CS1632, Lecture 7: Automated vs. Manual Testing

Wonsun Ahn

Manual Testing

- What we have been doing so far
- A human runs the test plan on the software

Automated Tests

- Mostly what we’ll be doing from here on out
- A computer runs the test plan for us

- Makes a huge difference
  --- Typically a single test plan is run thousands of times

Manual / Automated Testing Comparison

Type	Pros	Cons
Manual Testing		
Automated Testing		

Manual / Automated Testing Comparison

Type	Pros	Cons
Manual Testing	+ Can be performed by unskilled staff + Does not require test automation tools + Flexible: humans can adapt a test case to small changes in software + Broad: humans can notice issues with software even if not explicitly in test case	
Automated Testing		

Manual / Automated Testing Comparison

Type	Pros	Cons
Manual Testing	+ Can be performed by unskilled staff + Does not require test automation tools + Flexible: humans can adapt a test case to small changes in software + Broad: humans can notice issues with software even if not explicitly in test case	- Time and resource intensive - Issues with reproducibility (e.g. if test is timing-sensitive, humans are not good at that) - Prone to human error - Limited to testing through user-interface (e.g. can’t do unit testing)
Automated Testing		

Manual / Automated Testing Comparison

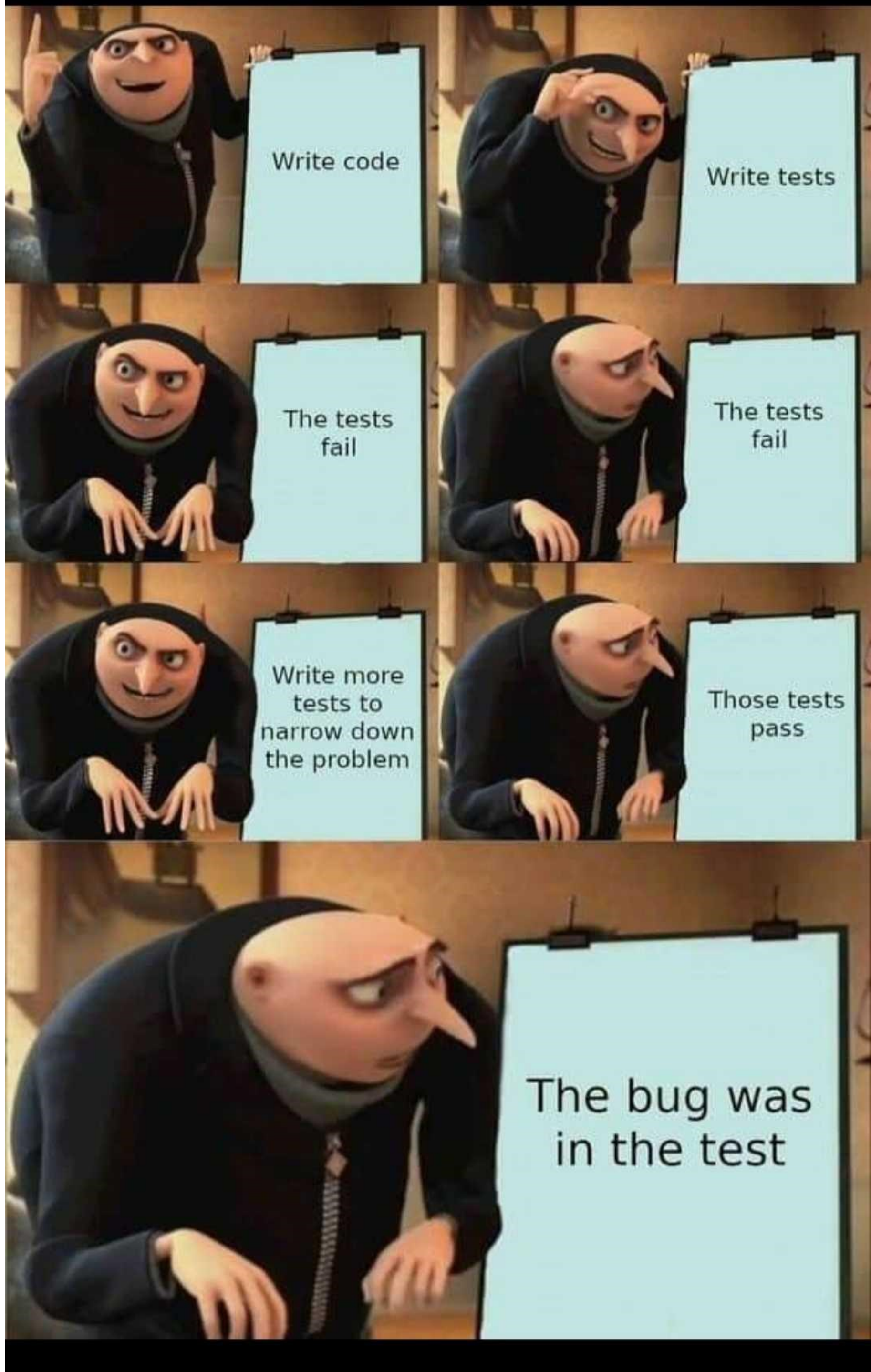
Type	Pros	Cons
Manual Testing	+ Can be performed by unskilled staff + Does not require test automation tools + Flexible: humans can adapt a test case to small changes in software + Broad: humans can notice issues with software even if not explicitly in test case	- Time and resource intensive - Issues with reproducibility (e.g. if test is timing-sensitive, humans are not good at that) - Prone to human error - Limited to testing through user-interface (e.g. can’t do unit testing)
Automated Testing	+ Once set up, faster and cheaper to run + Better at reproducing steps exactly + No chance of human error + Can do unit testing using tools like JUnit	

