


DA-IICT

IT 314: Software Engineering

*Psychology of Testing
&
Testing Principles*

Saurabh Tiwari

1



Software Testing is a process, or a series of processes, designed to make sure computer code does what it is designed to do and that it does not do anything unintended.

- Suppose you have a 15 input fields each one having 5 possible values.
- How many combinations to be tested?

$$5^{15} = 30517578125!!!$$

Education means considerably more than just teaching a student to read, write, and manipulate numbers. Computers, the Internet, and advanced electronic devices are becoming essential in everyday life and have changed the way information is gathered. How this new technology is utilized in the curriculum and managed by teachers will have an important role to play in widening the resource and knowledge base for all students. Technology affects the way teachers teach and students learn. To make the best use of information technology (IT), schools need a workable plan to fully integrate it into all aspects of the curriculum so students are taught how, why, and when to use technology to further enhance their learning. If a school does not have a clear plan of how and why it wishes to implement IT, then it runs the risk of wasting money. In schools today, nearly all classrooms have access to a computer. However, many schools mistake this as incorporating information technology into the curriculum. School staff needs to research what IT is available and what would best serve the school's purpose, not simply purchase the latest equipment. There should be a policy stating how IT is going to assist pupils' development and what teachers want pupils to achieve. Staff members need to be clear about what they want IT to do for them before they can start incorporating it into their lessons.



Psychology of Testing



The Mind of a TESTER

Four different kinds of thinking exhibited by a GOOD TESTER
[Kaner, Bach, Pettichord]

1. **Technical Thinking:** *the ability to model technology and understand causes and effects*
 2. **Creative Thinking:** *the ability to generate ideas and see possibilities*
 3. **Critical Thinking:** *the ability to evaluate ideas and make inferences*
 4. **Practical Thinking:** *the ability to put ideas into practice*
-



Example to TEST YOUR Mind

An example of these kinds of thinking is found in a fable called “The King’s Challenge”

The King’s Challenge (a fable)

Once upon a time, a mighty king wanted to determine which of his three court wizards was the most powerful.

So,

He put the three court wizards in the castle dungeon and declared whoever escaped from his respective dungeon first was the most powerful wizard in all the kingdom.

(If you are one of three wizard, what you would do.)



Example to TEST YOUR Mind...

The **first wizard** immediately started chanting musical poems to open his cell door.

The **second wizard** immediately started casting small polished stones and bits of bone on the floor to learn how he might open his cell door

The **third wizard** sat down across from his cell door and thought about the situation for a minute. Then he got up, walked over to the cell door and pulled on the door handle. The cell door swung open because it was closed but not locked.

Thus, the third wizard escaped his cell first and became known as the most powerful wizard in all the kingdom.

Example to TEST YOUR Mind...

What kind of “tester” thinking did the third wizard exercise in solving the king’s puzzle?

- Creative thinking: *the ability to see the possibility that the door was not locked in the first place.*
 - Practical thinking: *the ability to decide to try the simplest solution first.*
-


Another Example to TEST YOUR Mind...

BUYING A CAR

- Go for a test drive & What you are supposed to do?
 - Take test drive to *BREAK THE CAR*
 - OR
 - To improve the CAR’s *DESIGN*

Objectives of TEST DRIVE are:

- To validate affordability
 - To validate attractiveness
 - To validate comfort
 - To validate usefulness
 - To validate performance
-



When you have clear testing objectives, then we choose approaches that best validate the car against those Objectives.

Testing Approaches include:

- *Examine the sticker price and sale contract*
- *Trying out the radio, the air conditioner and the lights*
- *Trying acceleration, stopping and cornering*

These testing approaches are referred to by fairly common terminology in the testing industry.

How??



Examine = Static Testing

(observe, read, review without actually driving the car)

Try out = Functional and structural testing

(work different features of the car without actually driving the car)

Try = Performance Testing

(work different features of the car by actually driving the car)

The Psychology of Testing

One of the primary causes of POOR program testing is - most of the programmers begin with a FALSE definition of the term:

Such as,

- *“Testing is the process of demonstrating that error are not present”*
- *“The purpose of testing is to show that a program performs its intended functions correctly”*
- *“Testing is the process of establishing confidence that a program does what it is supposed to do”*

Testing is the process of executing a program with the intent of finding errors.

Testing Terms

- **Test case** - A set of Inputs, execution preconditions, and expected outcomes for testing an specific aspect of CUT
 - **Test Suite** - A collection of test cases for the CUT
 - **Test Criterion** - A set of test requirements
 - **Effectiveness** - Fault detection capability
 - **Efficiency** - The average testing cost (i.e. effort) to identify a fault in the program
-

General Testing Principles

The Seven Key Principles

1. Testing shows the presence of Defects

(it is not a ghost. It is a fact)

- We test to find Faults (as known as Defects)
- As we find more defects, the probability of undiscovered defects remaining in a system reduces (decreasing nature).
- However Testing **cannot prove that there are no defects present**

Why Testing is necessary



Testing Pearls of Wisdom

“The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that program”

- *“Do not plan a test effort under the tacit assumption that no errors will be found”*
- *“A good test is one that has a high probability of detecting an as yet undiscovered error”*
- *“A successful test is one that detects an as-yet undiscovered error”*

To test a program is to try to make it fail

General Testing Principles

The Seven Key Principles

2. Exhaustive Testing is Impossible!

- We have learned that we cannot test **everything** (i.e. all combinations of inputs and pre-conditions).
- That is we must Prioritise our testing effort using a Risk Based Approach.

