

# Unit-7

## Metrics for Process and Product

- Software Measurement
- Metrics for software quality
- Software Quality Assurance
- Software Reliability
- The ISO 9000 quality standards

# Software Measurement

- A measurement is a manifestation of the size, quantity, amount, or dimension of a particular attribute of a product or process.
- Software measurement is a titrate impute of a characteristic of a software product or the software process.

## Software Measurement Principles

- The software measurement process can be characterized by five activities:
  - ❑ **Formulation:** The derivation of software measures and metrics appropriate for the representation of the software that is being considered.
  - ❑ **Collection:** The mechanism used to accumulate data required to derive the formulated metrics.
  - ❑ **Analysis:** The computation of metrics and the application of mathematical tools.
  - ❑ **Interpretation:** The evaluation of metrics results in insight into the quality of the representation.
  - ❑ **Feedback:** Recommendation derived from the interpretation of product metrics transmitted to the software team.

# Need for Software Measurement

Software is measured to:

- ☐ Create the quality of the current product or process.
- ☐ Anticipate future qualities of the product or process.
- ☐ Enhance the quality of a product or process.
- ☐ Regulate the state of the project concerning budget and schedule.
- ☐ Enable data-driven decision-making in project planning and control.
- ☐ Identify bottlenecks and areas for improvement to drive process improvement activities.
- ☐ Ensure that industry standards and regulations are followed.
- ☐ Give software products and processes a quantitative basis for evaluation.
- ☐ Enable the ongoing improvement of software development practices.

# Software Metrics

- A metric is a measurement of the level at which any impute belongs to a system product or process.
- Software metrics are a quantifiable or countable assessment of the attributes of a software product.
- There are 4 functions related to software metrics:
  - **Planning**
  - **Organizing**
  - **Controlling**
  - **Improving**

## Characteristics of software Metrics

- **Quantitative:** Metrics must possess a quantitative nature. It means metrics can be expressed in numerical values.
- **Understandable:** Metric computation should be easily understood, and the method of computing metrics should be clearly defined.
- **Applicability:** Metrics should be applicable in the initial phases of the development of the software.
- **Repeatable:** When measured repeatedly, the metric values should be the same and consistent.
- **Economical:** The computation of metrics should be economical.
- **Language Independent:** Metrics should not depend on any programming language.

# Types of Software Metrics

## ❑ Product Metrics:

- Product metrics are used to evaluate the state of the product, tracing risks and undercover prospective problem areas.
- The ability of the team to control quality is evaluated.
- Examples include lines of code, cyclomatic complexity, code coverage, defect density, and code maintainability index.

## ❑ Process Metrics:

- Process metrics pay particular attention to enhancing the long-term process of the team or organization.
- These metrics are used to optimize the development process and maintenance activities of

software.

- Examples include effort variance, schedule variance, defect injection rate, and lead time.

## ❑ Project Metrics:

- The project metrics describes the characteristic and execution of a project.
- Examples include effort estimation accuracy, schedule deviation, cost variance, and productivity.
- Usually measures:
  - Number of software developer
  - Staffing patterns over the life cycle of software
  - Cost and schedule
  - Productivity

## **Advantages of Software Metrics**

- Reduction in cost or budget.
- It helps to identify the particular area for improvising.
- It helps to increase the product quality.
- Managing the workloads and teams.
- Reduction in overall time to produce the product,.
- It helps to determine the complexity of the code and to test the code with resources.
- It helps in providing effective planning, controlling and managing of the entire product.

## **Disadvantages of Software Metrics**

- It is expensive and difficult to implement the metrics in some cases.
- Performance of the entire team or an individual from the team can't be determined. Only the performance of the product is determined.
- Sometimes the quality of the product is not met with the expectation.
- It leads to measure the unwanted data which is wastage of time.
- Measuring the incorrect data leads to make wrong decision making.

# Software quality metrics

- Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project.
- These are more closely associated with process and product metrics than with project metrics.
- Software quality metrics can be further divided into three categories:
  - ❑ Product quality metrics
  - ❑ In-process quality metrics
  - ❑ Maintenance quality metrics

## 1. Product Quality Metrics

- Product quality metrics focus on evaluating the quality of the software product itself.
- These metrics help assess the functionality, reliability, maintainability, and overall user satisfaction of the software.
- This metrics include the following
  - Mean Time to Failure
  - Defect Density
  - Customer Problems
  - Customer Satisfaction
- ❑ **Mean Time to Failure**
  - It is the time between failures. This metric is mostly used with safety critical systems such as the airline traffic control systems, avionics, and weapons.
- ❑ **Defect Density**
  - It measures the defects relative to the software size expressed as lines of code or function point, etc.

i.e., it measures code quality per unit. This metric is used in many commercial software systems.

