

COMS W4156 Advanced Software Engineering (ASE)

November 3, 2022

Agenda

1. Questions about First Individual Assessment
2. Continuous Integration
3. Github Actions demo



Date _____ COMPLETED

1)	_____	<input type="checkbox"/>
2)	_____	<input type="checkbox"/>
3)	_____	<input type="checkbox"/>
4)	_____	<input type="checkbox"/>
5)	_____	<input type="checkbox"/>
6)	_____	<input type="checkbox"/>
7)	_____	<input type="checkbox"/>
8)	_____	<input type="checkbox"/>
9)	_____	<input type="checkbox"/>
10)	_____	<input type="checkbox"/>

First Individual Assessment

due 11:59pm November 4, that's tomorrow!



First Assessment Contribution to Overall Grade

First assessment ~= Midterm exam

20% of overall grade

20% matters!



Must submit by uploading to this assignment in courseworks, we will not grade assessments submitted any other way

First Assessment Timing



The assessment is due by 11:59pm
this Friday

Should take about four hours but you
have up to four days to work on it

Not timed!

Questions?

Now is your last chance to me ask questions, the IAs may continue answering on EdDiscussion.



Agenda

1. Questions about First Individual Assessment
2. **Continuous Integration**
3. Github Actions demo

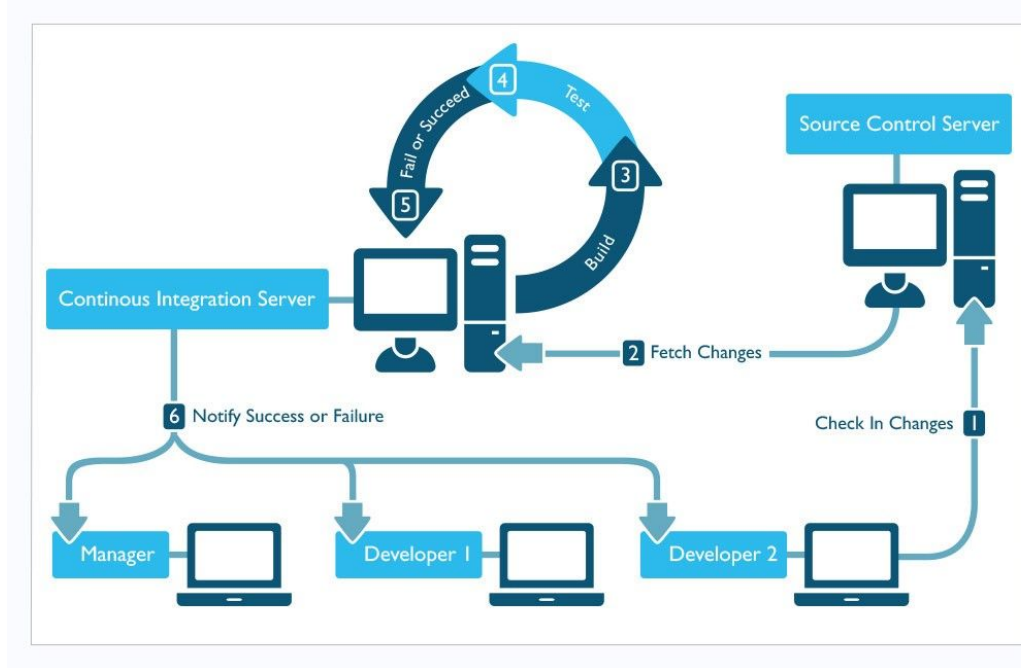


Continuous Integration (CI)

Tools like [Travis CI](#) and [Circle CI](#) hook into a version control system (like github) and automatically re-build after every commit to the shared repository

Can also run CI on pre-commit (partial build) and/or on a timer (e.g., [nightly build](#))

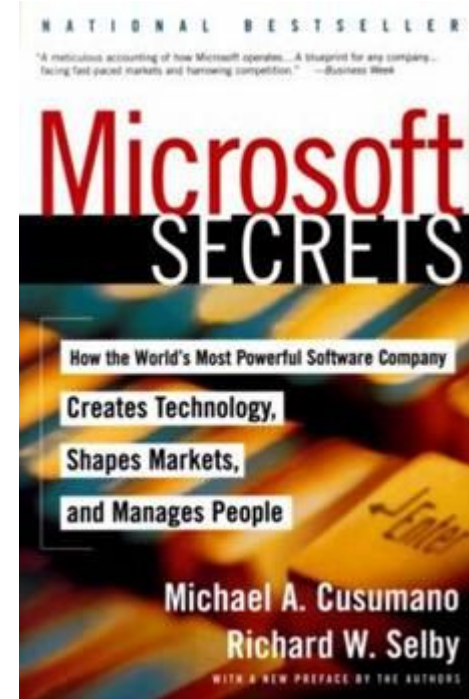
Goal is to detect errors as soon as they are introduced



Don't Break The Build

Breaking the build = Errors detected during CI were traced to *your* commit

If all developers had fully tested their changes locally (on their own machine) prior to commit, and fixed any errors prior to commit, *in theory* CI should not find any new errors



In Theory vs. In Practice

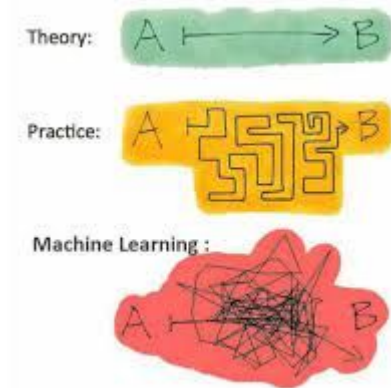
Developers test their own changes locally with only a subset of the full test suite

