by utilizing experience and subjective judgment. One can use methods like probability matrices, risk rating scales, and expert opinion.
- **Quantitative Analysis:** To evaluate the probability of risks, quantitative methods use numerical data and statistical models, in contrast to qualitative analysis. This could use methods like decision trees, historical data analysis, and Monte Carlo simulations.

2. Risk Impact

- **Financial Impact:** Consider the possible financial repercussions of a risk, including direct and indirect expenses as well as possible revenue loss.
- **Impact on Schedule:** Evaluate the potential effects of a risk on the project schedule, such as missed deadlines for completing tasks or reaching milestones.
- **Impact on Resources**: Take into account the effects on supplies, machinery, labor, and other project resources.
- **Impact on Quality:** Assess the potential effects of a risk on project results or deliverable quality requirements.

3. Risk Exposure

- **Assessing Acceptability:** Use the risk exposure calculation to ascertain whether the company is prepared to take on the possible losses that come with a risk. This computation aids in risk prioritization according to likelihood and total impact.
- **Risk Mitigation:** Strategies for reducing risk likelihood or impact should be created in order to lessen the predicted risk exposure if it is deemed unacceptable.
- **Risk Transfer or Avoidance:** If an organization's risk exposure is judged to be too large or to be outside of its risk tolerance threshold, it may decide to transfer or completely avoid hazards.

Project Risk Analysis Tools & Techniques

Managers can make better decisions by using a variety of risk analysis techniques and resources. Project management documents and charts are examples of instruments used in risk analysis that are used in some of these. Now let's explore these risk analysis techniques and see how they might benefit you.

1. Team Brainstorming Sessions

Participating in brainstorming sessions with team members guarantees that different viewpoints are taken into account when calculating the probability and effect of risks. A more accurate risk assessment can be achieved by utilizing the team's collective expertise and experience to identify potential threats in a more thorough manner. Involvement in the team also promotes ownership and dedication to the risk management procedure, which raises the possibility that risk mitigation techniques will be effective.

2. Delphi Technique

The Delphi method uses a panel of experts' knowledge to predict risks and their possible effects. Through expert discussion and debate, the method helps identify biases and blind spots, resulting in better informed risk assessments. This method's consensus-building offers a strong basis for making decisions, especially in risk scenarios that are unclear or complex.

3. SWOT Analysis

A project's internal strengths and weaknesses as well as exterior possibilities and dangers can be seen holistically with the use of a SWOT analysis. Project managers can use SWOT analysis as a method for risk analysis to find any weaknesses and outside variables that could endanger the success of their project. Through the consideration of both external and internal aspects, SWOT analysis aids in the proactive development of plans to reduce risks and take advantage of opportunities.

4. Risk Analysis Matrix

The risk analysis matrix offers an organized framework for assessing a danger's likelihood and seriousness. Project managers can efficiently prioritize risks and allocate resources based on their level of importance. The matrix is a useful tool for directing risk management efforts and making sure that major hazards are addressed promptly, even though it only provides a qualitative assessment of risks.

5. Risk Register

For recording and monitoring project risks over the course of the project lifetime, the risk register acts as a central repository. The risk register offers a thorough perspective of the project's risk environment by gathering crucial information about risks, including their nature, possible impact, and mitigation techniques. The risk register assists with proactive risk management by identifying and addressing possible issues before they become more serious. It does this by utilizing inputs from multiple sources, such as the project team and historical data.

Types of Project Risk Analysis

1. Qualitative Risk Analysis

Qualitative risk analysis involves experts from the project team estimating the impact and likelihood of different risks based on their experience and past project data. To rate risks according to their impact (severity of consequences) and probability (chance of occurrence), they employ a scale. When a danger has a likelihood of 0.5, for instance, there is a 50% chance that it will materialize. On a five-point rating system, one represents the least severe impact and five the most severe. Following risk identification and analysis, a team member is designated as the risk owner, who is in charge of organizing and carrying out a response. By concentrating on high-impact risks and designating owners to handle them successfully, qualitative analysis helps projects become less uncertain.

2. Quantitative Risk Analysis

Quantitative risk analysis is a more statistical approach that examines how identified risks might affect the overall project. It entails calculating the likelihood that project goals will still be met in spite of these risks by counting the alternative outcomes. This analysis enhances risk control initiatives and gives project managers more confidence when making decisions. It assists, for example, in establishing reasonable goals for project scope, budgets, and schedules. The Monte Carlo simulation, which employs computational techniques to predict the possibility of various risks occurring, is one often used tool in quantitative analysis. During the planning and execution of a project, project managers can use this data to make well-informed decisions.

Case Studies of Project Risk Analysis

Case Study 1: Building a High-Rise Residential Structure

1. Recognizing Dangers

- Identified hazards include unfavorable weather, problems with the supply chain, a labor shortage, and problems with regulatory compliance.
- Organized risk brainstorming sessions with project managers, engineers, contractors, and regulatory agencies.

2. Evaluating Hazards

- Evaluated each detected risk's likelihood and its consequences using a qualitative method.
- Based on their seriousness and probability of happening, risks were ranked, with the greatest influence on project finances and schedules coming first.

3. Planning for Mitigation

- Developed mitigating measures, including recruiting backup workers, setting up alternate suppliers for essential commodities, and adjusting schedules to account for weather-related delays.
- Safety training initiatives and compliance audits were put in place to reduce regulatory risks and guarantee worker safety.

4. Emergency Preparedness

- Developed backup measures for high-impact risks, such as scheduling buffers and budget reserves for unforeseen expenses.
- Established criteria and triggers for triggering backup plans, and evaluated their efficacy on a regular basis.

5. Observation and Management

- Used important risk indicators, such as weather forecasts, supplier performance data, and regulatory compliance reports, to monitor project risks during the building phase.
- A risk management plan was put into place to monitor risk reduction initiatives, keep risk registers up to date, and inform project stakeholders of developments pertaining to risks.

Case Study 2: Financial Institution Software Development

1. Recognizing Dangers

- Hazards that have been identified include changes in scope, technical complexity, resource limitations, and security flaws.
- Conducted requirements analysis meetings and stakeholder interviews to find any hazards related to software development and integration.

2. Evaluating Hazards

- Evaluated the possibility and significance of each risk that was discovered using a combination of qualitative and quantitative techniques.
- Risks were ranked according to how they might affect data security, project deliverables, and regulatory compliance.

