

Welcome to

CSE-433

Software Quality Assurance

Text Book

1. **Software Testing and Quality Assurance Theory and Practice**
 - By Kshirasagar Naik and Priyadarshi Tripathy

2. **Mastering Software Quality Assurance: Best Practices, Tools and Techniques**
 - By Murali Chemuturi

3. **Introduction to Software Testing (2nd Edition)**
 - By Paul Ammann and Jeff Offutt

Learning Objectives

- Students will become better testers
 - Knowledge and skills for creating high-quality tests at all levels
- Students will become better programmers
 - More aware of potential problems in software
 - Knowledge and skills for creating high-quality developer tests
- Students will become better engineers
 - A quality-first engineering mindset
 - Know how to both program and test in a seamless and unified manner
- Students will become better thinkers
 - Encouragement to approach software problem solving in logical, analytical ways

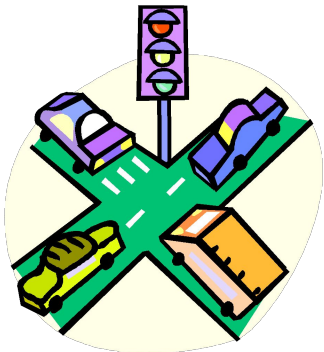
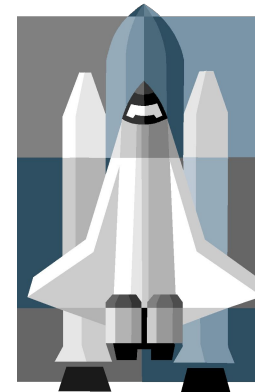
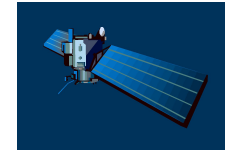
What is Software?

- Software is a program that enables a computer to perform a specific task.
- This includes:
 - **system software** such as an **operating system**, which enables other software to run properly, by interfacing with hardware and with other software.
 - **application software** such as a **word processor**, which enables a user to perform a task,
- Practical computer systems divide software into three major classes:
 - system software,
 - programming software and
 - application software,although the distinction is arbitrary, and often blurred.

Software in the 21st Century

- Software defines **behavior**
 - network routers, finance, switching networks, other infrastructure
- Today's software market :
 - is much bigger
 - is more competitive
 - has more users
- Embedded Control Applications
 - airplanes, air traffic control
 - spaceships
 - watches
 - ovens
 - remote controllers
 - memory seats
 - PDAs
 - DVD players
 - garage door openers
 - cell phones

Software is a Skin that Surrounds Our Civilization



Quote due to Dr. Mark Harman

The uniqueness of the software development process

- **High complexity**
- **Invisibility of the product**
- **Limited opportunities to detect defects (“bugs”)**

The characteristics of the SQA environment process

- Being contracted
- Subjection to customer-supplier relationship
- Requirement for teamwork
- Need for cooperation and coordination with other development teams
- Need for interfaces with other software systems
- Need to continue carrying out a project while the team changes
- Need to continue maintaining the software system for years

• What is Quality?

- Quality is defined as the product or services that should be "fit for use and purpose."
- Quality is all about meeting the needs and expectations of customers concerning functionality, design, reliability, durability, and price of the product.

What is Assurance?

- Assurance is a positive declaration on a product or service. It is all about the product which should work well. It provides a guarantee which would work without any problem according to expectations and requirements.

Def. (Software Quality)

- **IEEE definition**

Software quality is:

- (1) The degree to which a system, component, or process meets specified requirements.
- (2) The degree to which a system, component, or process meets customer or user needs or expectations.

Pressman's definition

Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

Quality Software

- **Quality software** refers to a software which is reasonably bug or defect free, is delivered in time and within the specified budget, meets the requirements and/or expectations, and is maintainable.
- In the software engineering context, software quality reflects both **functional quality** as well as **structural quality**.
 - **Software Functional Quality** – It reflects how well it satisfies a given design, based on the functional requirements or specifications.
 - **Software Structural Quality** – It deals with the handling of non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, and the degree to which the software was produced correctly.

