Software Dev

It works on my machine and how to develop more professionally

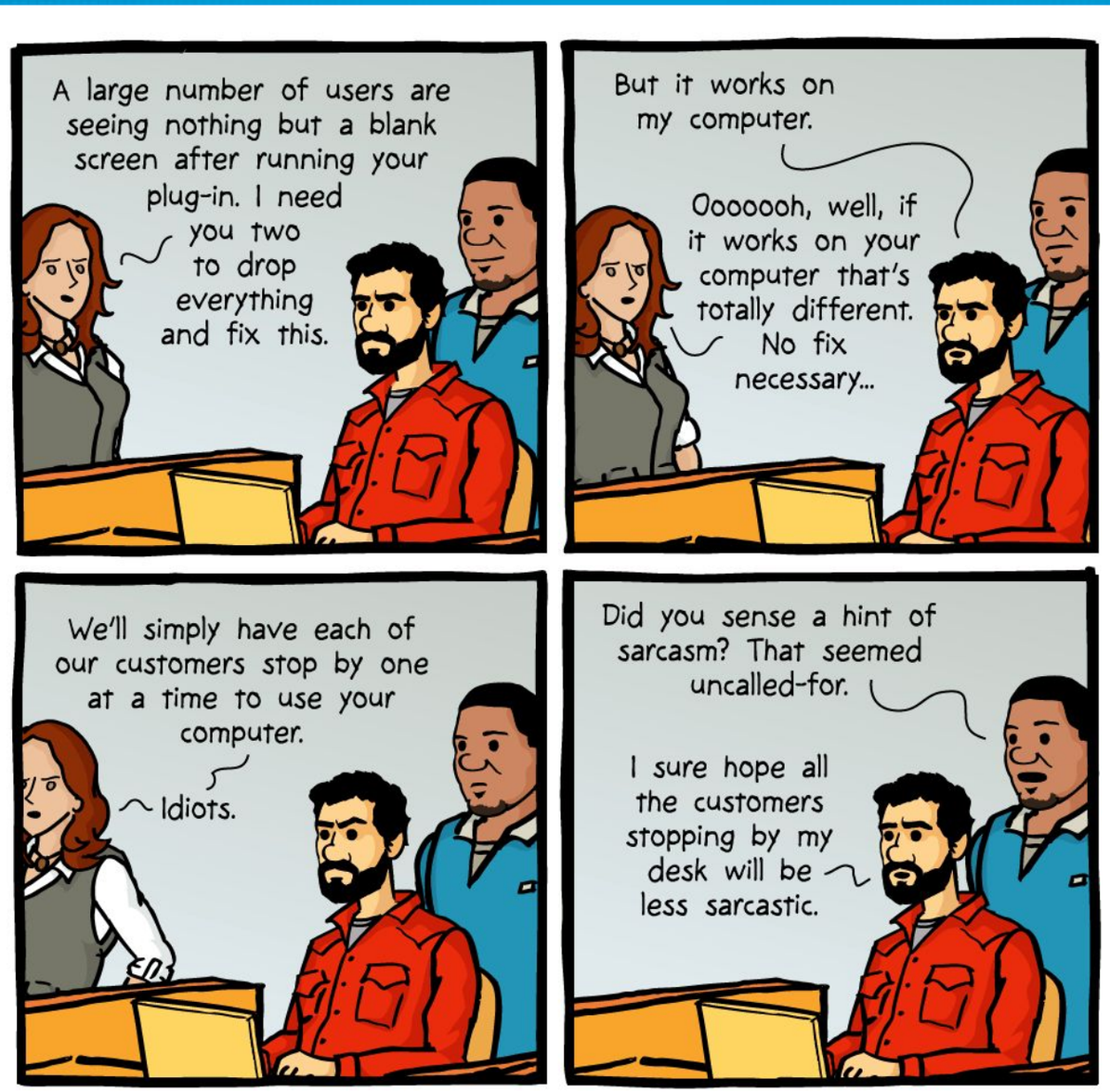
Admin

- Quizzes /assignment discussion
- Assignment here?
- Or next class?