Manual / Automated Testing Comparison

Type	Pros	Cons
Manual Testing	+ Can be performed by unskilled staff + Does not require test automation tools + Flexible: humans can adapt a test case to small changes in software + Broad: humans can notice issues with software even if not explicitly in test case	- Time and resource intensive - Issues with reproducibility (e.g. if test is timing-sensitive, humans are not good at that) - Prone to human error - Limited to testing through user-interface (e.g. can't do unit testing)
Automated Testing	+ Once set up, faster and cheaper to run + Better at reproducing steps exactly + No chance of human error + Can do unit testing using tools like JUnit	- Time up-front to build testing infrastructure - Need skilled staff to build testing infrastructure - Requires testing tools (and learning them) - Brittle: small changes in software may cause an automated test case to fail (costly to maintain) - Narrow: tests only what is explicitly scripted

Automated Testing is Brittle

- A machine is not flexible like a human being
- Even small changes in app cause tests to fail
  - E.g. a small change in the GUI of an app
  - Tests need to be updated with the app
  - Ultimately, test code becomes part of code base (And needs to be maintained as such)
  - Soon, you spend as much time debugging tests as you spend debugging your app!
- “Self-healing” tech makes tests resilient
  - But tech is far from mature



Automated Testing is Narrow

- “There are known knowns. These are things we know that we know. There are known unknowns. That is to say, there are things that we know we don't know. But there are also unknown unknowns. There are things we don't know we don't know.” -Donald Rumsfeld
- Automated test cases only check “known unknowns”
  - Things that test cases specify we should check
- Automated test cases do not check “unknown unknowns”
  - Things that nobody even thought to include in test cases
- Manual testing can check for “unknown unknowns”
  - Humans notice things in the course of running a test case even if not specified

Solution: A Mixture

- Most teams will use both manual and automated tests
- Automated tests typically far outnumber manual tests
    - Automated tests do most of the heavy lifting
    - Manual tests to make sure automated tests did not miss anything
        - E.g. the GUI “looks” right in terms of layout
        - E.g. the application functions well from the end-user perspective
- Sometimes full automation is possible
    - Especially if software doesn’t have a UI (e.g. platform, library, server)
    - Large software companies often have comprehensive enough automated test suites that they can deploy code without manual testing

Writing Automated Test Cases

- Automated test cases are identical to manual test cases.
  - IDENTIFIER
  - TEST CASE DESCRIPTION
  - PRECONDITIONS
  - EXECUTION STEPS
  - POSTCONDITIONS

- Except we have to be even MORE specific
  - Computers do exactly what they’re told to do
  - (Some) humans have common sense

Automated Testing: Blackbox or Whitebox?

- Answer: Can be both!
- Blackbox testing: testing at the *system* level using app interface
  - Script invokes application with various command line arguments
  - Script interacts with app, supplying user-input, and verifying program output
  - There are even scripts for GUI apps: will later learn a tool called *Selenium*
- Whitebox testing: (typically) testing at the *unit* level
  - Script directly invokes methods in app source code with test arguments
  - Script verifies each method by checking return values and/or postconditions
  - Often relies on frameworks: will learn *JUnit* (a framework for Java testing)

Now Please Read Textbook Chapter 12