3. Planning for Mitigation

- Created techniques for mitigation, including cross-training team members to lessen resource restrictions, introducing change control procedures to manage scope changes, and addressing technical complexity through modular development.
- Carried out frequent penetration tests and security assessments to find and fix any possible weaknesses in the software program.

4. Emergency Preparedness

- Plans for backup development resources in case of personnel turnover and emergency response procedures in case of security breaches have been developed as contingency measures for critical risks.
- Created channels of communication and escalation protocols to initiate backup plans when necessary.

5. Observation and Management

- Used data including code review reports, stakeholder comments, and security audit results to track project risks.
- Conducted frequent risk assessments and status reports to monitor risk reduction initiatives, reevaluate risk priorities, and modify mitigation plans as needed.

Challenges of Project Risk Analysis

1. **Uncertainty:** Projects can entail a large number of unknowns, which makes it difficult to precisely identify and forecast possible hazards.
2. **Subjectivity in Risk Assessment:** Risk assessment calls for subjective assessments that differ depending on the project's stakeholders. Subjectivity in risk assessment and prioritization might result in prejudices and conflicts.
3. **Lack of Historical Data:** Occasionally, particularly for novel or inventive initiatives, there could not be enough historical data or benchmarks available to guide risk analysis.
4. **Interrelated Risks:** Risks in a project are frequently interrelated, which means that addressing one risk could unintentionally cause or worsen others. Sustaining these interdependencies calls for meticulous planning and collaboration.
5. **Ignoring Certain Risks:** Project teams have a tendency to ignore certain hazards, particularly those that are less evident or concealed from view. This may lead to insufficient methods for mitigating risks or unforeseen problems when the project is being carried out.
6. **Dynamic Project Environments:** Project environments are dynamic, meaning that risks alter over time as a result of adjustments made to rules, market conditions, technology, or stakeholder expectations. Staying on top of these changes means constantly observing and adjusting.

Benefits of Project Risk Analysis

- **Proactive Risk Management:** Early risk identification allows project teams to take proactive steps to reduce or eliminate risks. This is known as proactive risk management. By being proactive, risks have less of an impact on the goals of the project.
- **Informed Decision Making through Risk Analysis:** Throughout the course of a project, risk analysis offers insightful information that facilitates well-informed decision making. Stakeholders in the project can evaluate the possible outcomes of various options and allocate resources appropriately.
- **Maximizing Resource Usage and Efficiency:** Time, money, and manpower may all be used more wisely when project risks are recognized. Project teams can increase project efficiency and maximize resource usage by concentrating resources on high-priority hazards.
- **Proactive Risk Management:** Enhanced Stakeholder trust: Showing that you have a solid grasp of project risks and are employing proactive risk management techniques helps to build stakeholder trust in the project's capacity to meet its goals. This in turn cultivates confidence and backing from clients, sponsors, and other stakeholders involved in the project.
- **Implementing Cost Control Strategies:** Project risk analysis makes it possible to implement better cost control strategies by seeing possible cost overruns early in the project lifecycle.
- **Schedule Risk Management:** To reduce financial risk, this entails creating a contingency budget, negotiating contracts with suppliers, and putting cost-cutting measures in place.

Best Practices for Effective Risk Analysis in Projects

1. Planning for Risk Management

Create a plan for risk management. Uncertainty surrounds every project. Establishing a well-defined risk management plan at the outset establishes the framework for managing hazards. Risk appetite, roles and responsibilities, data sources and technologies, and the frequency and timing of risk management actions should all be outlined in the strategy.

2. Qualitative and Quantitative Approaches

Both qualitative and quantitative approaches are used in qualitative and quantitative risk analysis. Quantitative risk analysis, such as Monte Carlo simulations, adds depth to the risk assessment by providing numerical estimates of possible outcomes, while qualitative risk analysis helps prioritize risks based on probability and impact.

3. Frequent Re-evaluation of Risk

Make iterative assessments of the risks. Projects change as they go, bringing with them new and evolving hazards. Plan frequent risk assessment meetings to identify and handle these situations, so the team isn't taken by surprise.

Risk Refinement :

- Risk Management is an important part of project planning activities. It involves identifying and estimating the probability of risks with their order of impact on the project.

Risk Management Steps:

Some steps need to be followed to reduce risk. These steps are as follows:

1. Risk Identification:

Risk identification involves brainstorming activities. It also involves the preparation of a risk list. Brainstorming is a group discussion technique where all the stakeholders meet together. This technique produces new ideas and promotes creative thinking.

Preparation of a risk list involves the identification of risks that are occurring continuously in previous software projects.

2. Risk Analysis and Prioritization:

It is a process that consists of the following steps:

- Identifying the problems causing risk in projects
- Identifying the probability of occurrence of the problem
- Identifying the impact of the problem
- Assigning values to step 2 and step 3 in the range of 1 to 10
- Calculate the risk exposure factor which is the product of values of Step 2 and Step 3
- Prepare a table consisting of all the values and order risk based on risk exposure factor

For example,

TABLE (Required)

Risk No	Problem	Probability of occurrence of problem	Impact of problem	Risk exposure	Priority
R1	Issue of incorrect password	2	2	4	10
R2	Testing reveals a lot of defects	1	9	9	7
R3	The design is not robust	2	7	14	5

3. Risk Avoidance and Mitigation:

The purpose of this technique is to eliminate the occurrence of risks. so the method to avoid risks is to reduce the scope of projects by removing non-essential requirements.

4. Risk Monitoring:

In this technique, the risk is monitored continuously by reevaluating the risks, the impact of risk, and the probability of occurrence of the risk. This ensures that:

- Risk has been reduced
- New risks are discovered
- The impact and magnitude of risk are measured.

Risk Mitigation, Monitoring, and Management (RMMM):

- **RMMM:**

A risk management technique is usually seen in the software Project plan. This can be divided into Risk Mitigation, Monitoring, and Management Plan (RMMM). In this plan, all works are done as part of risk analysis. As part of the overall project plan project manager generally uses this RMMM plan.

In some software teams, risk is documented with the help of a Risk Information Sheet (RIS). This RIS is controlled by using a database system for easier management of information i.e creation, priority ordering, searching, and other analysis. After documentation of RMMM and start of a project, risk mitigation and monitoring steps will start.

