QA Fundamentals

greenwich.edu.vn

Table of Contents

BUGS!

Software Defects

- Human being can make an **error (mistake)**
- Errors produce **defects**
  - Defects are **faults / bugs** in the program code, or in a document
- If a defect in code is executed, that might cause a **failure**:
  - Fail to do what it should do
  - Do something it shouldn't

Software Defects

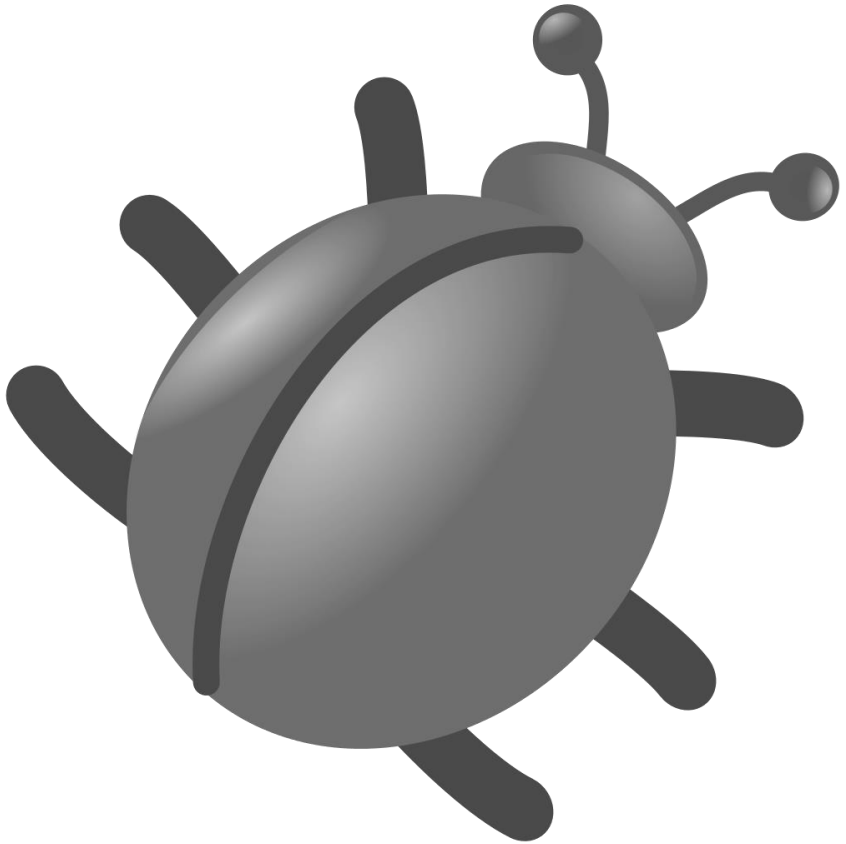
- The human factor
  - Humans make **mistakes**
  - Poor **training**
  - **Time** pressure
  - **Code** complexity
  - Complexity of **infrastructure**
  - Changing **technologies**

Software Defects

- Organizational factors
  - Inefficient **communication**
  - Unclearly defined **requirements**
- Environmental conditions
  - Radiation, Magnetism, Electronic fields, Pollution, Etc.
  - These can change the hardware conditions

Software Defects

- Anomaly
- Error
- Bug
  - Defect
  - Fault
  - Problem
- Failure
- Defect/fault masking



UNIVERSITY of GREENWICH
Alliance with FPT Education

Bug Fixing Importance

UNIVERSITY of GREENWICH
Alliance with FPT Education

Bug Fixing importance (2)

Bug Fixing Importance (3)

- Unfixed bugs camouflage other bugs
- Suggest quality isn’t important
- Duplicate effort
- Unreliable metrics and money loss
- Fixing a bug today costs less than tomorrow

Bug Fixing Importance

What Is Testing?

What Is Testing?

- The process of exercising software
  - To verify that it satisfies specified requirements and to detect errors
- The process of analyzing a software item
  - To detect the differences between existing and required conditions (that is, bugs)
  - To evaluate the features of the software item

What Is Testing?

