Software Quality Assurance & Testing

TOPIC1-LECTURE 1: INTRODUCTION

What is Software Quality?

- **Software Quality (as per ISO/ IEC 9126):**

  The totality of functionality and features of a software product that contribute to its ability to satisfy stated or implied needs.

- Software Quality (as **IEEE** Std 610):

  The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations.

What is Software Quality?

- According to **ISO 9126**, software quality consists of:
  - Functionality
  - Reliability
  - Usability
  - Efficiency
  - Maintainability
  - Portability

What's the main challenges of software development now-a-days?

- High **Cost**
- Difficult to deliver on **Time**
- Low **Quality**

- What is Software **Q**uality **A**ssurance?
- What is Software **T**esting?
- What are the differences between them?

Software Quality Assurance (SQA):

- Defined as a planned and systematic approach to the evaluation of the quality and adherence to software product standards, processes, and procedures.
- An umbrella activity that is applied throughout the software process.
- Consists of a means of monitoring the software engineering processes and methods used to ensure quality.
- An effective approach to produce high quality software.

Software Testing:

- Software Testing is the process of executing a system or component under specified conditions with the intent of finding defects/bugs and to verify that it satisfies specified requirements.
- Main goal ==> To detect bugs
- Have different levels
- **Static** testing vs. **Dynamic** testing
- **Manual** testing vs. **Automated** testing

QA vs. Testing

Software Quality Assurance

- Process-oriented activity
- Oriented to bug prevention

Software Testing

- Product-oriented activity
- Oriented to bug detection

Software Quality Assurance & Testing

TOPIC1-LECTURE 2

- What is the **COST** of a bug?

- Failure Example 01
- Flight Ariane 5

(Most Expensive Computer Bug in History)

On June 4, 1996, the rocket Ariane 5 tore itself apart 37 seconds after launch because of a malfunction in the control software making the fault most expensive computer bug in history.

==> mission critical issue

- Failure Example 02
- Lethal X-Rays :**Therac-25 system**

  Therac-25 was a radiation therapy machine produced by Atomic Energy of Canada Limited (AECL) in 1986. But initially lot of people died because of massive overdose of radiation. And this is happen because of a software bug.

  ==> safety critical issue

What is a computer bug?

- In 1947 Harvard University was operating a room-sized computer called the Mark II.
  - mechanical relays
  - glowing vacuum tubes
  - technicians program the computer by reconfiguring it
  - Technicians had to change the occasional vacuum tube.
- A moth flew into the computer and was zapped by the high voltage when it landed on a relay.
- Hence, the first computer bug!

Bugs a.k.a. …

- Defect
- Fault
- Problem
- Error
- Incident
- Anomaly
- Variance
- Failure
- Inconsistency
- Product Anomaly
- Product Incidence

Defective Software

- We develop programs that contain defects
  - How many? What kind?
- Hard to predict the future, however…
  it is highly likely, that the software we (including you!) will develop in the future will not be significantly better.

Sources of Problems

- **Requirements Definition:** Erroneous, incomplete, inconsistent requirements.
- **Design:** Fundamental design flaws in the software.
- **Implementation:** Mistakes in chip fabrication, wiring, programming faults, malicious code.
- **Support Systems:** Poor programming languages, faulty compilers and debuggers, misleading development tools.

Sources of Problems (Cont'd)

- **Inadequate Testing of Software:** Incomplete testing, poor verification, mistakes in debugging.
- **Evolution:** Sloppy redevelopment or maintenance, introduction of new flaws in attempts to fix old flaws, incremental escalation to inordinate complexity.

Software Quality Assurance & Testing

TOPIC1-LECTURE 3

Adverse Effects of Faulty Software

- **Communications:** Loss or corruption of communication media, non delivery of data.
- **Space Applications:** Lost lives, launch delays.
- **Defense and Warfare:** Misidentification of friend or foe.
- **Transportation:** Deaths, delays, sudden acceleration, inability to brake.
- **Safety-critical Applications:** Death, injuries.
- **Electric Power:** Death, injuries, power outages, long-term health hazards (radiation).

Adverse Effects of Faulty Software (Cont'd)

- **Money Management:** Fraud, violation of privacy, shutdown of stock exchanges and banks, negative interest rates.
- **Control of Elections:** Wrong results (intentional or non-intentional).
- **Control of Jails:** Technology-aided escape attempts and successes, accidental release of inmates, failures in software controlled locks.
- **Law Enforcement:** False arrests and imprisonments.

Bug in Space Code

- Project Mercury's FORTRAN code had the following fault:

  `DO I=1.10` instead of … `DO I=1,10`

- The fault was discovered in an analysis of why the software did not seem to generate results that were sufficiently accurate.
- The erroneous 1.10 would cause the loop to be executed exactly once!

Military Aviation Problems

- An F-18 crashed because of a missing exception condition: **if** ... **then** ... without the **else** clause that was thought could not possibly arise.
- In simulation, an F-16 program bug caused the virtual plane to flip over whenever it crossed the equator, as a result of a missing minus sign to indicate south latitude.

Shaky Math

- In the US, five nuclear power plants were shut down in 1979 because of a program fault in a simulation program used to design nuclear reactor to withstand earthquakes.
- This program fault was, unfortunately, discovered after the power plants were built!

Shaky Math (Cont'd)

- Apparently, the arithmetic sum of a set of numbers was taken, instead of the sum of the absolute values.
- The five reactors would probably not have survived an earthquake that was as strong as the strongest earthquake ever recorded in the area.

Making Rupee!

- An Australian man purchased $104,500 worth of Sri Lankan Rupees.
- The next day he sold the Rupees to another bank for $440,258.
- The first bank's software had displayed a bogus exchange rate in the Rupee position!
- A judge ruled that the man had acted without intended fraud and could keep the extra $335,758!

Software Quality Assurance & Testing

TOPIC1-LECTURE 4

Specification
"if you can't say it, you can't do it"

- You have to know what your product is before you can say if it has a bug.
- A *specification* defines the product being created and includes:
  - Functional requirements that describes the features the product will support. E.g., on a word processor
    - Save, print, check spelling, change font, …
  - Non-functional requirements are constraints on the product. E.g,
    - Security, reliability, user friendliness, platform, …

A software bug occurs when at least one of these rules is true

- The software does not do something that the specification says it should do.
- The software does something that the specification says it should not do.
- The software does something that the specification does not mention.
- The software does not do something that the product specification does not mention but should.
- The software is difficult to understand, hard to use, slow …

Most bugs are not because of mistakes in the code …

- Specification (~= 55%)
- Design (~= 25%)
- Code (~= 15%)
- Other (~= 5%)

Relative cost of bugs
"bugs found later cost more to fix"

- Cost to fix a bug increases exponentially (10^x)
  - i.e., it increases tenfold as time increases
- E.g., a bug found during specification costs $1 to fix.
- … if found in design cost is $10
- … if found in code cost is $100
- … if found in released software cost is $1000

Bug Free Software

- Software is in the news for the wrong reason
  - Security breach, Mars Lander lost, hackers getting credit card information, etc.
- Why can't software engineers develop software that just works?
  - As software gets more features and supports more platforms it becomes increasingly difficult to make it create bug-free.

Discussion …

- Do you think bug free software is unattainable?
  - Are their technical barriers that make this impossible?
  - Is it just a question of time before we can do this?
  - Are we missing technology or processes?