Works on my machine

- Discuss the history
  - For a while I got a reputation for this
  - So I dropped it
  - But this semester I’m hearing more and more works on my machine.
  - You don’t get downloaded and installed with the software

From my door


A large number of users are seeing nothing but a blank screen after running your plug-in. I need you two to drop everything and fix this.
But it works on my computer.
Oooooooh, well, if it works on your computer that's totally different. No fix necessary...
We'll simply have each of our customers stop by one at a time to use your computer.
Idiots.
Did you sense a hint of sarcasm? That seemed uncalled-for.
I sure hope all the customers stopping by my desk will be less sarcastic.
©2016 Jeff Lofvers
Don't Hit Save - donthitsave.com

Works on my machine

- What we are doing here is (perhaps unconsciously) blaming the users
  - Yes I say this in jest (users are the worst thing that can happen to your software)
  - But in real life our software should take users into account

Works on My machine

- As per articles, several reasons that it doesn't work for your customers
  - Bad user interface
    - customer doesn't know how to use the program
  - Broken build/untested/buggy
    - Someone someone snuck a bug into production code.
  - Environmental problems
    - Missing/confused dependencies/'dll hell'
  - Lets take a look at these one at a time

Bad User interface

- The most infamous example in the year 2019 of bad user interface?
  - Including an initial corporate response that was functional equivalent to blaming the user?

Bad User interface

- The most infamous example in the last year of bad user interface?
  - Including an initial corporate response that was functional equivalent to blaming the user?
  - Boeing 737 MAX
    - “A long-standing procedure taught to pilots could have halted the dive, according to the regulator and the manufacturer. The FAA ordered airlines to add an explanation into flight manuals,”

      - Boeing response to the Lion Air crash Oct 2018
  - When did Boeing add information about the AI system that caused the crash to the manuals and other information given to pilots?

Bad User interface

- The most infamous example in the last year of bad user interface?
  - Including an initial corporate response that was functional equivalent to blaming the user?
  - Boeing 737 MAX
  - When did Boeing add information about the AI system that caused the crash to the manuals and other information given to pilots?
    - A week after the first crash
    - https://www.extremetech.com/extreme/280521-boeing-737-crash-caused-by-new-safety-system-pilots-werent-told-existed

Lesson

- If your interface kills people it isn’t ‘operator error’
  - We appear to finally be leaving this era
- Three Mile Island story
  - The beginning of the ‘operator error’ approach
    - As far as I know

Bad User Interface Example 1

- One of our current administrative software:
  - How do you supposed we select an application?
  - See next slide

ellucian | Recruitment | Recruitment Home

Apps for Dynamics 365 View Dynamics 365 information on the go with apps for your phone, tablet, Outlook, and more! Get Apps for Dynamics 365

HELP SAVE AS NEW EDIT DELETE SHARE DASHBOARD ASSIGN REFRESH ALL

Graduate Admissions Appl...

GR - Applications Ready for Grad Coordinator Rev...

Prospect (Application)	Academic Program (Application)	Anticipated Entry T...	Admit Type (Applica...	Folder Status	MSW Advan...	MSW Cohort Preference 1 (A...	MSW Cohort Preference 2 (...	MSW Cohort Preference 3 (...	MSW Cohort Preferenc

No Application Folder records found.

GR - Application Reviews with Reviewer

Search for records

Prospect (Application Folder)	Decision	Review Stat...	Entry Term (Opportunity Id)	Academic Program (Opportu...	Folder Status (Application F...	Decision Plan (Oppo...	MSW Cohor...	SI
[redacted] Renuka		Completed	2019 FALL	Computer Science, MS	Reviewer Assigned	International		K...
[redacted] Thejaswini		Completed	2019 FALL	Computer Science, MS	Reviewer Assigned	International		K...
[redacted] ram reddy		Completed	2019 FALL	Computer Science, MS	Reviewer Assigned	International		K...
[redacted] ng		Completed	2019 FALL	Computer Science, MS	Reviewer Assigned	International		K...
[redacted] n		Completed	2019 FALL	Computer Science, MS	Reviewer Assigned	Regular Decision		K...
[redacted] M.		Completed	2019 FALL	Computer Science, MS	Reviewer Assigned	Regular Decision		K...
[redacted] yn		In Progress	2019 FALL	Computer Science, MS	Reviewer Assigned	Regular Decision		K...