Quality Software Cont...

- **Software Quality Assurance** – Software Quality Assurance (SQA) is a set of activities to ensure the quality in software engineering processes that ultimately result in quality software products. The activities establish and evaluate the processes that produce products. It involves process-focused action.
- **Software Quality Control** – Software Quality Control (SQC) is a set of activities to ensure the quality in software products. These activities focus on determining the defects in the actual products produced. It involves product-focused action.

Quality Control VS Quality Assurance

- **Quality control** is defined as “a set of activities designed to evaluate the quality of a developed or manufactured product” (IEEE, 1991); in other words, activities whose main objective is the withholding of any product that does not qualify. Accordingly, quality control inspection and other activities take place as the development or manufacturing of the product is completed yet before the product is shipped to the client.
- The main objective of **quality assurance** is to minimize the cost of guaranteeing quality by a variety of activities performed throughout the development and manufacturing processes/stages. These activities prevent the causes of errors, and detect and correct them early in the development process. As a result, quality assurance activities substantially reduce the rate of products that do not qualify for shipment and, at the same time, reduce the costs of guaranteeing quality in most cases.

The Software Quality Challenges

- In the software industry, the **developers will never declare that the software is free of defects**, unlike other industrial product manufacturers usually do.
- This difference is due to the following reasons.
 - **Product Complexity** - It is the number of operational modes the product permits. Normally, an **industrial product allows only less than a few thousand modes of operation** with different combinations of its machine settings. However, **software packages allow millions of operational possibilities**. Hence, assuring of all these operational possibilities correctly is a major challenge to the software industry.
 - **Product Visibility** - Since **the industrial products are visible**, most of its defects can be detected during the manufacturing process. Also the absence of a part in an industrial product can be easily detected in the product but, the **defects in software products which are stored on diskettes or CDs are invisible**.
 - The uniqueness of software quality assurance.
 - The environments for which SQA methods are developed.

Software vs Other Products

- Comparison **software products vs industrial products**:

| Characteristic | Software Products | Other Industrial Products |
|--|--|---|
| Complexity | Millions of operational options | Thousand operational options |
| Visibility of product | Invisible Product Difficult to detect defects by sight | Visible Product Effective detection of defects by sight |
| Nature of development and production process | Can detect defects in only development phase | Can detect defects in all of the following phases <ul style="list-style-type: none">•Product development•Product production planning•Manufacturing |

**How we will get quality
software?**

What is the Quality Attribute of a software?

1. Functionality- Quality of software is defined as how effectively the software interacts with other components of the system. The software must provide appropriate functions as per requirement, and these functions must be implemented correctly.

2. Reliability- It is defined as the capability of the software to perform under specific conditions for a specified duration.

3. Usability- Usability of software is defined as its ease of use. Quality of the software is also identified as how easily a user can understand the functions of the software and how much effort is required to follow the features.

What is the Quality Attribute of a software?

4. Efficiency- The efficiency of the software is dependent on the architecture and coding practice followed during development.

5. Maintainability- Maintainability is also one of the significant factors to define the quality of the software. It refers to identifying the fault and fixing in the software. It should be stable when the changes are made.

6. Portability- Portability of the software, defined as how easily a system adapts to changes in the specifications. Quality of the software is also determined by the portability of the system how easy it is to install the software and how easy it is to replace a component of the order in a given environment.

What is Software Testing?

- **Software testing** is a process, to **evaluate the functionality** of a software application with an intent to find whether the developed **software met the specified requirements or not** and to **identify the defects to ensure** that the product is **defect free** in order to produce the quality product.
- **According to ANSI/IEEE 1059 standard** – A **process of analyzing a software item to detect the differences between existing and required conditions** (i.e., defects) and to **evaluate the features** of the software item.

