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?