01 Introduction to Software Testing

PowerPoint by Wanida Khamprapai

Why does software have defects?

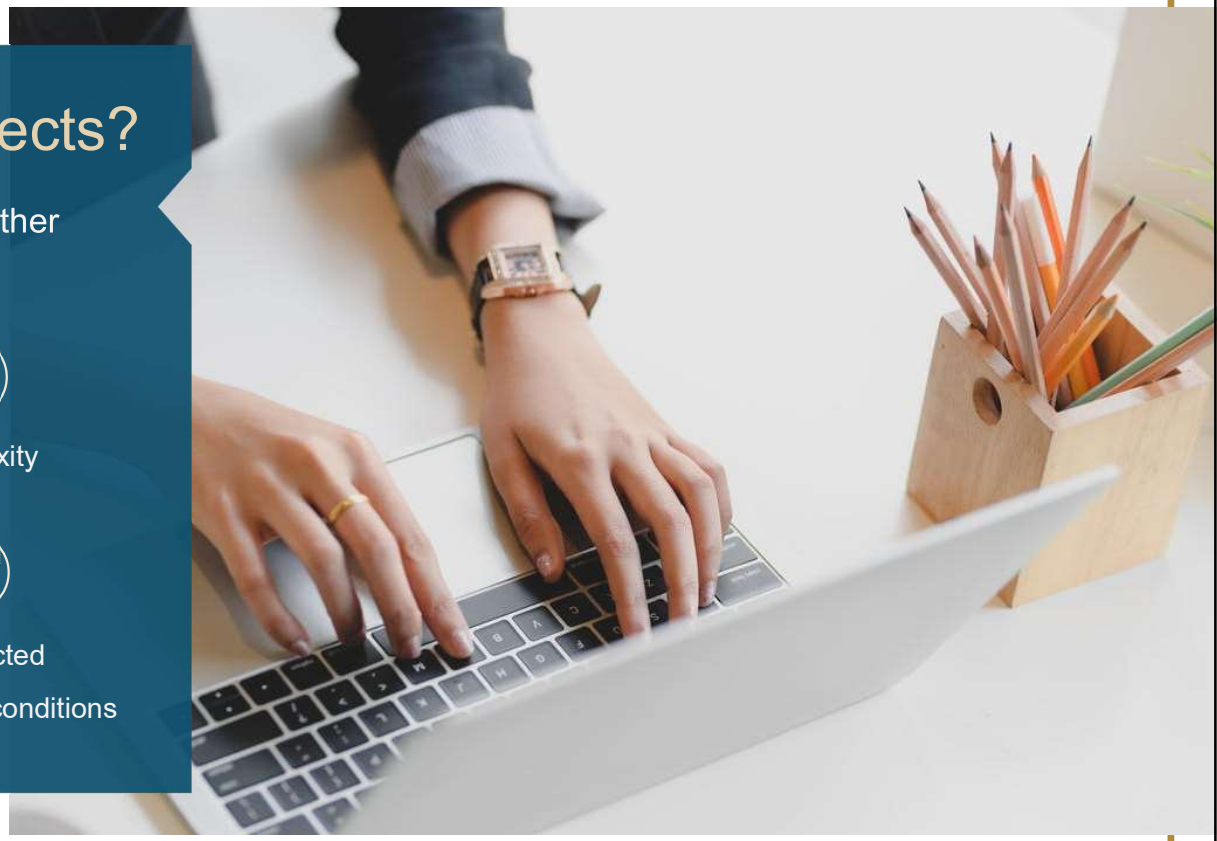
A survey conducted a few years back for the success rate of various projects revealed that

- Only 34% of the projects were successful
- 51% of projects were completed but they were challenged for some of their functionalities
- 15% of projects were completely failed
- The cost of failed projects was as high as $38 Billion
- The survey also revealed that 52% of required functions do not make it to the final product

Why does software have defects?

Every software developed will have some or the other issues due to:

- Time pressure
- Low Capabilities
- Complexity
- Wrong understanding of requirements
- Communication gaps
- Unexpected environment conditions



Impact of defects

A defective baby-food formula

In November 2003 a defective baby-food formula resulted in the deaths of three babies in Israel. Due to a wrong formula in the dispensing system, the food contained less vitamin B1 than required by babies as shown on the label.

Medical device software failures

3140 medical devices were recalled between 1992 and 1998 in the US because 242 of them (7.7%) were attributable to software failures. 79% of them were caused by software defects introduced during changes to the software after its initial production and distribution.