- The process of operating a system or component under specified conditions
  - Observing or recording the results
  - Making an evaluation of some aspect of the system or component

Main Test Activities

- Testing is not just running tests, but also:
  - Planning and control
  - Choosing test conditions
  - Designing and executing test cases
  - Checking results
  - Evaluating exit criteria
  - Reporting on the testing process and system under test
  - Finalizing or completing closure activities

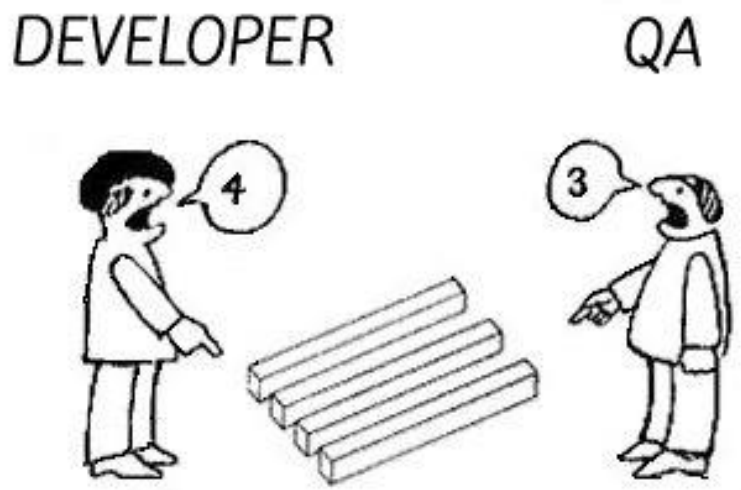
Main Test Objectives

- Testing pursues several **objectives**:
  - Finding defects
  - Gaining confidence about the level of quality
  - Providing information for decision-making
  - Preventing defects

QA vs Devs

QA testers

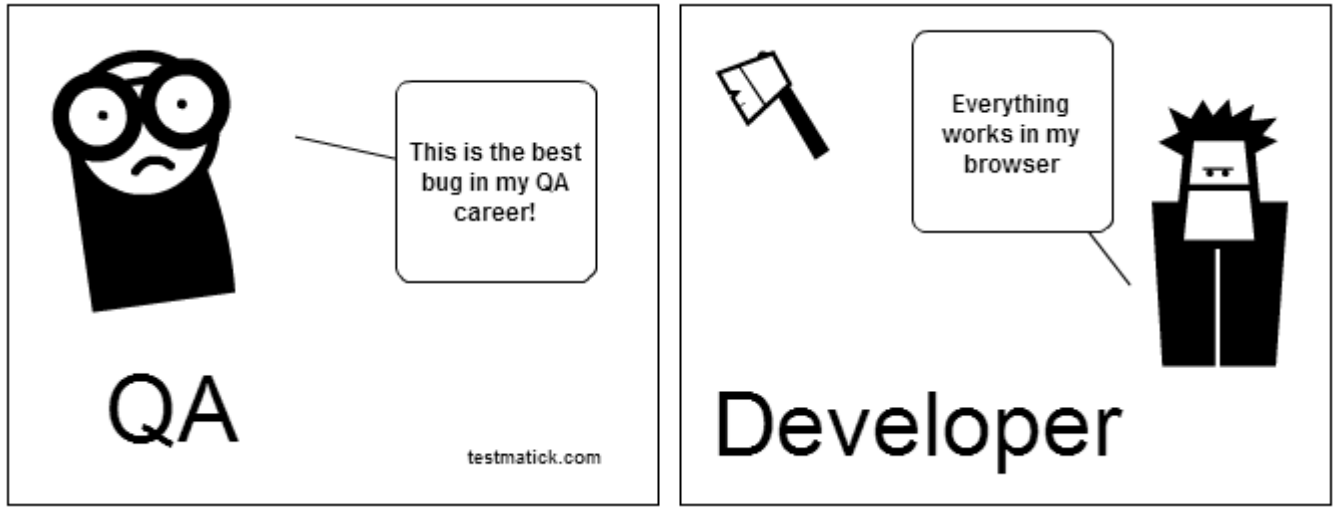
- Are **perceived as destructive** – only happy when they are finding faults!
- Usually require **good communication skills, tact & diplomacy.**
- Normally need to be **multi-talented** (technical, testing, team skills).