Why Testing is necessary

Why don't we test everything ?

System has 20 screens
 Average 4 menus / screen
 Average 3 options / menu
 Average of 10 fields / screen
 2 types of input per field
 Around 100 possible values

Approximate total for exhaustive testing
 $20 \times 4 \times 3 \times 10 \times 2 \times 100 = 480,000$ tests

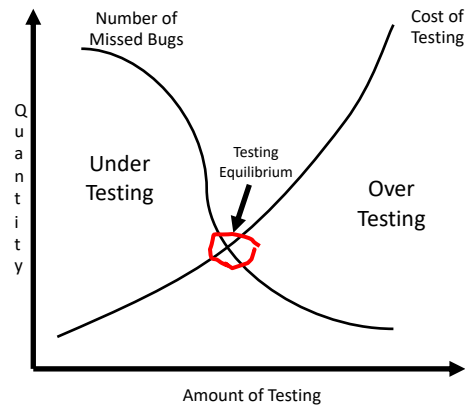
Test length = 1 sec then test duration = 17.7 days
 Test length = 10 sec then test duration = 34 weeks
 Test length = 1 min then test duration = 4 years
 Test length = 10 mins then test duration = 40 years!



It is not a matter of time. But, time is money (salary is taken by hour. So second is valuable for software houses)

Urgency of Equilibrium

- If you test too little, the probability of software failure increases
- If you try to test too much, the development cost becomes unaffordable
- So, we need to conduct some sort of equilibrium



General Testing Principles

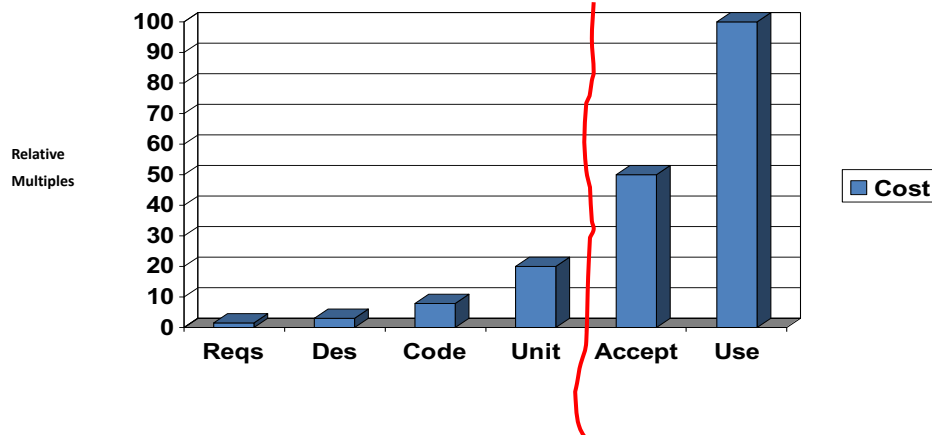
The Seven Key Principles

3. Early testing

- Testing activities should start as early as possible in the development life cycle
- These activities should be focused on defined objectives - outlined in the Test Strategy
- Remember from our Definition of Testing, that Testing doesn't start once the code has been written!



Cost of Fault Correction




General Testing Principles

The Seven Key Principles


4. Defect Clustering

- Defects are not **evenly distributed** in a system
- They are **'scattered'** and may get **'clustered'**
- In other words, most defects found during testing are usually confined to a small number of modules (**80% of uncovered errors focused in 20% modules of the whole application**)
- Similarly, most operational failures of a system are usually confined to a small number of modules
- An important consideration in test prioritisation!






General Testing Principles




The Seven Key Principles




5. The Pesticide Paradox

- Testing identifies bugs, and programmers respond to fix them
- As bugs are eliminated by the programmers, the software improves
- As software improves the effectiveness of previous tests erodes
- Therefore we must learn, create and use new tests based on new techniques to catch new bugs (i.e. It is not a matter of repetition. It is a matter of learning and improving)



General Testing Principles



The Seven Key Principles

6. Testing is Context (background) Dependent

- Testing is done differently in different contexts
- For example, safety-critical software is tested differently from an e-commerce site
- Whilst, Testing can be 50% of development costs, in NASA's Apollo program (it was 80% testing)
- 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software
- 1 to 3 failures per KLOC typical for industrial software
- 0.01 failures per KLOC for NASA Shuttle code!
- Also different industries impose different testing standards