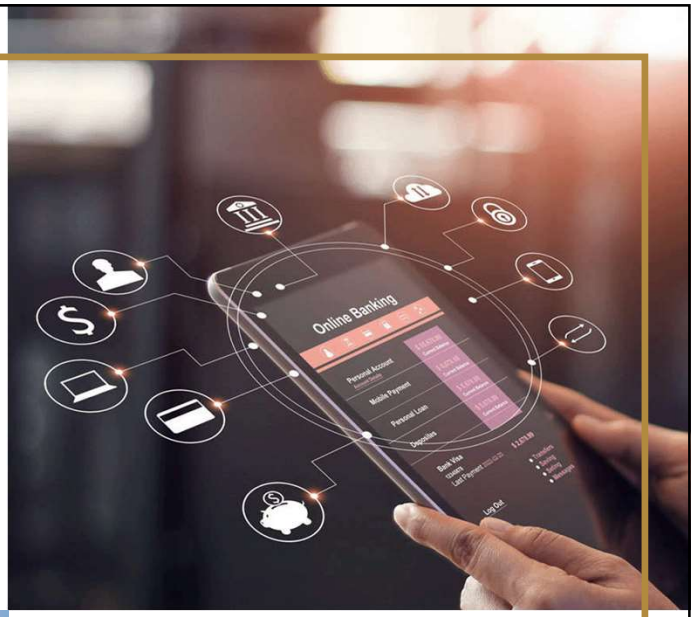
Incorrect fetal weight estimation

Between January 4th and May 24th, 2000, 158 women had been told that they need not worry about having a child with Down Syndrome, even though four of them were carrying fetuses with the abnormality. An investigation into the incident revealed that the software automatically assumed that patients weighed zero pounds if an actual weight was not known.

Impact of defects

The Explosion of the Ariane 5

The first flight of the crewless Ariane 5 rocket, ended after 39 seconds in an unholy ball of smoke and fire. The cause of the failure was integer overflow, essentially meaning that numbers are too big to be stored in a computer system, and sometimes this can cause malfunction.



Software Error at Knight Capital

In 2012 US-based Knight Capital Group Inc, mistakenly sent out more than four million stock orders in less than an hour, costing half a billion dollars to reverse. The reason - The code change was not deployed to all the servers.

MobiKwik glitch

October 2017: Digital Payments firm MobiKwik lost over Rs 19 crore due to a technical glitch.

Software quality

Software quality (IEEE std. 730-2014) is The degree to which a software product meets established requirements; however, quality depends upon the degree to which established requirements accurately represent stakeholder needs, wants, and expectations.

Meeting the requirements

A high-quality software is expected to meet all written development requirement

Customer/stakeholder satisfaction

Quality is also achieved through the fulfillment of stakeholder needs and wants.

Cost of Quality

Maintaining high quality requires time and effort and hence cost. Although testing is itself an expensive activity, the cost of not testing is potentially much higher. **Cost of Quality (COQ)** is a means to quantify the total cost of quality-related efforts and deficiencies. While calculating the cost of quality, we need to consider the following three types of costs:

Failure Cost

These costs are incurred only because there was a defect.

Appraisal Cost

It involves cost of reviews (of all the work product), testing, and debugging.

Prevention Cost

It involves training, process implementation, planning, developing reusable components etc.

Cost of Quality

Failure Cost: Any effort or cost involved in rework, replacement, prestige business loss. These costs are incurred only because there was a defect. So, it is also known as the **cost of poor quality**.

Internal failure costs

incurred when the defects are detected before the application is moved to the production environment. It includes direct cost

- To reproduce, analyze and fix the problem
- Retesting to determine that the defect is removed
- Reimplementation of the corrected code
- Costs involved in mitigating the risk of any possible side effects due to rework done for fixing the problem