QA vs Devs

Developers

- Are **perceived as** very **creative** – they write code without which there would be no system
- Are **rarely good communicators**
- Can often **specialize** in just **one or two skills** (e.g. VB ,C++,JAVA,SQL)

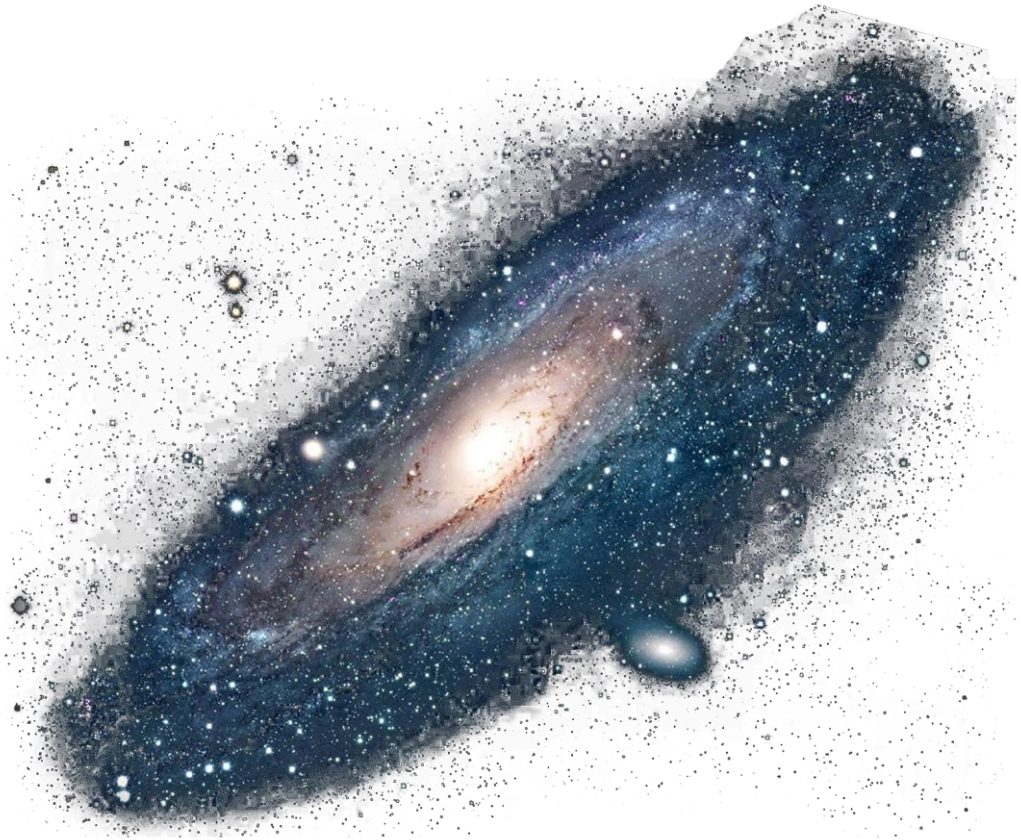


Seven Testing Principles

- Testing shows presence of defects
  - Testing can show that defects are present
  - Cannot prove that there are no defects
  - Appropriate testing reduces the probability for defects

