CS1632: Software Quality Assurance

Lecture 1: Introduction

History of Software Testing

Sources:

- Gelperin, D.; B. Hetzel (1988). “The Growth of Software Testing”.
- “Bug”, The Jargon File, ver. 4.4.7.
- Turing, Alan (October 1950), “Computing Machinery and Intelligence”.

The Debugging-Oriented Period (pre 1956)

- Focused on hardware.
- The first “bug”.
- Testing not considered a separate part of the software development lifecycle (SDLC).
- Alan Turing writes the first article that discusses software testing.

The Demonstration-Oriented Period (1957 - 1978)

- Charles Baker is the first to identify two different kinds of testing.
  - Verification - Did we build the product right?
  - Validation - Did we build the right product?
- Testing still focuses on the “happy path”.
- Employers begin posting job listings explicitly requesting testing skills.
- Companies begin to realize the cost of deficiencies in complex computer applications.

The Destruction-Oriented Period (1979 - 1982)

- Glenford J. Myers publishes the *The Art of Software Testing* and defines testing as:
  - “The process of executing a program with the intent of finding errors.”
- “Sad path” testing.
- Quality assurance (QA) activities begin to emerge.
  - Before this, testing has focused on quality control (QC).

QA vs. QC

- QA focuses on defect prevention. QC focuses on defect detection.
- Example:
  - Bob is a tester that is tasked with insuring the quality of a flight reservation system.
  - Bob identifies a need for requirements documentation for the product so that the team knows what to build. **QA.**
  - Bob identifies a need for unit testing and agrees with the team that the product should meet unit testing code coverage standards. **QA.**
  - Bob writes test plans that insure the product meets specified requirements. **QC.**
  - Bob identifies a discrepancy between the product and the documentation and enters a bug report. **QC.**

The Evaluation-Oriented Period (1983 - 1987)

- The Institute for Computer Sciences and Technology publishes *Guideline for Lifecycle Validation, Verification, and Testing of Computer Software*.
- QA is defined as a process that begins at product inception and spans all the way to product delivery.

The Prevention-Oriented Period (1988 - present)

- Testers realize, the earlier they find a defect, the cheaper it is to fix.
- QA activities begin focussing on preventing bugs before they occur.
- Test driven development (TDD) is introduced.
- The Golden Rule of Testing is that you should find defects as early as you can.

Responsibilities of the tester

- It’s not the tester’s responsibility to find every defect in an application.
- It’s the tester’s responsibility to understand and describe the quality of an application.
- Testers act as advocates for the customer.
- The tester’s job doesn’t begin with a handoff from a development team.
- The tester’s job spans the entire lifecycle of a product.
- Testing is not the tester’s only job.
- Testing is not only the tester’s job.

There’s still a lot of testing.

Unit testing, automated testing, acceptance testing, requirements analysis, equivalence classes, white/grey/black box testing, verification, validation, combinatorial testing, performance testing, usability testing, formal analysis, static analysis, linting, TDD, fuzz testing, software profiling, resource analysis, usability analysis, regression testing, smoke testing, security analysis, penetration testing

Why test software?

According to the National Institute of Standards and Technology, software defects cost the US economy almost $60 billion in 2002.

Testers tend to have a broad perspective of the system as a whole.

The reason for testing software boils down to minimizing risk.