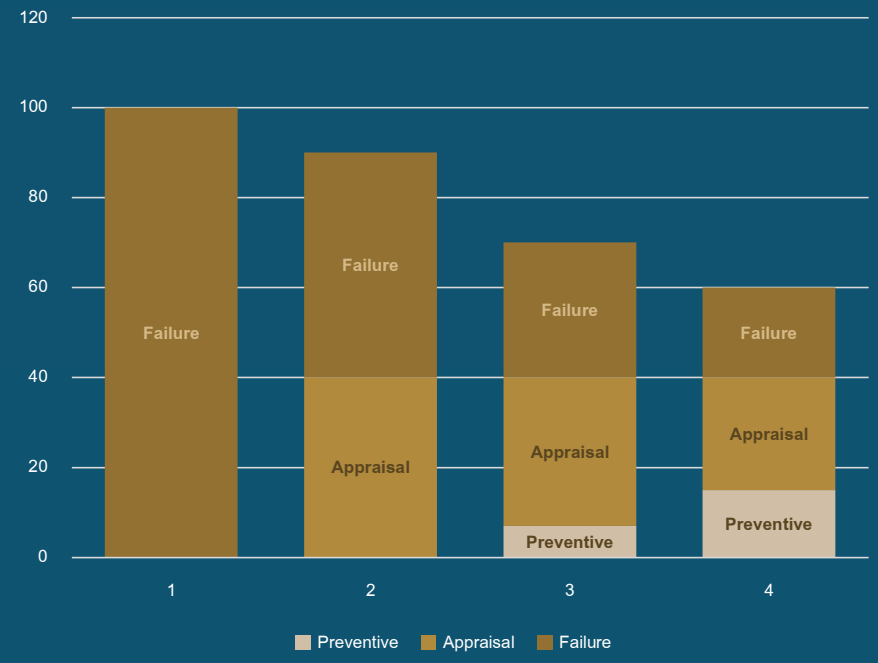
External failure costs

associated with the defects found after the product has been shipped to the customer.

- Analysis of issue / complaint resolution
- Product return and replacement
- Helpline support
- Labor costs associated with warranty work
- Indirect costs such as unsatisfied customers, loss of reputation, and loss of business

Cost of Quality

Total Cost of Quality = Failure cost + Appraisal cost + Preventive cost



In the indicative bar chart depicted below, you can find four different stack bars indicating the total cost of quality with different levels of effort in prevention and appraisal activities.

It has been observed that if you increase the time and effort for an appraisal (testing and reviews), failure cost will reduce resulting in an overall reduction in the total cost of quality. It has been statistically proven and experienced that testing the application at the right time reduces the overall cost.

Software testing is no different than testing any other product. All manufacturing companies would test their product with respect to functional and non-functional requirements before launching their product in the market. Not only the final product, but all individual components of the product are first tested before they are assembled together.

Software testing

Software testing is defined as a process of evaluation that either the specific system meets its originally specified requirements or not. Software Testing refers to finding bugs, errors or missing requirements in the developed system or software. So, this is an investigation that provides the stakeholders with the exact knowledge about the quality of the product.

Principles of software testing

Testing shows presence of defects

Testing is done to find as many defects as possible. Testing is not done to prove that the software is bug-free

Exhaustive Testing is impossible

Testing all combinations of input data values and scenarios are impossible. Prioritize important tests based on risk.

Early Testing

Software testing starts from requirement gathering. Testing should be done parallelly along with other product development phases.

Defect Clustering

It's more like 80-20 Pareto principle that is 20% of the defects cause 80% of the problem.

Testing is context dependent

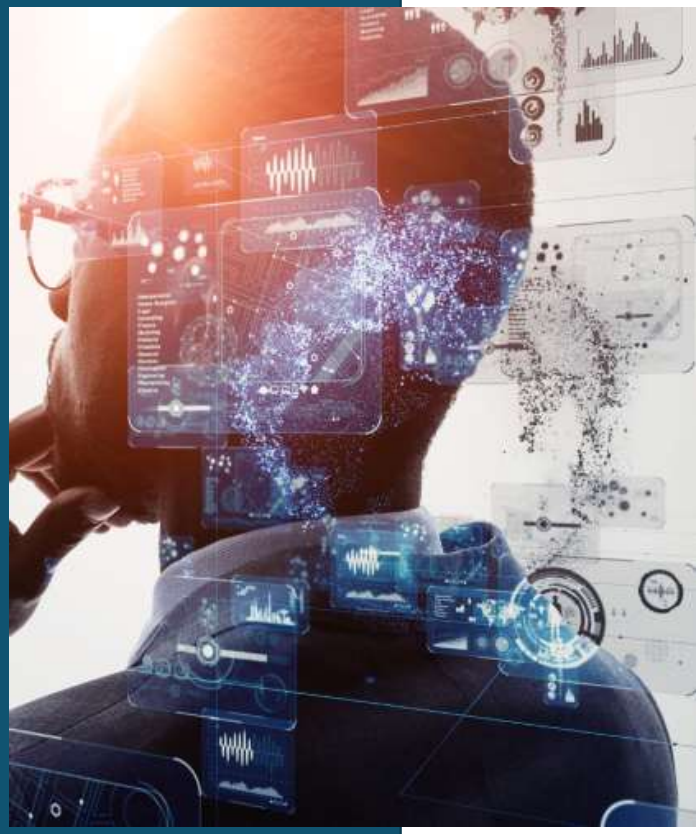
Software should be tested based on the purpose of the product that has been built. Different software is tested differently.

Pesticide Paradox

Executing same test cases, again and again, will not help us to identify new defects. Different and new test cases should be added to find more defects in the software

Absence-of-errors fallacy

It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement.