Risk Mitigation :

It is an activity used to avoid problems (Risk Avoidance).

Steps for mitigating the risks as follows.

1. Finding out the risk.
2. Removing causes that are the reason for risk creation.
3. Controlling the corresponding documents from time to time.
4. Conducting timely reviews to speed up the work.

Risk Monitoring:

It is an activity used for project tracking.

It has the following primary objectives as follows.

1. To check if predicted risks occur or not.
2. To ensure proper application of risk aversion steps defined for risk.
3. To collect data for future risk analysis.
4. To allocate what problems are caused by which risks throughout the project.

Risk Management and planning:

It assumes that the mitigation activity failed and the risk is a reality. This task is done by Project manager when risk becomes reality and causes severe problems. If the project manager effectively uses project mitigation to remove risks successfully then it is easier to manage the risks. This shows that the response that will be taken for each risk by a manager. The main objective of the risk management plan is the risk register. This risk register describes and focuses on the predicted threats to a software project.

Example:

Let us understand RMMM with the help of an example of high staff turnover.

Risk Mitigation:

To mitigate this risk, project management must develop a strategy for reducing turnover. The possible steps to be taken are:

- Meet the current staff to determine causes for turnover (e.g., poor working conditions, low pay, competitive job market).
- Mitigate those causes that are under our control before the project starts.
- Once the project commences, assume turnover will occur and develop techniques to ensure continuity when people leave.
- Organize project teams so that information about each development activity is widely dispersed.
- Define documentation standards and establish mechanisms to ensure that documents are developed in a timely manner.

- Assign a backup staff member for every critical technologist.

Risk Monitoring:

As the project proceeds, risk monitoring activities commence. The project manager monitors factors that may provide an indication of whether the risk is becoming more or less likely. In the case of high staff turnover, the following factors can be monitored:

- General attitude of team members based on project pressures.
- Interpersonal relationships among team members.
- Potential problems with compensation and benefits.
- The availability of jobs within the company and outside it.

Risk Management:

Risk management and contingency planning assumes that mitigation efforts have failed and that the risk has become a reality. Continuing the example, the project is well underway, and a number of people announce that they will be leaving. If the mitigation strategy has been followed, backup is available, information is documented, and knowledge has been dispersed across the team. In addition, the project manager may temporarily refocus resources (and readjust the project schedule) to those functions that are fully staffed, enabling newcomers who must be added to the team to "get up to the speed**".**

Drawbacks of RMMM:

- It incurs additional project costs.
- It takes additional time.
- For larger projects, implementing an RMMM may itself turn out to be another tedious project.
- RMMM does not guarantee a risk-free project, in fact, risks may also come up after the project is delivered.

Risk Assessment

The purpose of the risk assessment is to identify and priorities the risks at the earliest stage and avoid losing time and money.
Under risk assessment, you will go through:

- **Risk identification:** It is crucial to detect the type of risk as early as possible and address them. The risk types are classified into
    - People risks: related to the people in the software development team
    - Tools risks: related to using tools and other software
    - Estimation risks: related to estimates of the resources required to build the software
    - Technology risks: are related to the usage of hardware or software technologies required to build the software
    - Organizational risks: are related to the organizational environment where the software is being created.
- **Risk analysis:** Experienced developers analyze the identified risk based on their experience gained from previous software . In the next phase, the Software Development team estimates the probability of the risk occurring and its seriousness
- **Risk prioritization:** The risk priority can be identified using the formula below

p = r * s

Where,
p stands for priority
r stands for the probability of the risk becoming true or false
s stands for the severity of the risk.
After identifying the risks, the ones with the probability of becoming true and higher loss must be prioritized and controlled.

Risk control

Risk control is performed to manage the risks and obtain desired results. Once identified, the risks can be classified into the most and least harmful.
Under risk control, you will go through:

- **Risk management planning:** You can leverage three main strategies to plan risk management.
    - Reduce the risk: This method involves planning to reduce the loss caused by the risk. For instance, planning to hire new employees to replace employees serving notice.
    - Transfer the risk: This method involves buying insurance or hiring a third-party organization to solve a challenging problem that might pose harmful risks
    - Avoid the risk: This method involves implementing various strategies, such as incentivizing underpaid, hardworking engineers who might quit the organization
- **Risk monitoring:** It includes tracking and evaluating different levels of risk in the software development team. After completing the risk monitoring process, the findings can be utilized to devise new strategies to update ineffective methods
- **Risk resolution:** It involves eliminating the overall risk or finding solutions. This method includes techniques such as design to cost approach, simulating the prototype, benchmarking, etc.

Key Benefits of Software Risk Analysis

There are multiple benefits to using software risk analysis techniques within your software in software development, ultimately leading you to complete your projects while successfully navigating obstacles along the way. Some of the most positive outcomes you can expect when using this framework include: There are many benefits to using software a

- **Better decision-making:** When you have the right information in front of you, it is much easier to make good decisions. Data-driven decision-making is one of the best ways to ensure the successful completion of a project, which can have knock-on benefits such as cost savings and faster turnaround times.
- **Early warning:** If you are aware of an issue before it affects your software and operations, then you will be able to prevent expensive and time-draining fixes from being necessary.
- **Reduced software costs and time:** Addressing potential risks ahead of time can help reduce software costs and time by avoiding costly rework or delays due to unexpected issues.
- **Improved software quality:** Risk analysis can help identify potential quality issues and ensure that software quality is maintained throughout the development process.
- **Increased stakeholder confidence:** Conducting risk analysis can increase stakeholder confidence in the software development process by demonstrating that potential risks are managed proactively.

- **Compliance with regulations:** Risk analysis can help ensure compliance with industry regulations and standards.

Best Tools for Software Risk Analysis

Some of the most commonly used tools for software risk analysis are as follows:

- **Failure Mode and Effects Analysis (FMEA)**
  - FMEA is an organized method for locating, evaluating, and ranking possible flaws in a process or system. It is a qualitative method that evaluates the possibility and seriousness of prospective failures using the opinion of experts. When risks are found and addressed early in the software development lifecycle, FMEA is a useful technique.
- **Fault Tree Analysis (FTA)**
  - FTA is a logical method for assessing system failure reasons. It begins with an undesirable occurrence at the highest level and proceeds downward to find the lower-level events that may have contributed to the event. FTA is a helpful tool for comprehending the intricate connections that exist between various system hazards.
- **Risk Matrix**
  - Prioritizing risks according to likelihood and impact may be done easily with a risk matrix. A likelihood and impact rating is given to each risk, and the two ratings are then compounded to provide a risk score. Prioritisation of more research and mitigation is given to risks with high risk ratings.
- **Decision Tree**
  - A decision tree is a diagram that represents a series of decisions and their possible outcomes. Decision trees are helpful in weighing the advantages and disadvantages of various options.
- **Monte Carlo Simulation**
  - Monte Carlo is a quantitative technique for calculating the probability of different outcomes. It includes running computer simulation multiple times, using random values as input each time. The results of these simulation can be used to calculate the chances of different outcomes.

