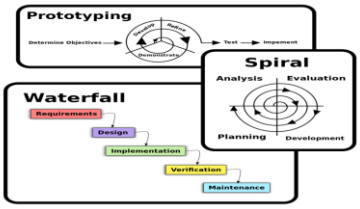
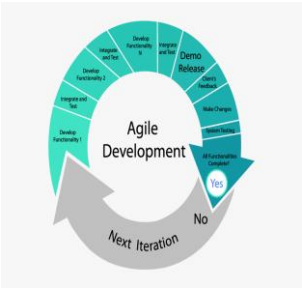
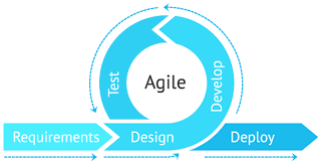


Software Engineering Quality Assurance





Outline

- Verification and Validation
- Software Quality Assurance
- Software Quality
- Capability Maturity Model (SEI-CMM)
- International Standard Organization (ISO)
- Comparison of ISO-9000 Certification
- Reliability Issues
- Reliability Metrics.

Verification & Validation

Verification

Are we building the product right?

The objective of Verification is to make sure that the product being develop is as per the requirements and design specifications.

Validation

Are we building the right product?

The objective of Validation is to make sure that the product actually meet up the user's requirements, and check whether the specifications were correct in the first place.

3

Verification & Validation

Verification	Validation
Process of **evaluating** products of a **development phase** to find out whether they meet the specified requirements.	Process of evaluating software **at the end of the development** to determine whether software meets the customer expectations and requirements.
Activities involved: **Reviews**, **Meetings** and **Inspections**	Activities involved: **Testing** like black box testing, white box testing, gray box testing
Carried out by **QA team**	Carried out by **testing team**
Execution of code is **not comes** under Verification	**Execution** of code is **comes** under Validation
Explains whether the **outputs** are **according to inputs** or not	**Describes** whether the **software** is **accepted by** the **user** or not
Cost of **errors** caught is **less**	**Cost** of **errors** caught is **high**

What is Quality

- **Quality :**
  - Developed product meets it's specification
- **Quality Management**
  - Ensuring that required level of product quality is achieved
  - Defining procedures and standards
  - Applying procedures and standards to the product and process
  - Checking that procedures are followed
  - Collecting and analyzing various quality data

5

Two kinds of Quality

- **Quality of Design:** Quality of Design refers to the characteristics that designers specify for an item. The grade of materials, tolerances, and performance specifications that all contribute to the quality of design.
- **Quality of conformance:** Quality of conformance is the degree to which the design specifications are followed during manufacturing. Greater the degree of conformance, the higher is the level of quality of conformance.

6

Software Quality

- **Software Quality remains an issue**
- **Who is to blame?**
- **Customers blame developers**
  - Arguing that careless practices lead to low-quality software
- **Developers blame Customers & other stakeholders**
  - Arguing that irrational delivery dates and continuous stream of changes force the to deliver software before it has been fully validated

Who is Right? Both – and that's the problem

7

Software Quality

- **Software Quality:** Software Quality is defined as the conformance to explicitly state functional and performance requirements, explicitly documented development standards, and inherent characteristics that are expected of all professionally developed software.
- **Quality Control:** Quality Control involves a series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements place upon it. Quality control includes a feedback loop to the process that created the work product.
- **Quality Assurance:** Quality Assurance is the preventive set of activities that provide greater confidence that the project will be completed successfully.

8

Software Quality Assurance (SQA)

- **Software quality assurance** (also called **quality management**) is an **umbrella activity** that is applied throughout the software process
- It is planned and systematic pattern of **activities** necessary to **provide high degree of confidence in the quality**
- Software quality assurance **(SQA) encompasses**
  - An **SQA process**
  - Specific **quality assurance** and quality **control tasks**
  - **Effective software engineering** practice
  - **Control** of all software **work products**
  - A **procedure** to **ensure compliance** with software **development standards**
  - **Measurement** and **reporting mechanisms**

9

SQA Activities

- **Prepare** an **SQA plan** for a project
- **Participate** in the **development** of the project's **software process** description
- **Review** software engineering **activities** to **verify compliance** with the **defined software process**
- **Audit** designated **software work products** to verify compliance with those defined as part of the software process
- **Ensure** that **deviations** in software **work** and **work products** are **documented** and **handled** according to a documented procedure
- **Records** any **noncompliance** and reporting **to senior management**

