Software Verification and Validation SENG 22233

MISS K.THADSAYINI
BSc (Hons) in Sofware Engineering, UOK
Temporary Lecturer, UOK
Visiting Instructor, NAITA

Software Testing Terms and Definitions

Precision and Accuracy

Verification and Validation

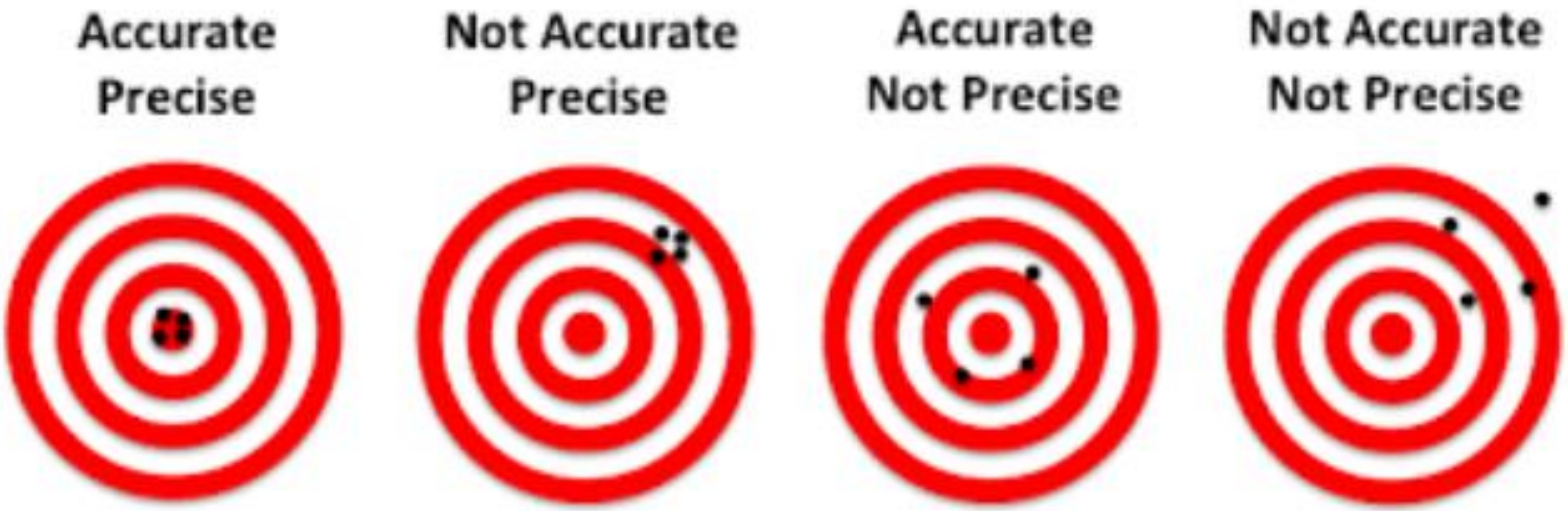
Quality and Reliability

Testing and Quality Assurance

Precision and Accuracy

Accuracy refers to the closeness of a measured value to a standard or known value. Must close to target. Can be determine by one measurement.

Precision refers to the closeness of two or more measurements to each other. Must be close to each other


Accurate
Precise
Not Accurate
Precise
Accurate
Not Precise
Not Accurate
Not Precise

Verification and Validation

Verification is the process confirming that something software meets its specification.
Ensures that “**are we building product right**”

Validation is the process confirming that it meets the user’s requirements and expectations.
Ensures that “**we have developed right product**”

Quality and Reliability

Quality –

“a degree of excellence” or “superiority in kind.”

If a software product is of high quality, it will meet the customer’s needs.

Reliability –

"quality changing over time" .

shows how well this object maintains its original level of quality over time, through various conditions.

Testing and Quality Assurance (QA)

The goal of a **software tester** is to find bugs, find them as early as possible, and make sure they get fixed.

A **software quality assurance (SQA)** person's main responsibility is to create and enforce standards and methods to improve the development process and to prevent bugs from ever occurring.

Effects related to the software product

Product Specification

Schedules

Design Documents

Product Reviews

Feedback from Previous Versions

Customer Surveys

CODES

Test Plans

Customer Requirements

Software is written to fulfill some need that a person or a group of people has

Find out what the customer wants:

1. simply guess
2. Surveys
3. feedback from previous versions of the software
4. competitive product information
5. magazine reviews
6. focus groups

Specification

The result of the customer requirements studies is really just raw data. It doesn’t describe the proposed product, it just confirms whether it should (or shouldn’t) be created and what features the customers want.