QUALITY MANAGEMENT

1) QUALITY CONCEPTS:

Quality management encompasses

(1) a quality management approach,
(2) effective software engineering technology (methods and tools),
(3) formal technical reviews that are applied throughout the software process,
(4) a multitiered testing strategy,
(5) control of software documentation and the changes made to it,
(6) a procedure to ensure compliance with software development standards (when applicable), and
(7) measurement and reporting mechanisms.

Variation control is the heart of quality control.

Quality

- ✓ The *American Heritage Dictionary* defines *quality* as "a characteristic or attribute of something."
- ✓ ***Quality of design*** refers to the characteristics that designers specify for an

item.

Quality of conformance is the degree to which the design specifications are followed during manufacturing.

In software development, quality of design encompasses requirements, specifications, and the design of the system. Quality of conformance is an issue focused primarily on implementation. If the implementation follows the design and the resulting system meets its requirements and performance goals, conformance quality is high.

Robert Glass argues that a more "intuitive" relationship is in order:

User satisfaction = compliant product + good quality + delivery within budget and schedule

Quality Control

Quality control involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it.

A key concept of quality control is that all work products have defined, measurable specifications to which we may compare the output of each process. The feedback loop is essential to minimize the defects produced.

Software Quality Assurance – Software Engineering

- **Software Quality Assurance (SQA)** is simply a way to assure quality in the software. It is the set of activities that ensure processes, procedures as well as standards are suitable for the project and implemented correctly.

Software Quality Assurance is a process that works parallel to Software Development. It focuses on improving the process of development of software so that problems can be prevented before they become major issues. Software Quality Assurance is a kind of Umbrella activity that is applied throughout the software process.

For those looking to deepen their expertise in SQA and elevate their professional skills, consider exploring a specialized training program – **Manual to Automation Testing: A QA Engineer's Guide** . This program offers practical, hands-on experience and advanced knowledge that complements the concepts covered in this guide.

Generally, the quality of the software is verified by third-party organizations like international standard organizations .

What is quality?

Quality in a product or service can be defined by several measurable characteristics. Each of these characteristics plays a crucial role in determining the overall quality.

Software Quality Assurance (SQA) en compasses

SQA process Specific quality assurance and quality control tasks (including technical reviews and a multitiered testing strategy) Effective software engineering practice (methods and tools) Control of all software work products and the changes made to them a procedure to ensure compliance with software development standards (when applicable) measurement and reporting mechanisms

Elements of Software Quality Assurance (SQA)

1. **Standards:** The IEEE, ISO, and other standards organizations have produced a broad array of software engineering standards and related documents. The job of SQA is to ensure that standards that have been adopted are followed and that all work products conform to them.
2. **Reviews and audits:** Technical reviews are a quality control activity performed by software engineers for software engineers. Their intent is to uncover errors. Audits are a type of review performed by SQA personnel (people employed in an organization) with the intent of ensuring that quality guidelines are being followed for software engineering work.
3. **Testing:** Software testing is a quality control function that has one primary goal—to find errors. The job of SQA is to ensure that testing is properly planned and efficiently conducted for primary goal of software.
4. **Error/defect collection and analysis** : SQA collects and analyzes error and defect data to better understand how errors are introduced and what software engineering activities are best suited to eliminating them.
5. **Change management:** SQA ensures that adequate change management practices have been instituted.
6. **Education:** Every software organization wants to improve its software engineering practices. A key contributor to improvement is education of software engineers, their managers, and other stakeholders. The SQA organization takes the lead in software process improvement which is key proponent and sponsor of educational programs.
7. **Security management:** SQA ensures that appropriate process and technology are used to achieve software security.
8. **Safety:** SQA may be responsible for assessing the impact of software failure and for initiating those steps required to reduce risk.
9. **Risk management** : The SQA organization ensures that risk management activities are properly conducted and that risk-related contingency plans have been established.

Software Quality Assurance (SQA) focuses

The Software Quality Assurance (SQA) focuses on the following

Software's portability: Software's **portability** refers to its ability to be easily transferred or adapted to different environments or platforms without needing significant modifications. This ensures that the software can run efficiently across various systems, enhancing its accessibility and flexibility.

- **software's usability: Usability** of software refers to how easy and intuitive it is for users to interact with and navigate through the application. A high level of usability ensures that users can effectively accomplish their tasks with minimal confusion or frustration, leading to a positive user experience.
- **software's reusability: Reusability** in software development involves designing components or modules that can be reused in multiple parts of the software or in different projects. This promotes efficiency and reduces development time by eliminating the need to reinvent the wheel for similar functionalities, enhancing productivity and maintainability.
- **software's correctness: Correctness** of software refers to its ability to produce the desired results under specific conditions or inputs. Correct software behaves as expected without errors or unexpected behaviors, meeting the requirements and specifications defined for its functionality.

- **software's maintainability: Maintainability** of software refers to how easily it can be modified, updated, or extended over time. Well-maintained software is structured and documented in a way that allows developers to make changes efficiently without introducing errors or compromising its stability.
- **software's error control: Error control** in software involves implementing mechanisms to detect, handle, and recover from errors or unexpected situations gracefully. Effective error control ensures that the software remains robust and reliable, minimizing disruptions to users and providing a smoother experience overall.

Software Quality Assurance (SQA) Include

1. A quality management approach.
2. Formal technical reviews.
3. Multi testing strategy.
4. Effective software engineering technology.
5. Measurement and reporting mechanism.