Test objectives

The typical test objectives are:

- Evaluating work products such as requirements, user stories, designs, and code
- Triggering failures and finding defects
- Ensuring required coverage of a test object
- Reducing the level of risk of inadequate software quality
- Verifying whether specified requirements have been fulfilled
- Verifying that a test object complies with contractual, legal, and regulatory requirements
- Providing information to stakeholders to allow them to make informed decisions
- Building confidence in the quality of the test object
- Validating whether the test object is complete and works as expected by the stakeholders

Software tester responsibilities

- Reviewing software requirements and preparing test scenarios.
- Executing tests on software usability.

- Analyzing test results on database impacts, errors or bugs, and usability.
- Interacting with clients to understand product requirements.

- Preparing reports on all aspects related to the software testing carried out and reporting to the design team.

- Participating in design reviews and providing input on requirements, product design, and potential problems.

Career path for software tester

Trainee/Junior Software Tester

You'll review requirements and specifications to define test conditions, design and build test cases and interpret and execute moderately complex test scripts.

0-2 years

Software Tester

4-5 years

You'll create test cases using in-depth technical analysis of both functional and non-functional specifications. You'll specify requirements for environment, data, resources and tools and design and produce re-usable test scripts.

Senior Software Tester

6-8 years

You'll coordinate and manage planning of testing. You'll define and communicate the test strategy for projects and contribute to your organization's testing standards and best practices.

Lead Software Tester

8-10 years

You'll take responsibility for the management of all testing activity within complex development or integration projects and programs. You'll give direction and leadership to other testers and define strategies to ensure effective and efficient testing.

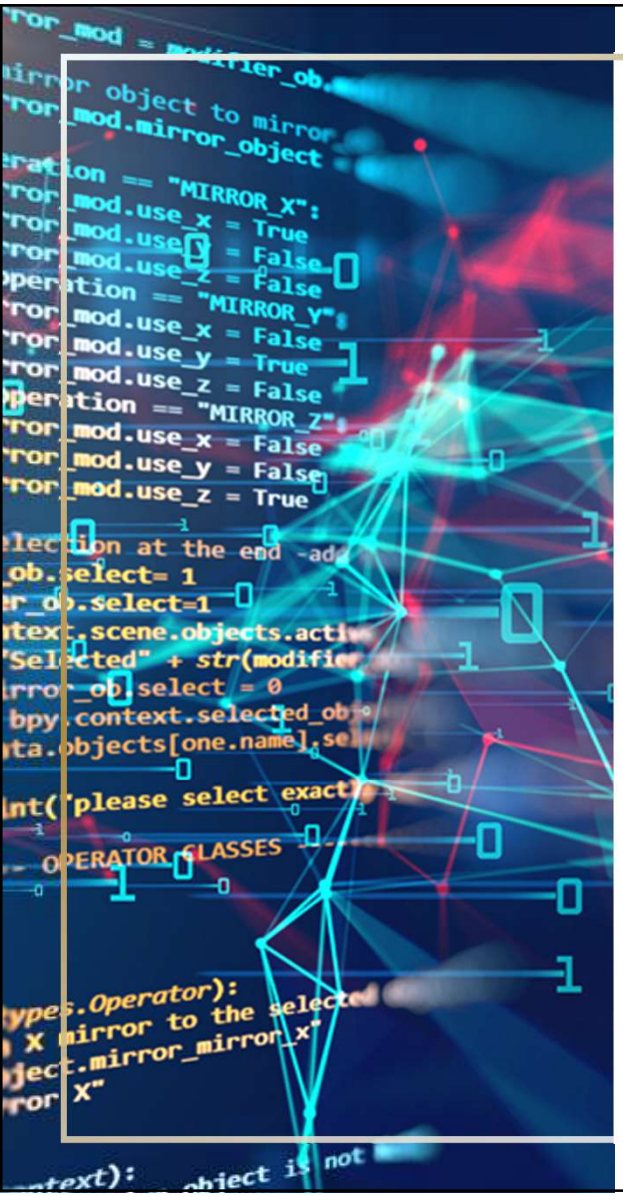
Software tester skills

Technical skills

- Programming skills
- Front-end skills
- Database skills
- API skills
- Testing tool skills
- Version control system
- UNIX/Bash commands

Soft skills

- Communication
- Thinking as end users
- Knowledge sharing
- Flexible
- Team player
- Learn new technologies
- Problem solving
- Time management



How to be a good software tester?

- Always ask frequent questions
- Build healthy relations with developers
- Switching from manual testing to automation testing can be a great career move.
- Double-check the environment (keep it safe for the user)
- Don’t rush for deadlines (like UT/permission, etc.)
- Keep ‘testing’ notes for yourself – or prepare a chart for future references
- Enroll yourself in ISTQB (International Software Testing Qualifications Board) and focus on building strong skills.

Questions & Answers