GR - Applications Reviewed by Coordinat...

First Name ...	Last Name (...	Academic Program (Application)	Anticipated...	Proposed D...	Folder Status	Student's A...

No Application Folder records found.

Bad User Interface Example 1

- One of our current administrative software:
  - How do you supposed we select an application?
  - See previous slide
  - Discuss
  -

Bad User Interface example 2

- Banner purchasing
  - ‘cheap’ banner vs full-banner (order of magnitude price difference)
  - Show page selections from the 2021 banner “user guide”

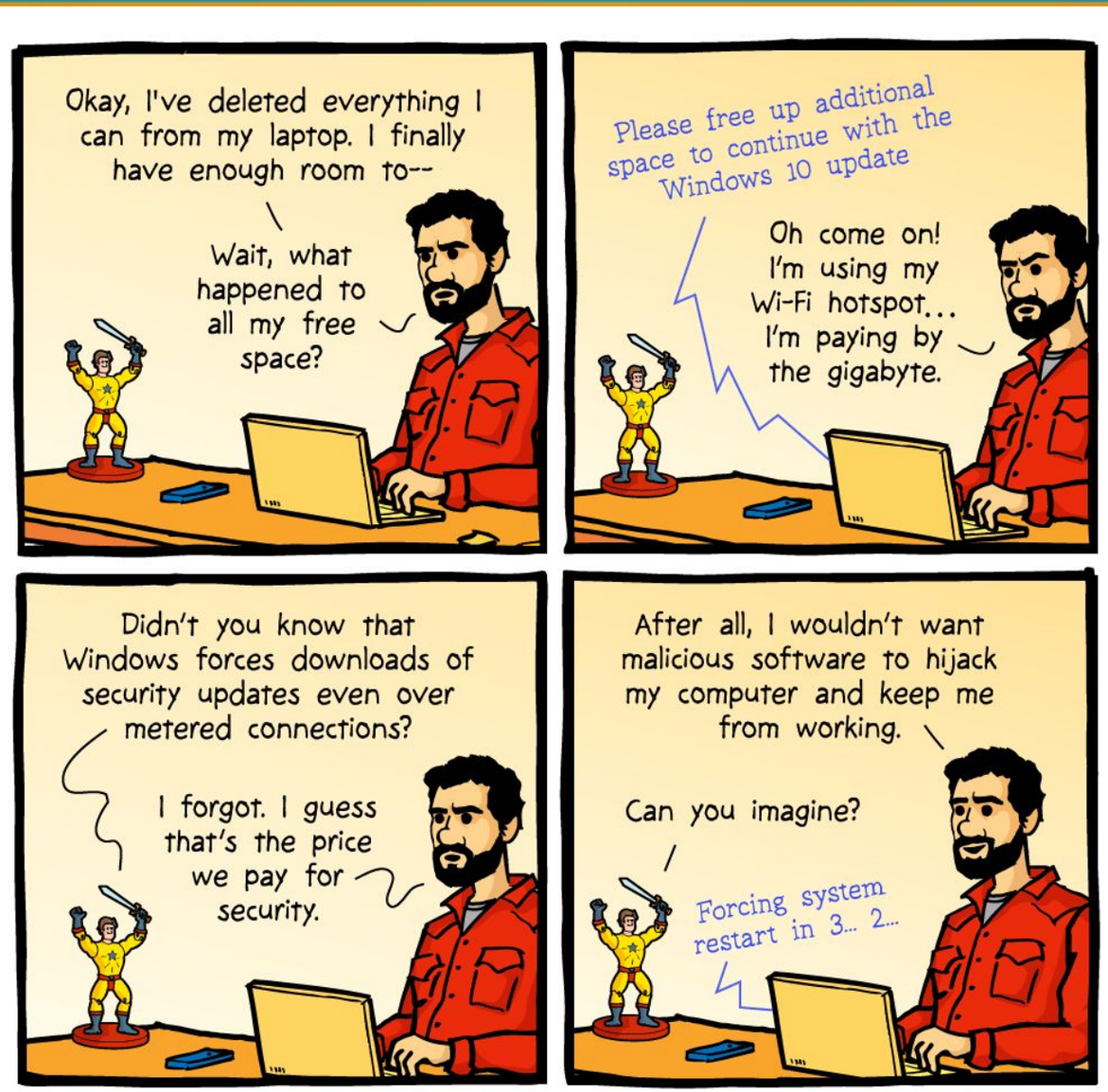
Bad User Interface 3

- According to industry publication The Inquirer
  - Not Enquirer you wags,
  - What is the 2nd worst UI of all time?