Major Software Quality Assurance (SQA) Activities

1. **SQA Management Plan:** Make a plan for how you will carry out the SQA throughout the project. Think about which set of software engineering activities are the best for project. check level of SQA team skills.
2. **Set The Check Points:** SQA team should set checkpoints. Evaluate the performance of the project on the basis of collected data on different check points.
3. **Measure Change Impact:** The changes for making the correction of an error sometimes re introduces more errors keep the measure of impact of change on project. Reset the new change to check the compatibility of this fix with whole project.
4. **Multi testing Strategy:** Do not depend on a single testing approach. When you have a lot of testing approaches available use them.
5. **Manage Good Relations:** In the working environment managing good relations with other teams involved in the project development is mandatory. Bad relation of SQA team with programmers team will impact directly and badly on project. Don't play politics.
6. **Maintaining records and reports:** Comprehensively document and share all QA records, including test cases, defects, changes, and cycles, for stakeholder awareness and future reference.
7. **Reviews software engineering activities:** The SQA group identifies and documents the processes. The group also verifies the correctness of software product.
8. **Formalize deviation handling:** Track and document software deviations meticulously. Follow established procedures for handling variances.

Benefits of Software Quality Assurance (SQA)

1. SQA produces high quality software.
2. High quality application saves time and cost.
3. SQA is beneficial for better reliability.
4. SQA is beneficial in the condition of no maintenance for a long time.
5. High quality commercial software increase market share of company.
6. Improving the process of creating software.
7. Improves the quality of the software.
8. It cuts maintenance costs. Get the release right the first time, and your company can forget about it and move on to the next big thing. Release a product with chronic issues, and your business bogs down in a costly, time-consuming, never-ending cycle of repairs.

Disadvantage of Software Quality Assurance (SQA)

There are a number of disadvantages of quality assurance.

- **Cost:** Some of them include adding more resources, which cause the more budget its not, Addition of more resources For betterment of the product.
- **Time Consuming:** Testing and Deployment of the project taking more time which cause delay in the project.
- **Overhead** : SQA processes can introduce administrative overhead, requiring documentation, reporting, and tracking of quality metrics. This additional administrative burden can sometimes outweigh the benefits, especially for smaller projects.
- **Resource Intensive** : SQA requires skilled personnel with expertise in testing methodologies, tools, and quality assurance practices. Acquiring and retaining such talent can be challenging and expensive.
- **Resistance to Change** : Some team members may resist the implementation of SQA processes, viewing them as bureaucratic or unnecessary. This resistance can hinder the adoption and effectiveness of quality assurance practices within an organization.
- **Not Foolproof** : Despite thorough testing and quality assurance efforts, software can still contain defects or vulnerabilities. SQA cannot guarantee the elimination of all bugs or issues in software products.
- **Complexity** : SQA processes can be complex, especially in large-scale projects with multiple stakeholders, dependencies, and integration points. Managing the complexity of quality assurance activities requires careful planning and coordination.

Software Reviews – Software Engineering

Software Review is a systematic inspection of software by one or more individuals who work together to find and resolve errors and defects in the software during the early stages of the Software Development Life Cycle (SDLC). A software review is an essential part of the Software Development Life Cycle (SDLC) that helps software engineers in validating the quality, functionality, and other vital features and components of the software. It is a whole process that includes testing the software product and it makes sure that it meets the requirements stated by the client.

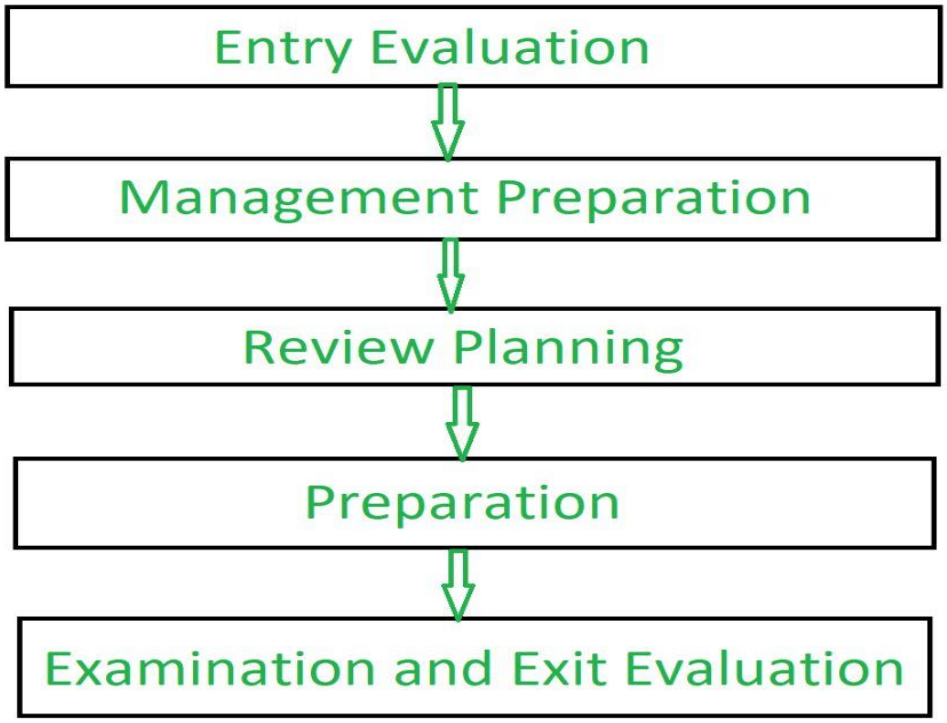
Usually performed manually, software review is used to verify various documents like requirements, system designs, codes, test plans, and test cases.

Objectives of Software Review

The objective of the software review is:

1. To improve the productivity of the development team.
2. To make the testing process time and cost-effective.
3. To make the final software with fewer defects.
4. To eliminate the inadequacies.

Process of Software Review



Software Review process

1. **Entry Evaluation:** By confirming documentation, fulfilling entry requirements and assessing stakeholder and team preparation, you can determine the software's availability.
2. **Management Preparation**: To get ready for the review process, assign roles, gather resources and provide brief management.
3. **Review Planning:** Establish the review's goals and scope, invite relevant parties and set a time for the meeting.
4. **Preparation**: Distribute appropriate resources, give reviewers time to get familiar and promote issue identification to help them prepare.
5. **Examination and Exit Evaluation:** Reviewers should collaborate to examine the results, record concerns, and encourage candid communication in meetings. It assess the results, make remedial plans based on flaws that have been reported and assess the process's overall efficacy.