Software Faults, Errors & Failures

- **Software Fault** : A static defect in the software
- **Software Error** : An incorrect internal state that is the manifestation of some fault
- **Software Failure** : External, incorrect behavior with respect to the requirements or other description of the expected behavior

Faults in software are equivalent to design mistakes in hardware.

Software does not degrade.

Fault and Failure Example

- A patient gives a doctor a list of **symptoms**
 - Failures
- The doctor tries to **diagnose the root cause** of the ailment
 - Fault
- The doctor may look for **anomalous internal conditions** (high blood pressure, irregular heartbeat, bacteria in the blood stream)
 - Errors

Most medical problems result from external attacks (bacteria, viruses) or physical degradation as we age. Software faults were there at the beginning and do not “appear” when a part wears out.

A Concrete Example

Fault: Should start searching at 0, not 1

```
public static int numZero (int [ ] arr)
{
    // Effects: If arr is null throw NullPointerException
    // else return the number of occurrences of 0
    int count = 0;
    for (int i = 1; i < arr.length; i++)
    {
        if (arr [ i ] == 0)
        {
            count++;
        }
    }
    return count;
}
```

Test 1
[2, 7, 0]
Expected: 1
Actual: 1

Error: i is 1, not 0, on the first iteration
Failure: none

Test 2
[0, 2, 7]
Expected: 1
Actual: 0

Error: i is 1, not 0
Error propagates to the variable count
Failure: count is 0 at the return statement

The nine causes of software errors

1. Faulty requirements definition
2. Client–developer communication failures
3. Deliberate deviations from software requirements
4. Logical design errors
5. Coding errors
6. Non-compliance with documentation and coding instructions
7. Shortcomings of the testing process
8. Procedure errors
9. Documentation errors

The Term Bug

- *Bug* is used informally
- Sometimes speakers mean fault, sometimes error, sometimes failure ... often the speaker doesn't know what it means !
- This class will try to use words that have precise, defined, and unambiguous meanings

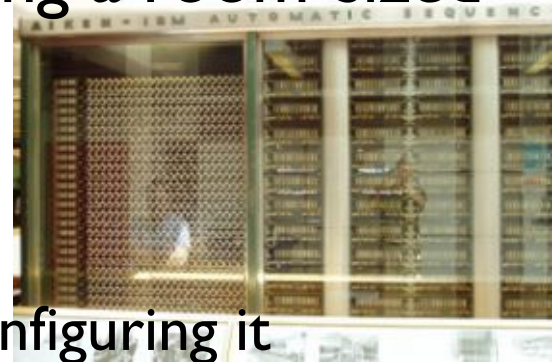


“It has been just so in all of my inventions. The first step is an intuition, and comes with a burst, then difficulties arise—this thing gives out and *[it is]* then that 'Bugs'—as such little faults and difficulties are called—show themselves and months of intense watching, study and labor are requisite. . .” – Thomas Edison

“an analyzing process must equally have been performed in order to furnish the Analytical Engine with the necessary operative data; and that herein may also lie a possible source of error. Granted that the actual mechanism is unerring in its processes, the cards may give it wrong orders.” – Ada, Countess Lovelace (notes on Babbage’s Analytical Engine)

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!
 - I am not making this up :-)



Bugs a.k.a. ...

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

We will use the term in proper way.

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.
- **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.

Adverse Effects of Faulty S/W

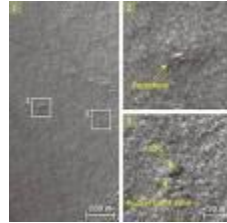
- **Communications:** Loss or corruption of communication media, non-delivery of data.
- **Space Applications:** Lost lives, launch delays.
- **Defense and Warfare:** Misidentification of friend or foe.
- **Transportation:** Deaths, delays, sudden acceleration, inability to brake.
- **Safety-critical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

Adverse Effects of Faulty S/W Cont

- **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.