10

Differentiation of SQA & SQC

Criteria	SQA	SQC
Definition	**SQA** is a **set of activities** for **ensuring quality in** software engineering **processes** (that ultimately result in quality in software products). The activities establish and evaluate the processes that produce products	**SQC** is a **set of activities** for **ensuring quality** in software **products**. The activities focus on identifying defects in the actual products produced
Focus	**Process** focused	**Product** focused
Orientation	**Prevention** oriented	**Detection** oriented
Breadth	**Organization** wide	**Product/project** specific
Scope	**Relates to all products** that will ever be **created by a process**	**Relates** to **specific product**
Activities	Process **Definition** and **Implementation**, **Audits**, **Training**	**Reviews**, **Testing**

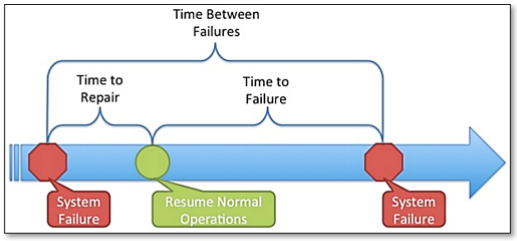
11

Software Reliability

- Software **reliability** is defined **in statistical terms**

> The **probability** of **failure-free operation** of a **computer program** in a **specified environment** for a **specified time**

- A simple **measure** of **reliability** is **meantime-between-failure** (MTBF):



MTBF = MTTF + MTTR

MTTF = mean-time-to-failure, **MTTR** = mean-time-to-repair

12

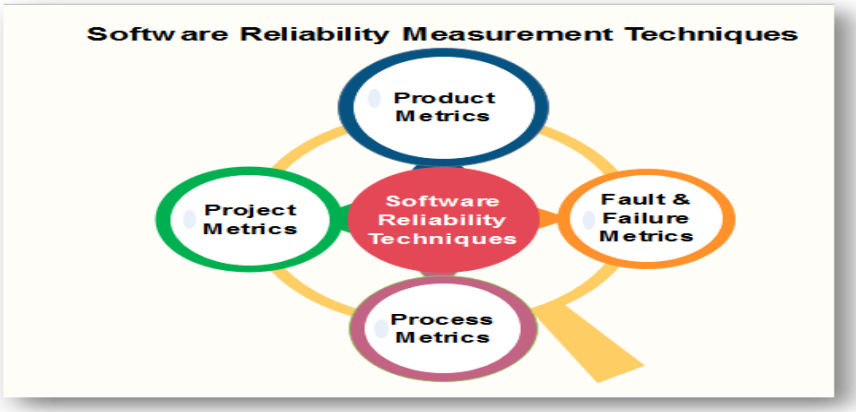
Measures of Reliability

- Many researchers **argue** that **MTBF** is a far more useful measure than other quality-related software metrics
- An **end user** is **concerned** with **failures**, **not with** the total **defect count**
- Because **each defect** contained within a program **does not** have the **same failure rate**, the total defect count provides little indication of the reliability of a system
- An alternative measure of reliability is **failures-in-time (FIT)**
  - a statistical measure of **how many failures** a component will have **over one billion hours** of **operation**

13

Software Reliability Measurement Techniques

- Reliability metrics are used to quantitatively express the reliability of the software product.
- The option of which parameter is to be used depends upon the type of system to which it applies & the requirements of the application domain.



Software Safety

- **Software safety** is a software **quality** assurance **activity**
  - that **focuses** on the **identification** and **assessment** of potential **hazards** that may **affect** software **negatively** and **cause** an entire **system to fail**
- If **hazards** can be **identified early** in the software process,
  - software **design features** can be **specified** that will either eliminate or control potential hazards
- A **modelling** and **analysis** process is **conducted** as **part of** software **safety**
- Initially, **hazards** are **identified** and **categorized** by criticality and risk

15

Software Safety

- Although software **reliability** and software **safety** are **closely related** to one another, it is important to understand the subtle difference between them
  - **Software reliability** uses **statistical analysis** to determine the **likelihood that a software failure will occur**
  - However, the **occurrence of a failure** does **not necessarily** result **in a hazard** or accident
  - **Software safety** examines **the ways** in which **failures** can **lead to an accident**