### ❑ **Customer Problems**

- It measures the problems that customers encounter when using the product. It contains the customer's perspective towards the problem space of the software, which includes the non-defect oriented problems together with the defect problems.

### ❑ **Customer Satisfaction**

- Customer satisfaction is often measured by customer survey data through the five-point scale :
  - Very satisfied
  - Satisfied
  - Neutral
  - Dissatisfied
  - Very dissatisfied



## 2. In-process Quality Metrics

- In-process quality metrics deals with the tracking of defect arrival during formal machine testing for some organizations.

This metric includes

- Defect density during machine testing
- Defect arrival pattern during machine testing
- Phase-based defect removal pattern
- Defect removal effectiveness

### Defect density during machine testing

- Defect rate during formal machine testing (testing after code is integrated into the system library) is correlated with the defect rate in the field.
- Higher defect rates found during testing is an indicator that the software has experienced higher error injection during its development process, unless the higher testing defect rate is due to an extraordinary testing effort.

### ❑ Defect arrival pattern during machine testing

- The overall defect density during testing will provide only the summary of the defects.
- The pattern of defect arrivals gives more information about different quality levels in the field.

### ❑ Phase-based defect removal pattern

- This is an extension of the defect density metric during testing.
- In addition to testing, it tracks the defects at all phases of the development cycle, including the design reviews, code inspections, and formal verifications before testing.

### ❑ Defect removal effectiveness

- This metric can be calculated for the entire development process, for the front-end before code integration and for each phase.
- It is called **early defect removal** when used for the front-end and **phase effectiveness** for specific phases.

### 3. Maintenance Quality Metrics

- Although much cannot be done to alter the quality of the product during this phase, following are the fixes that can be carried out to eliminate the defects as soon as possible with excellent fix quality.
  - Fix backlog and backlog management index
  - Fix response time and fix responsiveness
  - Percent delinquent fixes
  - Fix quality
- ❑ Fix backlog and backlog management index
  - Fix backlog is related to the rate of defect arrivals and the rate at which fixes for reported problems become available.
  - It is a simple count of reported problems that remain at the end of each month or each week.

### ❑ Fix response time and fix responsiveness

- The fix response time metric is usually calculated as the mean time of all problems from open to close.
- Short fix response time leads to customer satisfaction.

### ❑ Percent delinquent fixes

It is calculated as follows –

*Percent Delinquent Fixes =*

$$\frac{\text{Number of fixes that exceeded the response time criteria by severity level}}{\text{Number of fixes delivered in a specified time}} \times 100\%$$

### ❑ Fix Quality

- Fix quality or the number of defective fixes is another important quality metric for the maintenance phase.
- A fix is defective if it did not fix the reported problem, or if it fixed the original problem but injected a new defect.

# Software Quality Assurance (SQA)

- 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.

# Elements of Software Quality Assurance (SQA)

- ❑ **Standards:** The IEEE, ISO, and other standards organizations have produced a broad array of software engineering standards and related documents.
- ❑ **Reviews and audits:** Technical reviews are a quality control activity performed by software engineers for software engineers. Their intent is to uncover errors.
- ❑ **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.
- ❑ **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.
- ❑ **Change management:** SQA ensures that adequate change management practices have been instituted.
- ❑ **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.
- ❑ **Security management:** SQA ensures that appropriate process and technology are used to achieve software security.
- ❑ **Safety:** SQA may be responsible for assessing the impact of software failure and for initiating those steps required to reduce risk.
- ❑ **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.

### ❑ software's usability:

- **Usability** of software refers to how easy and intuitive it is for users to interact with and navigate through the application.

### ❑ 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.

### ❑ 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.

## **Software Quality Assurance (SQA) Include**

- ❑ A quality management approach.
- ❑ Formal technical reviews.
- ❑ Multi testing strategy.
- ❑ Effective software engineering technology.
- ❑ Measurement and reporting mechanism.

## **Major Software Quality Assurance (SQA) Activities**

- ❑ **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.
- ❑ **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.
- ❑ **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.
- ❑ **Multi testing Strategy:** Do not depend on a single testing approach. When you have a lot of testing approaches available use them.
- ❑ **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.
- ❑ **Maintaining records and reports:** Comprehensively document and share all QA records, including test cases, defects, changes, and cycles, for stakeholder awareness and future reference.
- ❑ **Reviews software engineering activities:** The SQA group identifies and documents the processes. The group also verifies the correctness of software product.
- ❑ **Formalize deviation handling:** Track and document software deviations meticulously. Follow established procedures for handling variances.