The specifications take all this information plus any unstated but mandatory requirements and truly define what the product will be, what it will do, and how it will look.

Schedules

Necessary to have some mechanism to track its progress

1. simple task lists
2. Gantt charts
3. detailed tracking of every minute task

Goal

- know which work has been completed.
- how much work is still left to do.
- and when it will all be finished

Software Design Documents

Architecture - A document that describes the overall design of the software, including descriptions of all the major pieces and how they interact with each other.

Data Flow Diagram - way of representing a flow of data through a process or a system.

State Transition Diagram - Another formalized diagram that breaks the software into basic states, or conditions and shows the means for moving from one state to the next.

Software Design Documents

Flowchart - A flowchart is a picture of the separate steps of a process in sequential order

Test plan - The overall method to be used to verify that the software meets the product specification and the customer’s needs

Test cases - The specific items that will be tested and describe the detailed steps that will be followed to verify the software.

Bug reports - The problems found as the test cases are followed. These could be done on paper but are often tracked in a database

Metrics, statistics, and summaries - convey the progress being made as the test work progresses. They take the form of graphs, charts, and written reports

Software Project Staff

PROJECT SPONSOR - the person or group that provides direction and resources, including financial resources for the software project.

Business Analyst - responsible for leading the business analysis process, preparing requirements documentation, and managing change.

SUBJECT MATTER EXPERTS - review requirements, and may have a role in validating and approving requirements documentation.

PRODUCT OWNER - a person who represents the business or end users and is responsible for working with the user group to determine what features will be in the product release.

Software Project Staff

Project managers - responsible for knowing the “who, what, where, when and why” of the software project. This means knowing the stakeholders of the project and being able to effectively communicate with each of them.

TECHNICAL LEAD - translates the business requirements into a technical solution.

Programmers, developers, or coders - Design, write, and fix bugs in the software

Testers or QA - Responsible for finding and reporting problems in the software product

Software Development Life Cycle (SDLC)

The Software Development Lifecycle is a systematic process for building software that ensures the quality and correctness of the software built.

SDLC process aims to produce high-quality software which meets customer expectations. The software development should be complete in the pre-defined time frame and cost.

Why SDLC?

Here, are prime reasons why SDLC is important for developing a software system.

1. It offers a basis for project planning, scheduling, and estimating.
2. Provides a framework for a standard set of activities and deliverables.
3. It is a mechanism for project tracking and control.
4. Increases visibility of project planning to all involved stakeholders of the development process.
5. Increased and enhance development speed.
6. Improved client relations.
7. Helps you to decrease project risk and project management plan overhead

SDLC Phases?

- Phase 1: Requirement collection and analysis
- Phase 2: Design
- Phase 3: Coding
- Phase 4: Testing
- Phase 5: Deployment
- Phase 6: Maintenance

01. Requirement Gathering/Analysis.

This is a process with much communication taking place between stakeholders, end users and the project team.

Meetings with managers, stake holders and users are held in order to determine the requirements like; who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirement gathering phase.

The QA engineer playing the role to configure the requirements using requirements traceability matrix (RTM).

02. Design

In this phase the software design is prepared from the requirement specifications which were studied in the first phase.

System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

In this phase the QA Engineers comes up with the Test strategy, where they mention what to test, how to test.

03. Implementation / Coding:

Upon receiving system design documents, the work is divided in modules/units and actual coding is started.

Since, in this phase the code is produced so it is the main focus for the developer.

This is the longest phase of SDLC.

In this phase the QA Engineers comes up with the Test Environment setup and test Case Documentation.

04. Testing

After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.

During this phase all types of like unit testing, integration testing, Smoke Testing, functional testing, Sanity Testing, system testing, acceptance testing is done as well as non-functional testing are also done.

05. Deployment

After successful testing the product is delivered / deployed to the customer for their use. As soon as the product is given to the customers, they will first do the beta testing/User Acceptance Testing.

If any changes are required or if any bugs are caught, then they will report it to the engineering team. Once those changes are made or the bugs are fixed then the final deployment will happen.

06. Maintenance

Ensure the ongoing functionality and improvement of the software.

Address and fix bugs reported by users.

THANK YOU