16

The quality standards

ISO 9000 and 9001
Six Sigma
CMM

17

ISO 9001

- In order to **bring quality** in **product** and **service**, many organizations are **adopting Quality Assurance System**
- **ISO standards** are **issued by** the **International Organization for Standardization (ISO)** in **Switzerland**
- **Proper documentation** is **an important part** of an ISO 9001 Quality Management System.
- **ISO 9001** is the **quality assurance standard** that **applies** to **software engineering**
- It **includes**, **requirements** that must be present for an effective quality assurance system
- ISO 9001 standard is **applicable to all engineering discipline**

18

ISO 9001

- In order for a **software organization** to become **registered to ISO 9001:2000**

1. It must **establish policies** and **procedures** to **address** each of the **requirements** just **noted**
2. **Able to demonstrate** that these **policies** and **procedures** are being **followed**

ISO

- The types of industries to which the various ISO standards apply are : ISO 9001, ISO 9002, and ISO 9003
- ISO 9001: This standard applies to the organizations engaged in design, development, production, and servicing of goods. This is the standard that applies to most software development organizations.
- ISO 9002: This standard applies to those organizations which do not design products but are only involved in the production. Examples of these category industries contain steel and car manufacturing industries that buy the product and plants designs from external sources and are engaged in only manufacturing those products. Therefore, ISO 9002 does not apply to software development organizations.
- ISO 9003: This standard applies to organizations that are involved only in the installation and testing of the products. For example, Gas companies.

21

Six Sigma

- Several Software Packages available to assist in measuring yield, defects per million opportunities, etc.
- Six sigma is "**A generic quantitative approach to improvement that applies to any process**"
- **Six Sigma** is a disciplined, **data-driven approach** and **methodology** for **eliminating defects** in **any process** - from manufacturing to transactional and from product to service
- To **achieve six sigma**, a **process** must **not produce more than 3.4 defects per million opportunities**
  - **4 Sigma → 6210** defects per million opportunities
  - **5 Sigma → 230** defects per million opportunities
- Six sigma have two methodologies.
  - **DMAIC** (Define, Measure, Analyze, Improve, Control)
  - **DMADV** (Define, Measure, Analyze, Design, Verify)

22

DMAIC - Six Sigma

- **Define:** Define the problem or process to improve upon related to the customer and goals
- **Measure:** How can you measure this process in a systematic way?
- **Analyze:** Analyze the process or problem and identify the way in which it can be improved. What are the root causes of problems within the process?
- **Improve:** Once you know the causes of the problems, present solutions for them and implement them.
- **Control:** Utilize Statistical Process Control to continuously measure your results and ensure you are improving

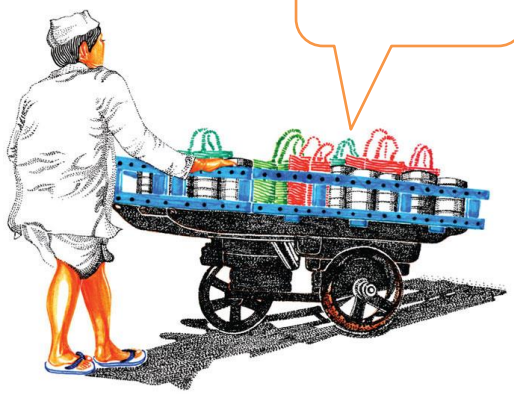
23

DMADV - Six Sigma

- **Define**, **Measure** and **analyze are similar to above method**
- **Design: Avoid root causes** of **defects** and **meet** the customer **requirements**
- **Verify:** To **verify** the **process**, **compare** the **process** with the **standard plan** and **find differences**

Example of Six Sigma Company

Mumbai's Dabbawalas



For **over 100 years** they have **delivered food** to **every part of the city**, earning them a **Six Sigma rating** (a Forbes rating of 99.9 % which **means one error in 6 million transactions**).