Types of Software Reviews

There are mainly 3 types of software reviews:

1. Software Peer Review

Peer review is the process of assessing the technical content and quality of the product and it is usually conducted by the author of the work product along with some other developers. Peer review is performed in order to examine or resolve the defects in the software, whose quality is also checked by other members of the team.

Peer Review has following types:

1. **Code Review:** Computer source code is examined in a systematic way.
2. **Pair Programming:** It is a code review where two developers develop code together at the same platform.

3. **Walkthrough:** Members of the development team is guided by author and other interested parties and the participants ask questions and make comments about defects.
4. **Technical Review:** A team of highly qualified individuals examines the software product for its client's use and identifies technical defects from specifications and standards.
5. **Inspection:** In inspection the reviewers follow a well-defined process to find defects.

2. Software Management Review

Software Management Review evaluates the work status. In this section decisions regarding downstream activities are taken.

3. Software Audit Review

Software Audit Review is a type of external review in which one or more critics, who are not a part of the development team, organize an independent inspection of the software product and its processes to assess their compliance with stated specifications and standards. This is done by managerial level people.

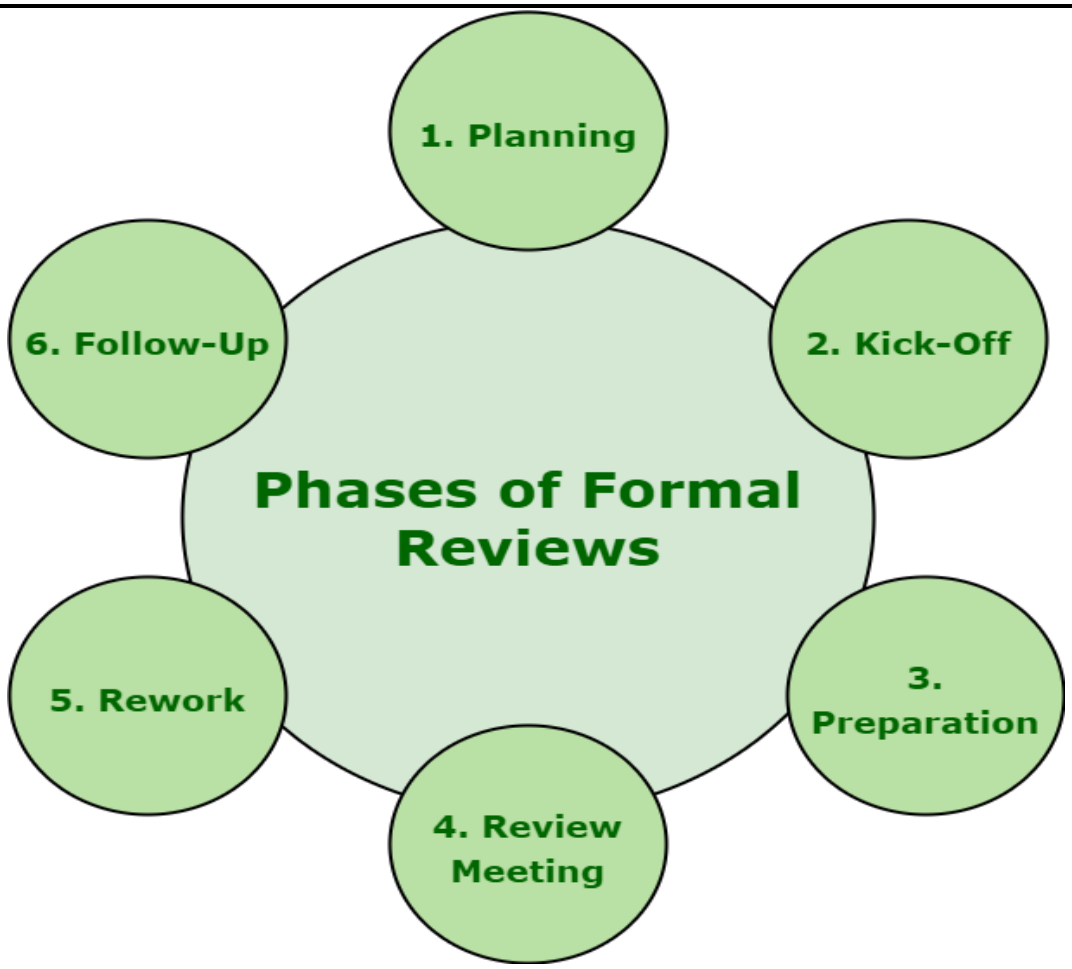
Advantages of Software Review

1. Defects can be identified earlier stage of development (especially in formal review).
2. Earlier inspection also reduces the maintenance cost of software.
3. It can be used to train technical authors.
4. It can be used to remove process inadequacies that encourage defects.

Formal Technical Reviews:

Formal Review generally takes place in piecemeal approach that consists of six different steps that are essential. Formal review generally obeys formal process. It is also one of the most important and essential techniques required in static testing.

Six steps are extremely essential as they allow team of developers simply to ensure and check software quality, efficiency, and effectiveness. These steps are given below :



Phases of Formal Reviews

1. **Planning :** For specific review, review process generally begins with 'request for review' simply by author to moderator or inspection leader. Individual participants, according to their understanding of document and role, simply identify and determine defects, questions, and comments. Moderator also performs entry checks and even considers exit criteria.
2. **Kick-Off :** Getting everybody on the same page regarding document under review is the main goal and aim of this meeting. Even entry result and exit criteria are also discussed in this meeting. It is basically an optional step. It also provides better understanding of team about relationship among document under review and other documents. During kick-off, Distribution of document under review, source documents, and all other related documentation can also be done.
3. **Preparation :** In preparation phase, participants simply work individually on document under review with the help of related documents, procedures, rules, and provided checklists. Spelling mistakes are also recorded on document under review but not mentioned during meeting. These reviewers generally identify and determine and also check for any defect, issue or error and offer their comments, that later combined and recorded with the assistance of logging form, while reviewing document.