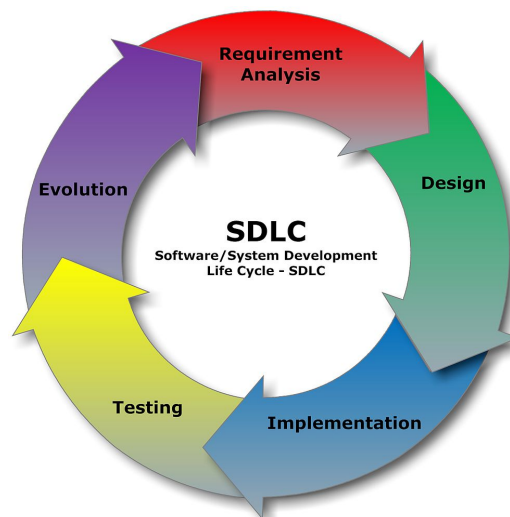
General Testing Principles

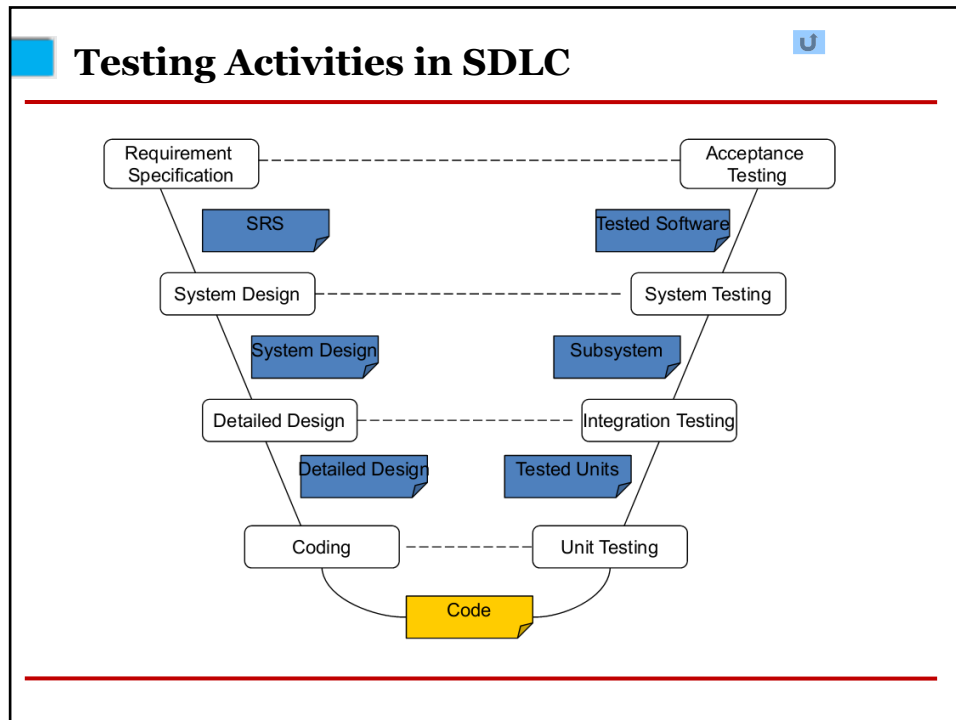
The Seven Key Principles

7. Absence of Errors Fallacy

- If we build a system and, in doing so, find and fix defects
It doesn't make it a good system
- Even after defects have been resolved, it may still be unusable
and/or does not fulfil the users' needs and expectations

SDLC: Where Testing FITS?





Static and Dynamic Testing

- **Static Testing** : Testing without executing the program
 - This include software inspections and some forms of analyses
 - Very effective at finding certain kinds of problems - especially “potential” faults, that is, problems that could lead to faults when the program is modified
- **Dynamic Testing** : Testing by executing the program with real inputs

Static Testing

Static testing is the process of carefully and methodically reviewing the software design, architecture, or code for bugs without executing it.



29

Desk Checking

- The process of a programmer reading his or her own program before testing it

Walkthrough

- a group of developers (#3-4) - performs the reviews
- Only one participant is the author of the program

Inspections

- Code Inspection is the most formal type of review
- Generally, done using the Checklist (or questionnaire based approach)

30

- Systematic use of human capability to think and analyze
 - Apply human analytical capability to check and evaluate complex issues
 - This done by through intensive reading and trying to understand the documents that are examined
 - Techniques : Reviews
 - Another terms : inspection
 - Peer reviews : if colleagues provide feedback
 - Types of reviews
 - Walkthrough
 - Inspection
 - Technical review
 - Informal review

31

Requirement Stage

- Test requirements documents by disciplined inspection and review
- Developing scenarios of expected system use (aka use cases)
- Preparation of test plan which should include:
 - I. specification
 - II. description of test procedures
 - III. test milestones
 - IV. test schedule
 - V. test data reduction
 - VI. evaluation criteria

32

