

CSE 403

Software Engineering Professional Practice

Lecture 4

Project Management, Quality Assurance and testing

Overview of today's lecture

- ❖ Project Management
 - ❖ Measures of Project Success
 - ❖ Causes of Project failure
 - ❖ Measures of Project Success
 - ❖ Software Quality
 - ❖ Software Quality Assurance
 - ❖ Software Testing
 - ❖ Characteristics of Software testing
 - ❖ Application of Software Testing
 - ❖ QA vs Testing
 - ❖ Price of having a bug in your software
 - ❖ Where else can things go wrong?
 - ❖ Some Software testing tools

Project Management

Project

A **project** is a [temporary] sequence of unique, complex, and connected activities having one goal or purpose and that must be completed by specific time, within budget, and according to specification.

Project Management

Project management is the process of scoping, planning, staffing, organizing, directing, and controlling the development of an acceptable system at a minimum cost within a specified time frame.

Measures of Project Success

- ❖ The resulting information system is acceptable to the customer.
- ❖ The system was delivered “on time.”
- ❖ The system was delivered “within budget.”
- ❖ The system development process had a minimal impact on ongoing business operations.

Causes of Project failure

1. Failure to establish upper-management commitment to the project
2. Lack of organization's commitment to the system development methodology
3. Taking shortcuts through or around the system development methodology
4. Poor expectations management
5. Premature commitment to a fixed budget and schedule
6. Poor estimating techniques
7. Overoptimism
8. The mythical man-month (Brooks, 1975)
9. Inadequate people management skills
10. Failure to adapt to business change
11. Insufficient resources
12. Failure to "manage to the plan"

Why is Software Project Management Difficult?

- ❖ The product is intangible
- ❖ Changing Requirements
- ❖ Resource Constraints
- ❖ Quality Assurance
- ❖ Risk Management
- ❖ People Management

Project Management Tools & Techniques

- ❖ **Gantt Chart:** a simple horizontal bar chart that depicts project tasks against a calendar.
- ❖ **Kanban Boards:** consists of a board and moving cards that represent activities.
- ❖ **Timesheets:** are a way to track the number of hours a person has worked over the course of a specific time, usually a week or more
- ❖ **Project Dashboard:** track metrics in the project
- ❖ **PERT chart:** is a graphical network model that depicts a project's tasks and the relationships between those tasks.
 - **Program Evaluation and Review Technique**

Quality Assurance and testing

Software Quality

Software quality is defined as “conformance to established requirements; the capability of a software product to satisfy stated conditions.

According to ISO 9126, software quality consists of:

- ❖ Functionality
- ❖ Reliability
- ❖ Usability
- ❖ Efficiency
- ❖ Maintainability
- ❖ Portability

Software Quality Assurance

Quality Assurance (QA) is the process of ensuring that the software product meets the specified quality standards

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 is to detect bugs

Characteristics of Software testing

- ❖ **Purpose:** The purpose of software testing is to identify any defects or gaps in the software in order to ensure the quality of the product.
- ❖ **Scope:** Software testing is focused on the functionality of the software and uncovering defects.
- ❖ **Involvement:** Software testers are actively involved in the testing process, executing tests, reporting issues, and verifying fixes.
- ❖ **Techniques:** Software testing typically involves the use of automation tools and manual tests.
- ❖ **Responsibilities:** Software testers are responsible for ensuring that the software meets the quality standards and requirements set out by the stakeholders.
- ❖ **Deliverables:** The deliverable of software testing is typically a report that includes a list of defects, a test summary, and recommendations for improvement.

Application of Software Testing

- ❖ **Unit Testing:** Unit testing is a type of software testing where individual units or components of the software are tested. A unit is the smallest testable part of any software.
- ❖ **Integration Testing:** Integration testing is a type of software testing that verifies the interfaces between components against a software design.
- ❖ **System Testing:** System testing is a type of software testing that verifies the system's compliance with specified requirements. This type of testing is often done on a complete, integrated system.
- ❖ **Acceptance Testing:** Acceptance testing is a type of software testing that verifies if the system meets the customer's requirements.
- ❖ **Performance Testing:** Performance testing is a type of software testing that verifies the performance of a system. It is done to determine the speed, scalability, and stability of a system under a particular workload.
- ❖ **Security Testing:** Security testing is a type of software testing that is done to verify the security of a system. It is done to ensure that the system is secure and protected against any malicious attacks or threats.

QA vs Testing

Software Quality Assurance

- ❖ Process-oriented activity
- ❖ Oriented to bug prevention

Software Testing

- ❖ Product-oriented activity
- ❖ Oriented to bug detection

So what is a 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!

So what is a bug?

- ❖ Defect
- ❖ Fault
- ❖ Problem
- ❖ Error
- ❖ Incident
- ❖ Anomaly
- ❖ etc.

Most bugs are not always from code

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

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.

Price of having a bug in your software

- ❖ The cost of removing a bug grow exponentially once you system is in production,
- ❖ So it is always better to detect bugs before shipping.
- ❖

Price of having a bug in your software example

Example:

- ❖ 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
- ❖ 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

Where else can things go wrong?

- ❖ **Communications:** Loss or corruption of communication media, non delivery of data.
- ❖ **Space Applications:** Lost lives, launch delays.
- ❖ **Defence 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).
- ❖ **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.

Some Software testing tools

❖ Test Management

- TestRail
- Zephyr (for Jira)
- qTest
- PractiTest

❖ Test Automation tools

- Selenium (Web apps)
- Playwright / Puppeteer (Modern browser automation)
- Cypress (JavaScript-based web testing)
- TestComplete
- Katalon Studio Appium (Mobile testing)
- Robot Framework

Some Software testing tools

- ❖ Frameworks for Unit testing
 - JUnit (Java)
 - NUnit / MSTest / xUnit (.NET)
 - pytest / unittest (Python)
 - Mocha / Jasmine / Jest (JavaScript)
- ❖ Performance & Load Testing Tools
 - JMeter
 - Gatling
 - LoadRunner
 - k6
 - BlazeMeter

Some Software testing tools

- ❖ Continuous Integration Tools (CI/CD)
 - Jenkins
 - GitHub Actions
 - GitLab CI/CD
 - Azure DevOps
- ❖ Bug/Issue Tracking Tools
 - Jira
 - Bugzilla
 - MantisBT
 - Redmine
 - YouTrack

Some Software testing tools

❖ API Testing Tools

- Postman
- SoapUI
- Insomnia

❖ Security Testing Tools

- OWASP ZAP
- Burp Suite
- SonarQube
- Nessus
- Fortify