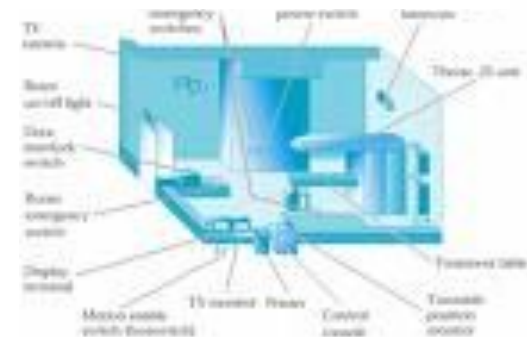
Spectacular Software Failures

- NASA's Mars lander: September 1999, crashed due to a units integration fault **Mars Polar**



Mars Polar Lander crash site?

THERAC-25 design



- THERAC-25 radiation machine : Poor testing of safety-critical software can cost *lives* : 3 patients were killed
- Ariane 5 explosion : Millions of \$\$
- Intel's Pentium FDIV fault : Public relations nightmare



**Ariane 5:
exception-handling
bug : forced self
destruct on maiden
flight (64-bit to 16-bit
conversion: about
370 million \$ lost)**

We need our software to be dependable.
Testing is one way to assess dependability.

Northeast Blackout of 2003

508 generating units and 256 power plants shut down

Affected 10 million people in Ontario, Canada

Affected 40 million people in 8 US states

Financial losses of \$6 Billion USD

The **alarm system** in the energy management system failed due to a **software error** and operators were not informed of the power overload in the system



Bank Generosity

- A Norwegian bank ATM consistently dispersed 10 times the amount requested.
- Many people joyously joined the queues as the word spread.

Bank Generosity (Cont'd)

- A software flaw caused a UK bank to duplicate every transfer payment request for half an hour.
- The bank lost 2 billion British pounds!
- The bank eventually recovered the funds but lost half a million pounds in potential interest.

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!

Bug in BoNY Software

- The Bank of New York (BoNY) had a \$32 billion overdraft as the result of a 16-bit integer counter that went unchecked.
- BoNY was unable to process the incoming credits from security transfers, while the NY Federal Reserve automatically debited BoNY's cash account.
- BoNY had to borrow \$24 billion to cover itself for one day until the software was fixed.
- The bug cost BoNY \$5 million in interest payments.

Costly Software Failures

- NIST report, “The Economic Impacts of Inadequate Infrastructure for Software Testing” (2002)
 - Inadequate software testing costs the **US alone between \$22 and \$59 billion annually**
 - Better approaches could cut this amount in half
- Huge losses due to web application failures
 - Financial services : **\$6.5 million per hour** (just in USA!)
 - Credit card sales applications : **\$2.4 million per hour** (in USA)
- In Dec 2006, *amazon.com*’s BOGO offer turned into a double discount
- 2007 : Symantec says that most **security vulnerabilities** are due to faulty software

World-wide monetary loss due to poor software is shocking

Testing in the 21st Century

- More **safety** critical, **real-time** software
- Embedded software is ubiquitous ... check your pockets
- Enterprise applications means bigger programs, more users
- Paradoxically, free software increases our expectations !
- **Security** is now all about software faults
 - Secure software is reliable software
- The web offers a new deployment platform
 - Very competitive and very available to more users
 - Web apps are distributed
 - **Web apps** must be highly reliable

***Industry desperately needs our
inventions !***

The objective of SQA in soft. maintenance

- 1) Assuring an acceptable level of confidence that the software maintenance activities will conform to the functional technical requirements.**
- (2) Assuring an acceptable level of confidence that the software maintenance activities will conform to managerial scheduling and budgetary requirements.**
- (3) Initiate and manage activities to improve and increase the efficiency of software maintenance and SQA activities.**

What Does This Mean?

Software testing is getting more important

**What are we trying to do when we test ?
What are our goals ?**