Special Topics

Yaay!!
Administrivia

- HW9 will be out by tomorrow. Is due about 10 days from tomorrow
- HW8 is due on Wednesday, with 1 late day allowed that can be up to 120 hours
- **No staff support** from Thursday through Monday evening for HW8
- Lots of Extra OH this week, but get to it sooner because a lot of the staff will not be available on Wednesday, so no expectations for Piazza/Email replies
- Those who submit HW8 on Wednesday will get feedback by Monday. Others will get it by next Friday (after HW9 is due)
- Mike doesn’t have OH this week. Avidant’s OH are extended today. Look at the calendar for more details of who/when/where for the next 60 hours
Typical Development Lifecycle

1. Get the source code
2. Install dependencies
3. Compile the code
4. Run static analysis
5. Generate documentation
6. Run tests
7. Create artifacts for customer
8. Ship!
Which of these do we have to manually?

1. Get the source code
2. Install dependencies
3. Compile the code
4. Run static analysis
5. Generate documentation
6. Run tests
7. Create artifacts for customer
8. Ship!

NONE!
NADA!
HECK NO!
ZIPPITY NOO WOP!
Build Systems
What is a build system?

A tool that does all of these tasks automatically:

1. Get the source code
2. Install dependencies
3. Compile the code
4. Run static analysis
5. Generate documentation
6. Run tests
7. Create artifacts for customer
8. Ship!
Tasks we’ve seen

- Build
- Test
- Reformat
- validateWorkingCopy
- validateLocalRepository
- validateRemoteRepository

- Generally in every build system
- Specific to this class
When you run "./gradlew build"

1. Some stuff prints to the console
2. ???
3. Everything breaks!
When you run “./gradlew build”

1. Some stuff prints to the console
2. ???
3. Everything breaks – works!
When you run “./gradlew build”

1. Installs all dependencies
2. Compiles all files
3. Ensures files are in correct location
4. Run tests
5. Runs static analysis tools
6. Everything breaks—works!
When you run "./gradlew validate...”

- Cleans the directory
- Runs everything “build” runs
- Checks whether required files in the correct location
- “validateLocalRepository” also checks all files are committed
- “validateRemoteRepository” also checks all commits are pushed
Which task gets to go first?

Compile hw3 → Run hw3 tests

Compile hw4 → Run hw4 tests

Run hw3 tests → validate...

Compile hw4 → Run hw4 tests

Run hw4 tests → validate...

Which task gets to go first?
Which task gets to go first?

- Compile hw3
- Run hw3 tests
- Compile hw4
- Run hw4 tests
- Validate...
Which task gets to go first?

- Compile hw3
- Run hw3 tests
- Compile hw4
- Run hw4 tests

validate...
Which task gets to go first?

Compile hw3 -> Run hw3 tests

Compile hw4 -> Run hw4 tests

Run hw3 tests -> validate...

Run hw4 tests -> validate...
Which task gets to go first?

- Compile hw3
- Run hw3 tests
- Compile hw4
- Run hw4 tests
- validate...
Which task gets to go first?

- Compile hw3
- Run hw3 tests
- Compile hw4
- Run hw4 tests
- validate...
Git
Basic Git Commands you all know (kinda)

- pull
- add
- commit
- push
What happens to the commits?

- git log
- git log --author=???
- git log --oneline
- git log --pretty=short/medium
- git log --graph
What is the history?

Pick a random commit

`git show <SHA>`

`git tag`

Pick a random tag

`git show <TAG-NAME>`
But that’s annoying

git blame .gitignore

git blame ci/pipeline.yml

git blame -e .gitignore
Navigating the history

`git checkout <COMMIT-SHA>`

`git checkout <TAG-NAME>`
Undoing mistakes

git revert <COMMIT-SHA>

git revert HEAD~3
Branches

git branch

git branch -a

git checkout -b <NEW BRANCH NAME>

git branch -D <BRANCH-NAME>
Stash

- `git stash`
- `git stash show`
- `git stash`
- `git stash show`
- `git stash list`
- `git stash show stash@{0}`
- `git stash pop`
Javascript