Bad User Interface 3

- According to industry publication The Inquirer
  - Not Enquirer you wags,
  - What is the 2nd worst UI of all time?
  - https://www.theinquirer.net/inquirer/feature/2459940/top-10-worst-user-interfaces/page/

They also call out this



Final Example

- Remember the ddos attack from a few years back that took down the east coast internet?
- October 21 2016
  - https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/
- Why? What happened?

Final Example

- Remember the ddos attack from a few years back that took down the east coast internet?
- October 21 2016
  - https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/
- IoT devices taken over and a good chunk of the East Coast Internet went offline.
- why/how were these IoT devices taken over and added to a botnet?
- Hint – what do these devices tell you to do first?

Final Example

- Remember the ddos attack from a few years back that took down the east coast internet?
- October 21 2016
  - https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/
- IoT devices taken over and a good chunk of the East Coast Internet went offline.
- why/how were these IoT devices taken over and added to a botnet?

Final Example

- Remember the ddos attack from a few years back that took down the east coast internet?
- IoT devices taken over and a good chunk of the East Coast Internet went offline.
- why/how were these IoT devices taken over and added to a botnet?
- Users never changed the default password
-

Final Example

- IoT devices taken over and a good chunk of the East Coast Internet went offline.
- why/how were these IoT devices taken over and added to a botnet?
- Users never changed the default password
- How would you as developers fix this issue with users?

Final Example

- IoT devices taken over and a good chunk of the East Coast Internet went offline.
- why/how were these IoT devices taken over and added to a botnet?
- Users never changed the default password
- How would you as developers fix this issue with users?
  - Make device not work till the default password is changed
    - Good first start.