24

CMM (Capability Maturity Model)

- To **determine** an **organization's current state** of **process maturity**, the **SEI (Software Engineering Institute)** uses an assessment that results in a **five point grading scheme**
- The grading scheme determines compliance with a **capability maturity model (CMM)** that **defines key activities required** at **different levels** of **process maturity**
- The **SEI** approach establishes **five process maturity levels** that are defined in the following manner
  - **Level 1: Initial**
    - The software **process** is **characterized** as **ad hoc** and **occasionally**
    - Few processes are defined and **success depends** on **individual effort**

25

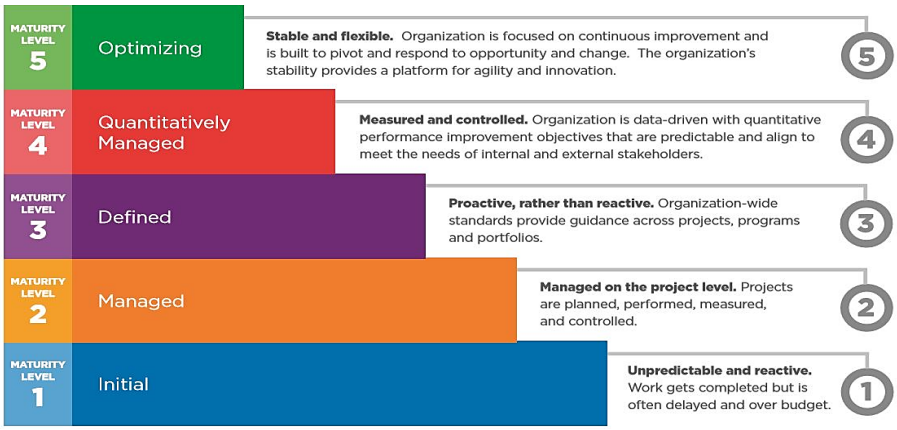
CMM (Capability Maturity Model)

  - **Level 2: Repeatable**
    - Basic project management **processes** are **established** to track cost, schedule, and functionality.
    - The necessary **process discipline** is in place to **repeat earlier successes** on Project
  - **Level 3: Defined**
    - The software **process** for **both management** and **engineering** activities is **documented**, **standardized** and **integrated**
    - This level **includes** all **characteristics** defined for **level 2**
  - **Level 4: Managed**
    - Detailed **measures** of the software **process** and **product quality** are **collected**
    - This level **includes** all **characteristics** defined for **level 3**

26

CMM (Capability Maturity Model)

- **Level 5: Optimizing**
  - Continuous **process improvement** is **enabled** by quantitative **feedback** from the **process** and **from testing innovative ideas** and **technologies**
  - This level **includes** all **characteristics** defined for **level 4**



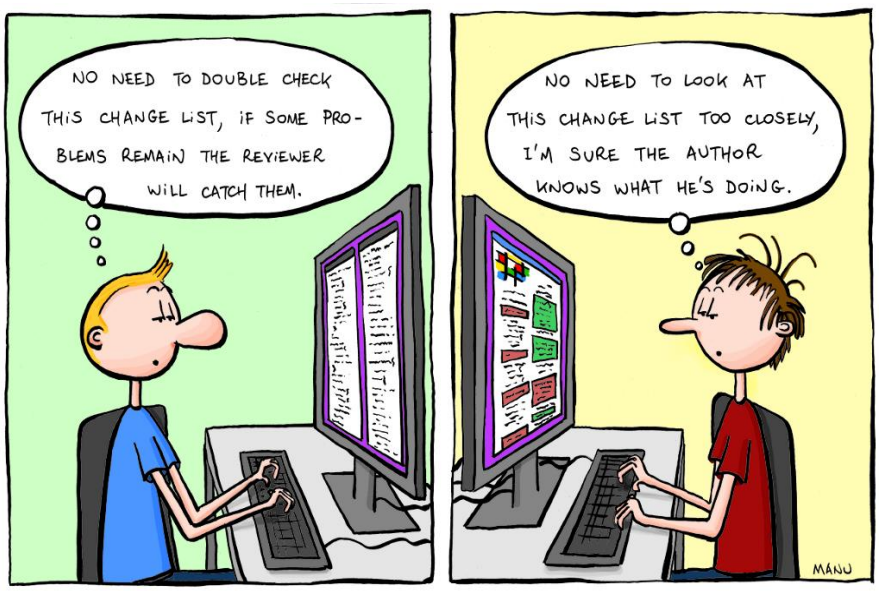
Software Testing

Testing is the **process** of exercising a program with the specific **intent of finding errors** prior to delivery to the end user.

Don't view testing as a **"safety net"** that will catch all errors that occurred because of weak software engineering practice.

28

Software Testing



29

30