# software quality assurance plan's

- A software quality assurance plan's main goal is to guarantee that the market's product or service is trouble- and bug-free.
- Additionally, it must fulfill the specifications listed in the SRS (software requirement specification).
- ❑ **An SQA plan serves three purposes. It includes the following:**
  - Determining the QA duties assigned to the concerned team.
  - A list of the areas that require review, audit, and examination.
  - Determines the work products for SQA.

## Importance of Software Quality

### Assurance Plan

- ❑ **Quality Standards and Guidelines:** The SQA Plan lays out the requirements and guidelines to make sure the programme satisfies predetermined standards for quality.
- ❑ **Risk management:** It is the process of recognizing, evaluating and controlling risks in order to reduce the possibility of errors and other problems with quality.
- ❑ **Standardization and Consistency:** The strategy guarantees consistent methods, processes, and procedures, fostering a unified and well-structured approach to quality assurance.
- ❑ **Customer Satisfaction:** The SQA Plan helps to ensure that the finished product satisfies customer needs, which in turn increases overall customer satisfaction.
- ❑ **Resource optimization:** It is the process of defining roles, responsibilities, and procedures in order to maximize resource utilization and minimize needless rework.
- ❑ **Early Issue Detection:** SQA Plans help identify problems early on, which lowers the expense and work involved in fixing them.

## Objectives And Goals of Software Quality Assurance Plan:

- The objectives and goals of a Quality Assurance Plan (QAP) are to ensure that the products or services meet specified quality standards and requirements.
- ❑ **Compliance with Standards and Regulations:**
  - **Objective:** Ensure that the project or product complies with relevant industry standards, regulatory requirements, and any other applicable guidelines.
  - **Goal:** Achieve and maintain adherence to established quality standards to meet legal and regulatory obligations.
- ❑ **Customer Satisfaction:**
  - **Objective:** Enhance customer satisfaction by delivering products or services that meet or exceed customer expectations.
  - **Goal:** Identify and prioritize customer requirements, and incorporate them into the quality assurance processes to create a positive customer experience.
- ❑ **Defect Prevention:**
  - **Objective:** Implement measures to prevent defects, errors, or issues in the early stages of the project lifecycle.
  - **Goal:** Identify potential sources of defects, analyze root causes, and take proactive steps to eliminate or minimize the occurrence of defects.
- ❑ **Consistency and Reliability:**
  - **Objective:** Establish a consistent and reliable process for the development or delivery of products and services.
  - **Goal:** Ensure that the quality of deliverables is consistent over time and across different phases of the project, promoting reliability and predictability.

#### ❑ **Process Improvement:**

- **Objective:** Continuously improve processes to enhance efficiency, effectiveness, and overall quality.
- **Goal:** Implement feedback mechanisms, conduct regular process assessments, and identify opportunities for improvement to optimize the quality assurance process.

#### ❑ **Risk Management:**

- **Objective:** Identify and manage risks that could impact the quality of the project or product.
- **Goal:** Develop strategies to assess, mitigate, and monitor risks that may affect the achievement of quality objectives.

#### ❑ **Clear Roles and Responsibilities:**

- **Objective:** Clearly define roles and responsibilities related to quality assurance activities.
- **Goal:** Ensure that team members understand their roles in maintaining and improving quality, fostering accountability and collaboration.

#### ❑ **Documentation and Traceability:**

- **Objective:** Establish a robust documentation process to track and trace quality-related activities and decisions.
- **Goal:** Create comprehensive records that enable

transparency, accountability, and the ability to trace the development or production process.

#### ❑ **Training and Competence:**

- **Objective:** Ensure that team members are adequately trained and possess the necessary competencies to perform quality assurance tasks.
- **Goal:** Provide ongoing training to enhance the skills and knowledge of individuals involved in quality assurance.

#### ❑ **Continuous Monitoring and Reporting:**

- **Objective:** Monitor quality metrics and report on the status of quality assurance activities.
- **Goal:** Implement regular monitoring and reporting mechanisms to track progress, identify issues, and make data-driven decisions to maintain or improve quality.

