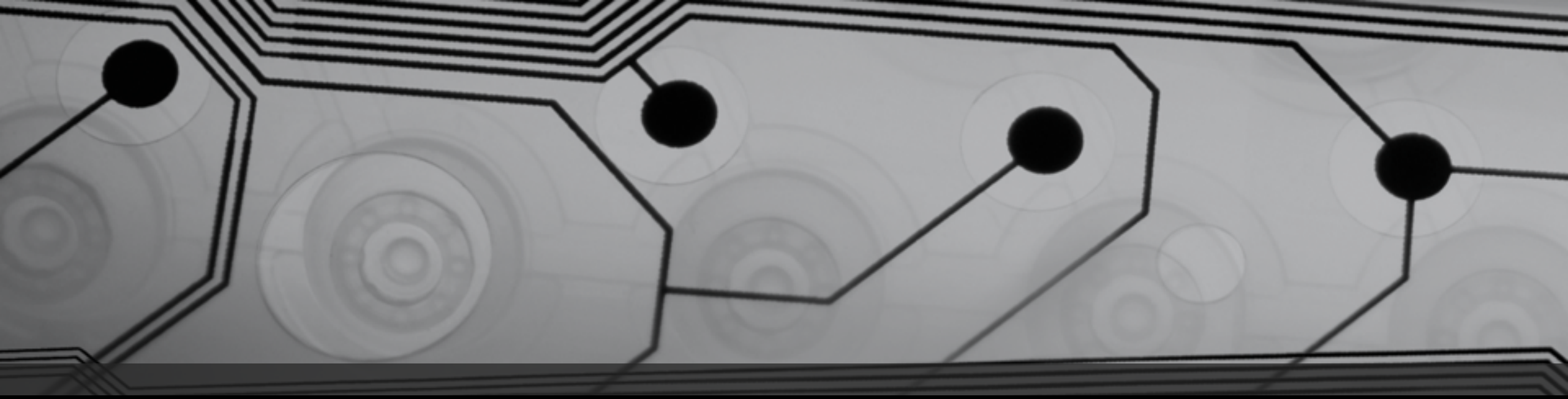

CS1632, Lecture 8: Manual vs Automated Testing

Bill Laboon

Manual Testing

- What we have been doing so far
- We write test plans
- A human executes them on the software

Automated Tests

- Mostly what we'll be doing from here on out
- We write tests which the computer executes for us

Benefits of Manual Testing

1. It’s simple!
2. It’s cheap (at first)
3. It’s easy to set up
4. No additional software to learn or write
5. Flexible
6. Can focus on things users care about
7. Humans catch issues that programs may not notice

Drawbacks of Manual Testing

1. It is BORING
2. It can be unrepeatable
3. Some tasks are difficult to test manually, e.g.:
    1. Timing
    2. Low-level interfaces
4. Human error is a possibility
5. It’s time and resource-intensive

Benefits of Automated Testing

1. No chance for human error (during execution)
2. Fast test execution
3. Easy to execute once set up
4. Repeatable
5. Less resource-intensive during testing
6. Ideal for testing some things that manual testing is bad for

Drawbacks of Automated Testing

1. Requires extra time up-front
2. May not catch user-facing bugs
3. Requires learning tools and frameworks (but that's one of the things this class can help with)
4. Requires more skilled staff
5. Big issue: It only tests what it is looking for

“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

Solution: A Mixture

- Most teams will use both manual and automated tests
- Usually, the number of automated tests will far outnumber the number of manual tests

When To Automate Tests?

- When in doubt, automate!
- If it can be automated in a reasonable amount of time, do it.
- However, humans should usually review things before release.
- Not in all cases! Google and other large companies often have comprehensive enough test suites that they can deploy code without human review.

Writing Automated Tests

Remember what you need for manual tests. Automated tests will be similar. We may have be even MORE specific for automated tests (computers are so literal-minded).

IDENTIFIER

TEST CASE DESCRIPTION

PRECONDITIONS (if any)

EXECUTION STEPS:

:POSTCONDITIONS (if any):

Input and Output Values

- Additional “kinds” of preconditions and postconditions
- **Input value** – value that you will be testing (e.g. an argument to a method)
- **Output value** – value that you expect will be output (e.g. the return value of a method)
- In manual testing, can be considered part of pre- and postconditions
- Often directly tested separately in automated tests

Minimize external dependencies

- You should be testing your own code, not third-party providers
- We'll discuss how to do this for events that seem to require it
  - Unit Testing lectures
  - Writing Testable Code lecture

Minimize randomness, emphasize repeatability

- Tests should be repeatable
- Use specific values, not random numbers
- Do not test different things in a test
  - If you use an “if..else” in a test, this is a code smell!

Tests should be independent

- They should not depend on each other or their ordering
- Allows faster execution
- Allow broader execution even in the event of failure

Tests should be specific

- You will need to be even more precise than your manual tests
  - Computers do exactly what they're told to do
  - (some) humans have common sense