Usability

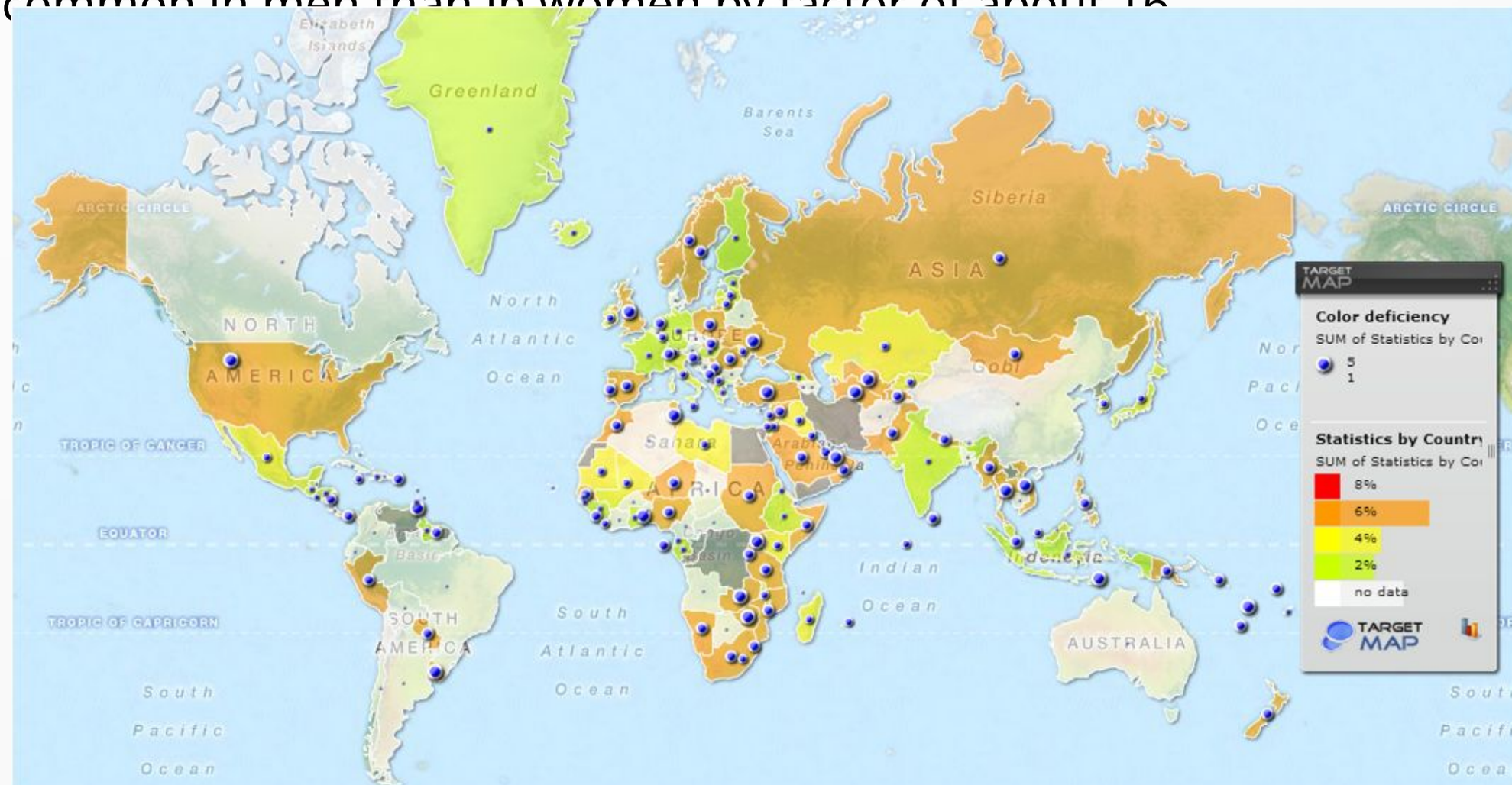
- Not all of you will be working on user facing tech
  - But all of you should be at least passingly familiar with first principles.
  - Dr. Liang covered several of these right?
  - At least accessibility?
  - Like?

Usability

- Not all of you will be working on user facing tech
  - Dr. Liang covered several of these right?
  - Accessibility?
    - Color schemes that work even for color-blind individuals
    - How common is color-blindness?
    - ref
      - https://medium.com/@courtneyjordan/designing-for-all-users-why-you-should-care-about-color-blindness-beabd61943eb

Usability

- How common is color-blindness?
  - More common in men than in women by factor of about 16



Usability

- Not all of you will be working on user facing tech
  - Accessibility?
    - Legal blindness.
      - Can your app be read by a text reader?
    - Deafness
      - Does your app require some auditory cues as the only way to function?
    - More?

Affordances

- 30+ years ago Don Norman wrote The Design of Everyday Things
  - Became **the** book for design in both engineering and CS design
  - 2013 revised edition is #1 and #5 in amazon's best seller list for the retail industry
    - #22 and #30 in the entire industry best seller list.

Affordances

- 30+ years ago Don Norman wrote The Design of Everyday Things
  - Perhaps the most important part of the book was the bring cognitive/perceptual psychologist James J. Gibson's concept of affordances to the area of design.

Affordances

- Affordances:
  - Originally: perceptual inputs which took no cognition to understand what they were
  - In the Norman sense
    - An understanding of a ‘thing’ and its uses which is almost instinctual
    - The perceived properties of how a thing is used
      - Eg knobs are for turning.

Affordances

- Give me more examples of affordances in the real world

Affordances

- Give me more examples of affordances in the real world –
  - for example if one of these was passed around when you were 12 what did you want to do to it?