## Steps to Develop Software Quality Assurance Plan:

- Developing a Quality Assurance Plan (QAP) involves a systematic process to ensure that the plan effectively addresses the quality requirements of a project or process. Here are the steps to develop a Quality Assurance Plan:

### ❑ Define Project Objectives and Scope:

- Clearly articulate the objectives and scope of the project or process for which the Quality Assurance Plan is being developed. Understand the goals, deliverables, and stakeholders involved.

### ❑ Identify Quality Standards and Criteria:

- Determine the relevant quality standards, regulations, and criteria that are applicable to the project. This may include industry standards, legal requirements, and internal organizational standards.

### ❑ Identify Stakeholders:

- Identify and involve key stakeholders who have an interest in or will be affected by the quality of the project or process. This includes project managers, team members, customers, and any regulatory bodies.

### ❑ Define Roles and Responsibilities:

- Clearly define the roles and responsibilities of individuals or teams involved in quality assurance activities. This ensures accountability and clarity in the execution of quality-related tasks.

### ❑ Conduct a Risk Assessment:

- Identify potential risks to the quality of the project. Assess the impact and likelihood of these risks and develop strategies for risk mitigation and management.

### ❑ Establish Quality Assurance Activities:

- Determine the specific activities and tasks that will be carried out to ensure quality throughout the project lifecycle. This may include reviews, inspections, audits, testing, and other quality control measures.

### ❑ Develop Testing and Inspection Procedures:

- If applicable, create detailed testing and inspection procedures. This includes developing test plans, test cases, and acceptance criteria to ensure that the product or service meets quality standards.

### ❑ **Document Processes and Procedures:**

- Document the processes and procedures that will be followed to implement quality assurance activities. This documentation serves as a reference for team members and provides a basis for consistency.

### ❑ **Establish Documentation and Reporting Mechanisms:**

- Define the documentation requirements for tracking and reporting quality-related information. This may include reports, checklists, and records of quality assessments.

### ❑ **Allocate Resources and Training:**

- Specify the resources, tools, and training that will be provided to team members to support quality assurance efforts. Ensure that team members have the necessary skills and knowledge.

### ❑ **Define Change Control Processes:**

- Establish a process for managing changes to the project or product to ensure that they do not negatively impact

quality. This includes a change control board and a structured process for reviewing and approving changes.

### ❑ **Review and Approval:**

- Seek input and feedback from relevant stakeholders on the draft Quality Assurance Plan. Revise the plan based on feedback and obtain formal approval before implementation.

### ❑ **Monitoring and Continuous Improvement:**

- Continuously monitor quality metrics, track progress, and identify opportunities for improvement. Regularly review and update the Quality Assurance Plan to adapt to changes and enhance its effectiveness.

### ❑ **Communication and Training:**

- Communicate the Quality Assurance Plan to all relevant stakeholders and provide training as necessary. Ensure that everyone involved understands their roles and responsibilities in maintaining and improving quality.

## **Benefits of Software Quality Assurance (SQA)**

- SQA produces high quality software.
- High quality application saves time and cost.
- SQA is beneficial for better reliability.
- SQA is beneficial in the condition of no maintenance for a long time.
- High quality commercial software increase market share of company.
- Improving the process of creating software.
- Improves the quality of the software.
- It cuts maintenance costs.

# Software Reliability

- Software Reliability means Operational reliability.
- It is described as the ability of a system or component to perform its required functions under static conditions for a specific period.
- Software reliability is also defined as the probability that a software system fulfills its assigned task in a given environment for a predefined number of input cases, assuming that the hardware and the input are free of error.
- Software Reliability is an essential connect of software quality, composed with functionality, usability, performance, serviceability, capability, installability, maintainability, and documentation.
- Software Reliability is hard to achieve because the complexity of software turn to be high.



## Applications of Software Reliability

- The applications of software reliability includes
  - **Comparison of software engineering technologies.**
    - What is the cost of adopting a technology?
    - What is the return from the technology : in terms of cost and quality?
  - **Measuring the progress of system testing :**The failure intensity measure tells us about the present quality of the system: high intensity means more tests are to be performed.
  - **Controlling the system in operation :**The amount of change to a software for maintenance affects its reliability.
  - **Better insight into software development processes :**Quantification of quality gives us a better insight into the development processes.

# International standards organization (ISO)

- 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 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

- ISO 9000 is defined as a set of international standards on quality management and quality assurance developed to help companies effectively document the quality system elements needed to maintain an efficient quality system.
- They are not specific to any one industry and can be applied to organizations of any size.
- ISO 9000 can help a company satisfy its customers, meet regulatory requirements, and achieve continual improvement.
- It should be considered to be a first step or the base level of a quality system.



## ➤ **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.