

Software Testing

CS 272 Software Development

An Introduction to Software Testing...

- Could be an entire **lecture**
- Could be an entire **course**
- Could be an entire **degree**
- Could be an entire **profession**
- Could be an entire **field of study**



Software Testing Questions

Who

What

When

Where

Why

How



Software Testing Questions

Who

What

When

Where

Why

How

- **Who performs testing?**
 - Developers? End users? Third party groups?
- **What (attributes) are you testing?**
 - Correctness? Efficiency? Flexibility?



Software Testing Questions

Who

What

When

Where

Why

How

- **When do you perform testing?**
 - At start, throughout or end of development cycle?
- **Where (or what level) do you perform testing?**
 - Individual components? Interactions? Entire system?



Software Testing Questions

Who

What

When

Where

Why

How

- **Why (what objectives) are you performing testing?**
 - Validation or verification?
- **How are you performing testing?**
 - Methodology? Automated? Toolkit?



Who Performs Testing?

- **Developers**

- + Allows immediate fixes
- Lowest cost?
- Too close to code?

- **End Users**

- + Realistic usage
- Limited to functionality

- **Third Party Groups**

- + No bias
- + Can examine code
- + Can test functionality
- Less familiar with code
- Very expensive



What Attributes To Test?

Operation

- Correctness
- Reliability
- Efficiency
- Usability
- Security
- Integrity

Revision

- Maintainability
- Testability
- Flexibility

Transition

- Portability
- Reusability
- Interoperability

https://en.wikipedia.org/wiki/List_of_system_quality_attributes



When Perform Testing?

- **Before** code development
 - Create tests before code, incrementally develop functionality to pass tests (test-driven)
- **Throughout** the development cycle
 - Continuous, test after each phase
- **End** of development cycle
 - After functionality developed, before reaching customer



What Level of Testing?

Unit Testing



Integration Testing



System Testing



Why Perform Testing?

- **Verification**

- Have we built the **system right**?
- *e.g. Did we build a calculator that can't add correctly?*

- **Validation**

- Have we built the **right system**?
- *e.g. Did we build a calculator when we needed a phone?
(both have number buttons after all!)*



How Perform Testing?

- **Who** (developers, users, third party) is doing the testing?
- **What** quality attributes are you testing?
- **When** (in the development cycle) are you testing?
- **What** level (unit, integration, system) are you testing?
- **Why** are you testing (verification vs validation)?
- **How** (which approach) will you take based on above?



Testing Approaches

- Accuracy versus usability versus accessibility versus performance versus load versus ... testing
- Open/clear/transparent ~~white~~ box versus closed/opaque ~~black~~ box testing
- Coverage versus fault versus error-based testing
- Fuzz testing versus mutation testing
- ...and many more

https://en.wikipedia.org/wiki/Software_testing



OSS-Fuzz

- Continuous fuzzing for open-source software
- Offered as a cloud-service for "critical" open source projects or run locally
- Found 25k+ bugs in 375 open source projects since 2020
- Itself also an open-source project supported by Google

<https://github.com/google/oss-fuzz> and <https://google.github.io/oss-fuzz/>





CHANGE THE WORLD FROM HERE