4. **Review Meeting :** This phase generally involves three different phases i.e. logging, discussion, and decision. Different tasks are simply related to document under review is performed.
5. **Rework :** Author basically improves document that is under review based on the defects that are detected and improvements being suggested in review meeting. Document needs to be reworked if total number of defects that are found are more than an unexpected level. Changes that are done to document must be easy to determine during follow-up, therefore author needs to indicate changes are made.
6. **Follow-Up :** Generally, after rework, moderator must ensure that all satisfactory actions need to be taken on all logged defects, improvement suggestions, and change requests. Moderator simply makes sure that whether author has taken care of all defects or not. In order to control, handle, and optimize review process, moderator collects number of measurements at every step of process. Examples of measurements include total number of defects that are found, total number of defects that are found per page, overall review effort, etc.
7. **Individual Assessment:** The stage prior to the official group meeting during which each reviewer conducts an independent examination of the artefacts.
8. **Meeting for Group Review:** The cooperative stage in which the review panel discusses over results, resolves conflicts and makes choices regarding the examined artifacts.
9. **Finalization and Record-Keeping**: Completing the formal review procedure, recording the results and being ready for any necessary follow-up measures.
10. **Metrics and Ongoing Improvement:** Finding opportunities for ongoing improvement and evaluating the success of the formal review process through the tracking and analysis of review metrics.

Importance of Different Phases of Formal Review

1. **Early Defect Detection:** By catching errors early in the development process, formal reviews lower the effort and expense involved in fixing them.
2. **Knowledge Sharing:** Team members collaborate and share knowledge during various stages, which promotes a culture of ongoing learning and development.
3. **Process Improvement:** Through the formal review process, reoccurring issues are found, which helps development processes become more refined and improved over time.
4. **Quality Assurance:** By verifying that the programme complies with established standards and criteria, formal reviews make an important contribution to quality assurance.

5. STATISTICAL SOFTWARE QUALITY ASSURANCE

For software, statistical quality assurance implies the following steps:

1. Information about software defects is collected and categorized.
2. An attempt is made to trace each defect to its underlying cause (e.g., non-conformance to specifications, design error, violation of standards, poor communication with thecustomer).
3. Using the Pareto principle (80 percent of the defects can be traced to 20 percent of all possible causes), isolate the 20 percent (the "vital few").
4. Once the vital few causes have been identified, move to correct the problems that have caused the For software, statistical quality assurance implies the following steps:

.

The application of the statistical SQA and the pareto principle can be summarized in a single sentence: *spend your time focusing on things that really*

matter, but first be sure that you understand what really matters.

Six Sigma for software Engineering:

Six Sigma is the most widely used strategy for statistical quality assurance in industry today.

The term "six sigma" is derived from six standard deviations—3.4 instances (defects) per million occurrences—implying an extremely high quality standard.

The Six Sigma methodology defines three core steps:

1. ***Define*** customer requirements and deliverables and project goals via well-defined methods of customer communication
2. ***Measure*** the existing process and its output to determine current quality performance (collect defect metrics)
3. ***Analyze*** defect metrics and determine the vital few causes.

If an existing software process is in place, but improvement is required, Six Sigma suggests two additional steps.

4. ***Improve*** the process by eliminating the root causes of defects.
5. ***Control*** the process to ensure that future work does not reintroduce the causes of defects These core and additional steps are sometimes referred to as the DMAIC (define, measure, analyze, improve, and control) method.

If any organization is developing a software process (rather than improving and existing process), the core steps are augmented as follows:

6. ***Design*** the process to
    1. avoid the root causes of defects and
    2. to meet customer requirements
7. ***Verify*** that the process model will, in fact, avoid defects and meet customer requirements. This variation is sometimes called the DMADV (define, measure, analyze, design and verify) method.

Reliability Testing – Software Testing

Last Updated : 06 Aug, 2024

-

Reliability Testing is a testing technique that relates to testing the ability of software to function and given environmental conditions that help in uncovering issues in the software design and functionality.

This article focuses on discussing Reliability testing in detail.

What is Reliability Testing?

Reliability testing is a **Type of software testing** that evaluates the ability of a system to perform its intended function consistently and without failure over an extended period.

1. Reliability testing aims to identify and address issues that can cause the system to fail or become unavailable.
2. It is defined as a type of software testing that determines whether the software can perform a failure-free operation for a specific period in a specific environment.
3. It ensures that the product is fault-free and is reliable for its intended purpose.

4. It is an important aspect of software testing as it helps to ensure that the system will be able to meet the needs of its users over the long term.
5. It can also help to identify issues that may not be immediately apparent during functional testing, such as memory leaks or other performance issues.

Reliability testing Categories

The study of reliability testing can be divided into three categories:-

1. Modeling

Modeling in reliability testing involves creating mathematical or statistical representations of how a product or system might fail over time. It's like making an educated guess about the product's lifespan based on its design and components. This helps predict when and how failures might occur without actually waiting for the product to fail in real life.

Example: Engineers might create a model to estimate how long a new smartphone battery will last before it degrades significantly.

2. Measurement

Measurement focuses on collecting real-world data about a product's performance and failures. This involves testing products under various conditions and recording when and how they fail. It's about gathering concrete evidence of reliability rather than just predictions.

Example: A car manufacturer might test drive hundreds of cars for thousands of miles, recording any issues that arise during these tests.

3. Improvement

Improvement uses the insights gained from modeling and measurement to enhance the reliability of a product or system. This involves identifying weak points, redesigning components, or changing manufacturing processes to make the product more reliable.

Example: After finding that a particular part in a washing machine fails frequently, engineers might redesign that part or choose a more durable material to improve its lifespan.

Different Ways to Perform Reliability Testing

Here are the Different Ways to Perform Reliability Testing are follows:

1. **Stress testing:** Stress testing involves subjecting the system to high levels of load or usage to identify performance bottlenecks or issues that can cause the system to fail
2. **Endurance testing:** Endurance testing involves running the system continuously for an extended period to identify issues that may occur over time
3. **Recovery testing:** Recovery testing is testing the system's ability to recover from failures or crashes.
4. **Environmental Testing:** Conducting tests on the product or system in various environmental settings, such as temperature shifts, humidity levels, vibration exposure or shock exposure, helps in evaluating its dependability in real-world circumstances.
5. **Performance Testing:** In Performance Testing It is possible to make sure that the system continuously satisfies the necessary specifications and performance criteria by assessing its performance at both peak and normal load levels.
6. **Regression Testing:** In Regression Testing After every update or modification, the system should be tested again using the same set of test cases to help find any potential problems caused by code changes.

7. **Fault Tree Analysis:** Understanding the elements that lead to system failures can be achieved by identifying probable failure modes and examining the connections between them.