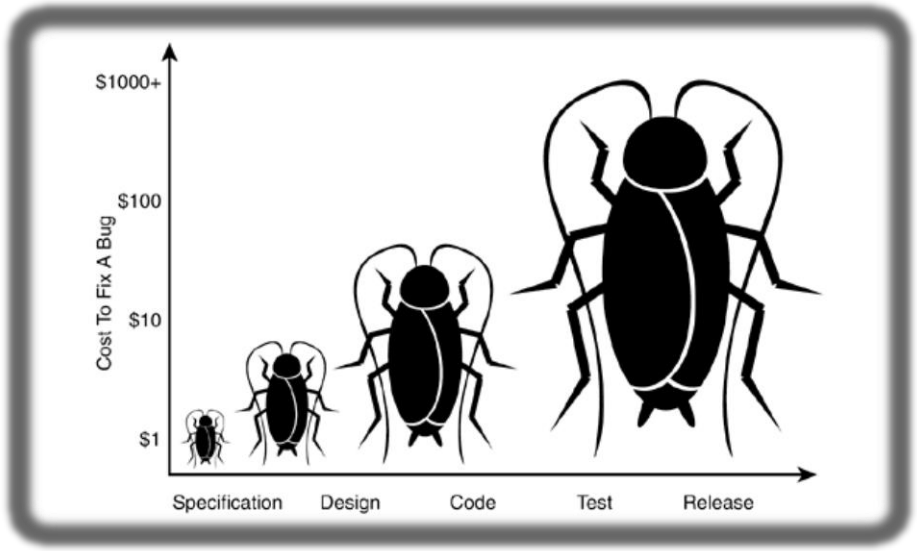
Seven Testing Principles (2)

- Exhaustive testing is impossible
  - All combinations of inputs and preconditions are usually almost infinite number
  - Testing everything is not feasible
  - Risk analysis and priorities should be used to focus testing efforts



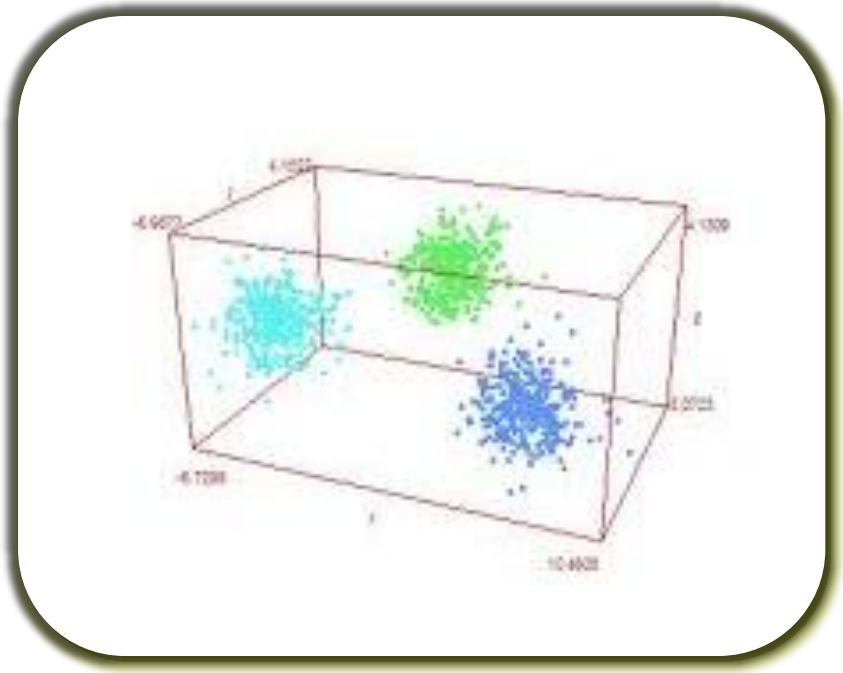
Seven Testing Principles (3)

- Early testing
  - Testing activities shall be started as early as possible
    - And shall be focused on defined objectives
  - The later a bug is found – the more it costs!



Seven Testing Principles (4)

- Defect clustering
  - Testing effort shall be focused proportionally
    - To the expected and later observed defect density of modules
  - A small number of modules usually contains most of the defects discovered