Sometimes just the unit tests for the changed units

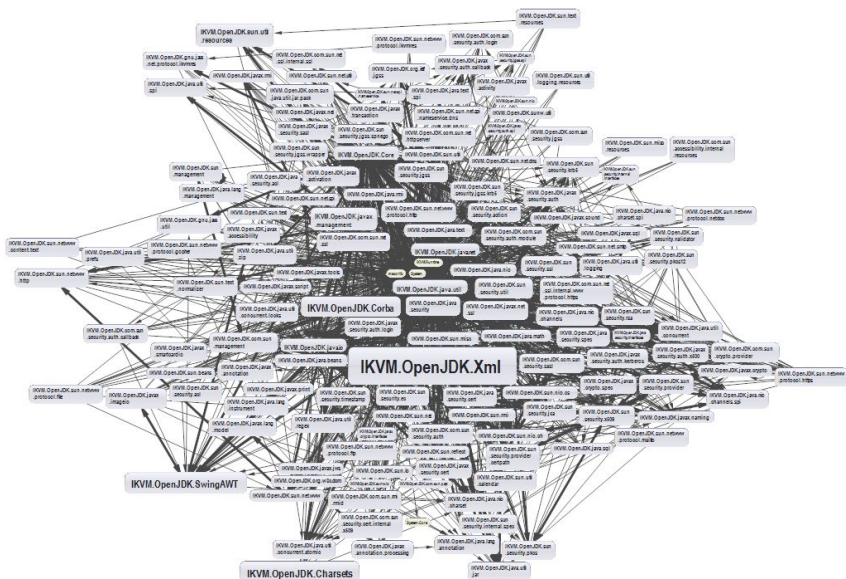
Plus maybe integration tests involving the changed units

Developer machines typically do not have the resources to run the system's full regression test suite

And developers do not have the time to wait for full builds in a test environment



I Thought Build Meant Compile



Build often refers to gathering all dependencies, followed by compile and link (compiled languages) or just gathering all dependencies (non-compiled languages)

In CI context, “build” also runs testing and analysis tools

Which test cases actually run when may vary - in some organizations there are multiple levels of CI running designated tests

Test Selection

A common choice for commit-time testing is to employ a “test minimization” strategy that selects only new/modified test cases plus those existing test cases whose results (pass or fail) could have been affected by the committed code changes ([change analysis](#))



This may be a relatively small subset of all the tests, but there's a cost to accurately computing the affected tests without actually running any tests

Change Analysis



Track all code c exercised by each test t during *previous* testing

If the commit changes c , or changes any other code that c depends on directly or indirectly (determined by static analysis), then re-run t

It may be less expensive to run full regression testing than to run full static analysis

Regression Testing

Regression testing is the typical choice for nightly builds

A *regression* is when fixing a bug or adding a new feature causes previously working code to break, or “regress”, sometimes code that seems completely unrelated in another part of the codebase

Regression:
"when you fix one bug, you
introduce several newer bugs."



Detecting Regressions



To detect all regressions, you need to re-run *all* test cases that previously passed - and also all tests that previously failed, in case they now fail in different ways or (unexpectedly) pass. In other words, the whole test suite

Of course, if the code change fixed a bug, you would expect the corresponding test case(s) that detected the bug to now pass. The developers should run those previously failed tests before committing

But other test cases that do not seem relevant may also now pass due to dependencies in the application (incomplete or incorrect change analysis) or in the test cases themselves (hidden dependencies)

More on test selection and test dependencies later in semester

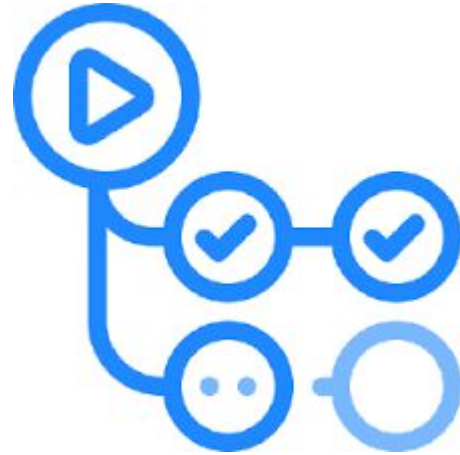
Agenda

1. Questions about First Individual Assessment
2. Continuous Integration
3. Github Actions demo



Github Actions demo: Prahlad Koratamaddi

[GitHub Actions demo presentation](#)

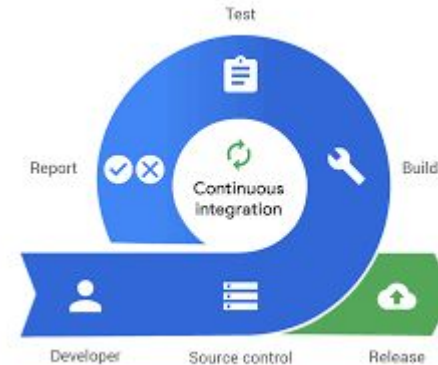


Read the Docs

[Github Actions Documentation](#)

[CI/CD Explained](#)

[Github Actions as a DevOps Platform](#)



Upcoming Assignments

[First Individual Assessment](#)

due 11:59pm November 4, that's tomorrow!



[Second Iteration](#) due November 28

[Second Iteration demo](#) due December 5

Next Week

No class Tuesday - Monday and Tuesday are Columbia “holidays” for the US elections

Thursday - finish github actions demo,
Integration Testing



Ask Me Anything