It is important to note that reliability testing may require specialized tools and test environments, and that it's often a costly and time-consuming process.

Objective of Reliability Testing

1. To find the perpetual structure of repeating failures.
2. To find the number of failures occurring is the specific period of time.
3. To discover the main cause of failure.
4. To conduct performance testing of various modules of software product after fixing defects.
5. It builds confidence in the market, stakeholders and users by providing a dependable product that meets quality criteria and operates as expected.
6. Understanding the dependability characteristics and potential mechanisms of failure of the system helps companies plan and schedule maintenance actions more efficiently.
7. It evaluates whether a system or product can be used continuously without experiencing a major loss in dependability, performance or safety.
8. It confirms that in the absence of unexpected shutdown or degradation, the system or product maintains constant performance levels under typical operating settings.

Types of Reliability Testing

Here are the **Types of Reliability Testing** are follows:

1. Feature Testing

Following three steps are involved in this testing:

- Each function in the software should be executed at least once.
- Interaction between two or more functions should be reduced.
- Each function should be properly executed.

2. Regression Testing

Regression testing is basically performed whenever any new functionality is added, old functionalities are removed or the bugs are fixed in an application to make sure with introduction of new functionality or with the fixing of previous bugs, no new bugs are introduced in the application.

3. Load Testing

Load testing is carried out to determine whether the application is supporting the required load without getting breakdown. It is performed to check the performance of the software under maximum work load.

4. Stress Testing

This type of testing involves subjecting the system to high levels of usage or load in order to identify performance bottlenecks or issues that can cause the system to fail.

5. Endurance Testing

This type of testing involves running the system continuously for an extended period of time in order to identify issues that may occur over time, such as memory leaks or other performance issues.

Recovery testing: This type of testing involves testing the system's ability to recover from failures or crashes, and to return to normal operation.

6. Volume Testing

Volume Testing is a type of testing involves testing the system's ability to handle large amounts of data. This type of testing is similar to endurance testing, but it focuses on the stability of the system under a normal, expected load over a long period of time.

7. Spike Testing

This type of testing involves subjecting the system to sudden, unexpected increases in load or usage in order to identify performance bottlenecks or issues that can cause the system to fail.

Measurement of Reliability Testing

Mean Time Between Failures (MTBF): Measurement of reliability testing is done in terms of mean time between failures (MTBF).

Mean Time To Failure (MTTF): The time between two consecutive failures is called as mean time to failure (MTTF).

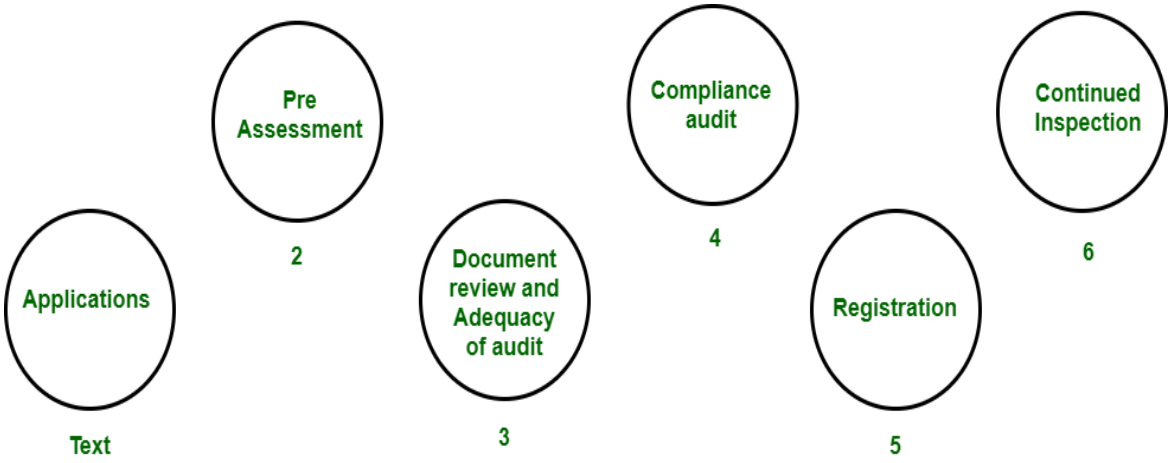
Mean Time To Repair (MTTR): The time taken to fix the failures is known as mean time to repair (MTTR).

MTBF = MTTF + MTTR

ISO 9000 Certification in Software Engineering

Last Updated : 23 Nov, 2020

- The International organization for Standardization is a world wide federation of national standard bodies. The **International standards organization (ISO)** is a standard which serves as a for contract between independent parties. It specifies guidelines for development of **quality system**. Quality system of an organization means the various activities related to its products or services. Standard of ISO addresses to both aspects i.e. operational and organizational aspects which includes responsibilities, reporting etc. An ISO 9000 standard contains set of guidelines of production process without considering product itself.



ISO 9000 Certification

Why ISO Certification required by Software Industry?

There are several reasons why software industry must get an ISO certification. Some of reasons are as follows :

- This certification has become a standards for international bidding.
- It helps in designing high-quality repeatable software products.
- It emphasis need for proper documentation.
- It facilitates development of optimal processes and totally quality measurements.

Features of ISO 9001 Requirements :

- **Document control** – All documents concerned with the development of a software product should be properly managed and controlled.
- **Planning** – Proper plans should be prepared and monitored.
- **Review** – For effectiveness and correctness all important documents across all phases should be independently checked and reviewed .
- **Testing** – The product should be tested against specification.
- **Organizational Aspects** – Various organizational aspects should be addressed e.g., management reporting of the quality team.

Advantages of ISO 9000 Certification :

Some of the advantages of the ISO 9000 certification process are following :

- Business ISO-9000 certification forces a corporation to specialize in "how they are doing business". Each procedure and work instruction must be documented and thus becomes a springboard for continuous improvement.
- Employees morale is increased as they're asked to require control of their processes and document their work processes
- Better products and services result from continuous improvement process.
- Increased employee participation, involvement, awareness and systematic employee training are reduced problems.

Shortcomings of ISO 9000 Certification :

Some of the shortcoming of the ISO 9000 certification process are following :

- ISO 9000 does not give any guideline for defining an appropriate process and does not give guarantee for high quality process.
- ISO 9000 certification process have no international accreditation agency exists.