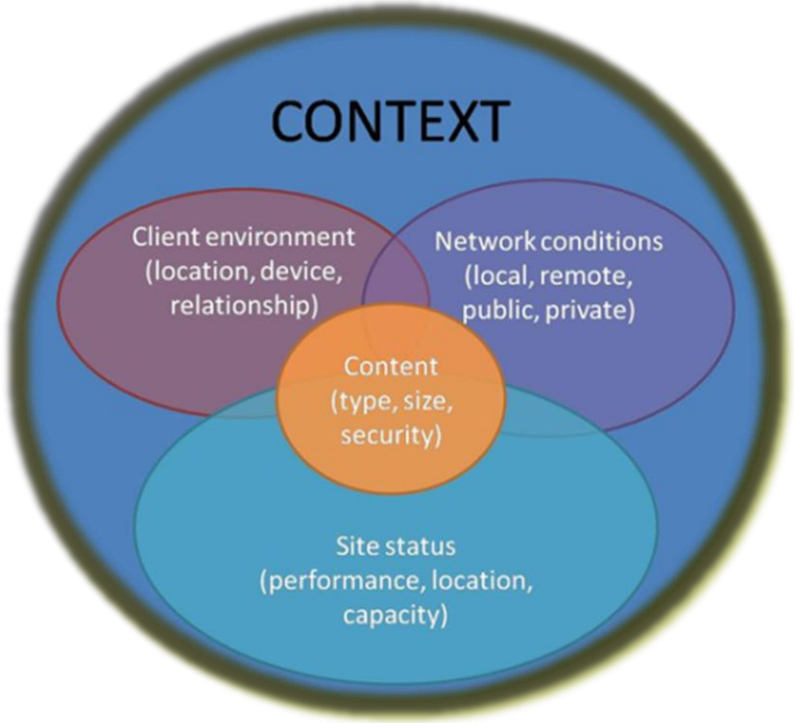


Seven Testing Principles (5)

- Pesticide paradox
  - Same tests repeated over and over again tend to loose their effectiveness
    - Previously undetected defects remain undiscovered
  - New and modified test cases should be developed

Seven Testing Principles (6)

- Testing is context dependent
  - Testing is done differently in different contexts
  - Safety-critical software is tested differently from an e-commerce site



Seven Testing Principles (7)

- Absence-of-errors fallacy
  - Finding and fixing defects itself does not help in these cases:
  - The system built is unusable
  - Does not fulfill the users needs and expectations

TEST CASES AND TEST LOGS

Why test?

- Make sure the system performs every task that it was intended to, **correctly and completely**.
- A list of aims for the system is drawn up at the **Specification of Requirements** stage.
- These aims detail what your system is supposed to do and how.

What to check?

- Does it meet each aim **accurately**
- Does it **reject erroneous data** that could produce incorrect or silly results
- Does it **displays adequate prompts** to guide the user
- Does it display all output **fully and clearly**
- Do all data capture forms include all the information required in a **clear manner**
- Do all outputs display all information in a **concise and clear way** with suitable headings, etc.

How to do it

- Each test on your test plan should be numbered.
- Must obtain either **output** or **screenshots** that show the results of testing.
- Number each test case.
- Annotate each test case

Test Plan/Test Case

Test	What is being tested	How	Test data used	Expected Results
1	Order of input on data entry screen	Enter data from data capture sheet into the form on the data entry screen	Data set 1	Data entry order is the same as that on the data capture sheet
2	Validation of input	Enter typical values, boundary value, values that should be rejected	Data set 2	Good data accepted, bad data rejected
3	Accurate calculations	…	Data set 3	Data set 3A
4	Scaled output	…	Data set 4	paginated output
5				

Test Log

Test	What is being tested	How	Test data used	Expected Results	Date	Actual results	Action taken
1	Order of input on data entry screen	Enter data from data capture sheet into the form on the data entry screen	Set 1	Data entry order is the same as that on the data capture sheet	1 Oct 2011	OK	None
2	Validation of input	Enter typical values, boundary value, values that should be rejected	Set 2	Good data accepted, bad data rejected	2 Oct 2011	Email not validating correctly	Recode and re-test
3	Accurate calculations	…	Set 3	Data set 3A			
4	Scaled output	…	Set 4	paginated output			
5	Retest validation of input						

The procedure

- Invented some suitable test data for the purpose.
- Run the system to see what actually happens, complete your test log, and correct any errors.
- You may need to do regression testing in case there are unexpected consequences of your changes

In practice

- Ignore simple de-bugging as code is written
- Just make a note of the testing you do, keep a record of the outcomes
- These can be added to your Test Log later.
- Don't throw away any examples of where things went wrong. These are your evidence of testing. Store them safely in date order

Summary

- It still has bugs
- Reasons does not change
- Importance of fixing bugs
  - More Customers
  - More money
  - No unsatisfied developers
- Test cases
- Test logs