SQA

Lec 01

Testing terminology:New standard BS 7925-1 adopted by the ISEB / ISTQB

A person makes an error, that creates a fault in the software,that can cause a failure in operation
Failure is an event; fault is a state of the software, caused by an error
https://www.tutorialspoint.com/software_testing_dictionary/fault.htm

Reliability:

- the probability
- software will not cause the failure
- for a specified time
- under specified conditions

Fault-free X
“fault-free” software application always reliable X

Why do faults occur

written by human

deliver to strict deadlines
Less experience

What do software faults cost

- huge sums:

  Medicine infusion pumps recalled for deadly flaw

  Software glitch in F-35 fighter planes causes target detection problems

  Software bug assists in bank heist

  The Equifax social security hack

  Hawaii Sends Out a State-Wide False Alarm About a Missile Strike

- very little or nothing at all
- software is not “linear”
- Safety-critical systems are critical to manage and high cost to develop and maintain.

why is testing necessary

because software is likely to have faults

to learn about the reliability of the software

because failures can be very expensive

to avoid being sued by customers

to stay in business

What is an Error?

When the system produces an outcome, which is not the expected one or a consequence of a particular action, operation, or course, is known as error.

Error or mistake leads to a defect and usually raises due to various reasons. It may be system specification issue or design issue or coding issue, which leads to a defect. Error leads to defects and if the defect uncovered by QA leads to Failure.

What is a Fault?

Software fault is also known as defect, arises when the expected result don't match with the actual results. It can also be error, flaw, failure, or fault in a computer program. Most bugs arise from mistakes and errors made by developers, architects.

Fault Types

Following are the fault types associated with any:

- Business Logic Faults
- Functional and Logical Faults
- Faulty GUI
- Performance Faults
- Security Faults

Preventing Faults

Following are the methods for preventing programmers from introducing Faulty code during development:

- Programming Techniques adopted
- Software Development methodologies
- Peer Review
- Code Analysis

What is a Failure?

Under certain circumstances, the product may produce wrong results. It is defined as the deviation of the delivered service from compliance with the specification.

Not all the defects result in failure as defects in dead code do not cause failure.

Lec 02

We don’t do total testing to our software because of Time,Cost and Resources Limitation constraints.

What is Exhaustive Testing?

Exhaustive testing is a test approach in which all possible data combinations are used for testing. Exploratory testing includes implicit data combinations present in the state of the software/data at the start of testing.

How much testing is enough?

-when you are confident that the system works correctly

If you can guarantee system will run properly at that moment you can stop the testing.

-it depends on the risks for your system

The context of the system is more important if it is a critical system or normal system. Usually normal system probability of defect can be 10%-25% but in critical system alike medical systems, aerospace systems, military systems and nuclear plant system has very low defects rates.

Depends on RISK

important faults

failure costs

releasing untested or under-tested software

credibility and market share

missing a market window

over-testing, ineffective testing

Use RISK to allocate the time available for testing by prioritizing testing.

RISK to determine
what to test first
what to test most
how thoroughly to test each item
what not to test (this time)

what does testing test?

system function, correctness of operation

non-functional qualities: reliability, usability, maintainability, re-usability, test ability, etc.

Lec 05

Testing

The execution of a program to find its faults

Verification

The process of proving the programs correctness.

Validation

The process of finding errors by executing the program in a real environment

Debugging

Diagnosing the error and correct it

Unit Testing (White Box):Individual components are tested. By developer.

https://www.tutorialspoint.com/software_testing_dictionary/unit_testing.htm

Integration Testing:

https://www.tutorialspoint.com/software_testing_dictionary/system_Integration_testing.htm

Function Testing (Black Box):Designed to exercise the to its external specifications

https://www.tutorialspoint.com/software_testing_dictionary/functional_testing.htm

Regression Testing:Test the effects of the newly introduced changes on all the previously integrated code.

https://www.tutorialspoint.com/software_testing_dictionary/regression_testing.htm

System Test:

https://www.tutorialspoint.com/software_testing_dictionary/system_testing.htm

Acceptance and Installation Tests:

https://www.tutorialspoint.com/software_testing_dictionary/acceptance_testing.htm

---

Test Planning

Define the functions, roles and methods

start during the requirements phase

Testing should be treated like an experiment.

Real-Time testing is necessary because the deployment system is usually more complicate than development system

---

Lec Test Estimation

Test Estimation
Estimate is a forecast or prediction. What it would Cost.
How long a Task would take

Estimate is based on

Past Experience/ Past Data
Available Documents/ Knowledge
Assumptions

Calculated Risks

Test Estimation Process

Step 1 - Define Function Point
Step 2 - Give Weight age to all Function Points

Step 3 - Define an Estimate Per Function Point

Step 4 - Calculate Total Effort Estimate