Affordances

- What do each of these afford?
  - Light switch?
  -
  -

Affordances

- What do each of these afford?
  - Light switch?
  - Electric outlet?
  -
  -

Affordances

- What do each of these afford?
  - Light switch?
  - Electric outlet?
  - Glass panel?
  - Plywood panel?
    - Discuss Norman’s experience with these last two (and DMF)
  -
  -

Affordances

- What do each of these afford?
  - Light switch?
  - Electric outlet?
  - Glass panel?
  - Plywood panel?
    - Discuss Norman's experience with these last two (and DMF)
  - Drywall panel example
  -

Affordances

- What do each of these afford?
  - Light switch?
  - Electric outlet?
  - Glass panel?
  - Plywood panel?
    - Discuss Norman’s experience with these last two (and DMF)
  - Button?
  -

Affordances

- What do each of these afford?
  - Light switch?
  - Electric outlet?
  - Glass panel?
  - Plywood panel?
    - Discuss Norman's experience with these last two (and DMF)
  - Button?
  - Underlined blue text?
  -

Affordances

- What do each of these afford?
  - Light switch?
  - Electric outlet?
  - Glass panel?
  - Plywood panel?
    - Discuss Norman's experience with these last two (and DMF)
  - Button?
  - Underlined blue text?
  - Flat surface about 1.5 to 2 feet above the ground?

Affordances

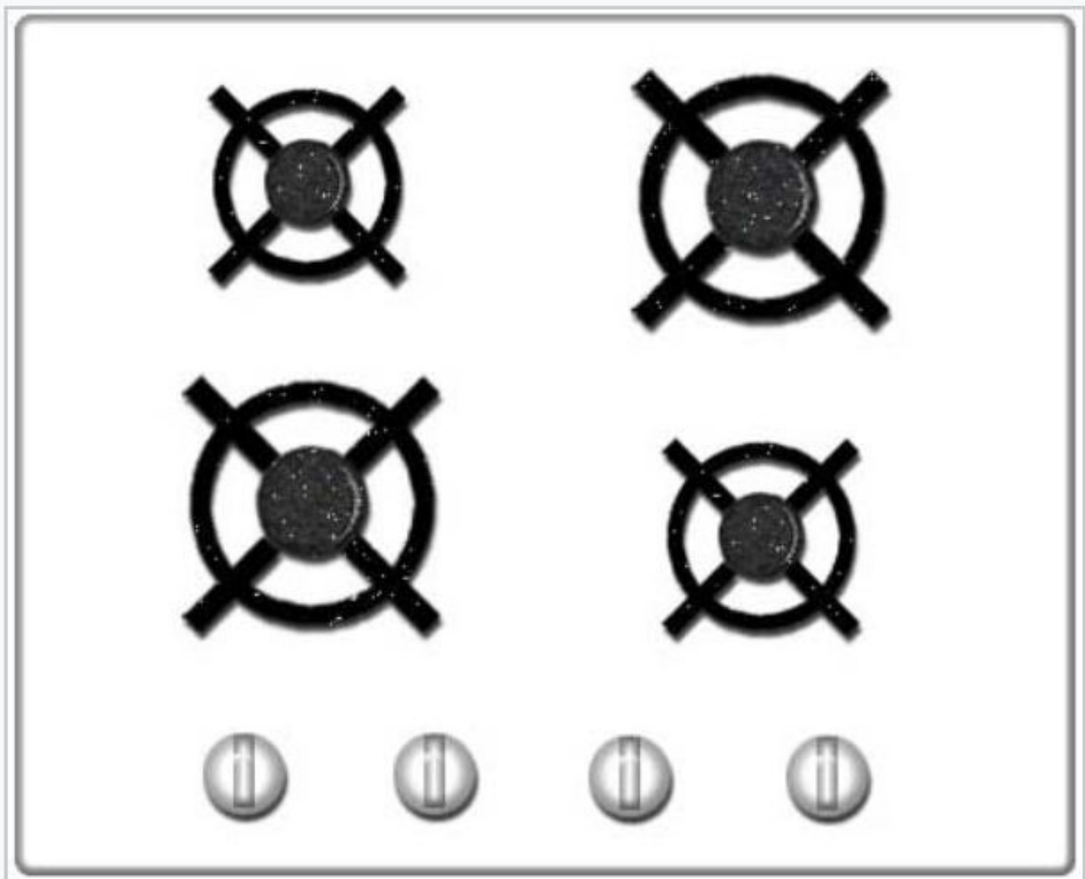
- Use Affordances to make your UI easier
- Don’t subvert affordances without very careful thought.
- Credit: Vox media article on Don Norman

Mapping

- Mapping is a second seminal concept codified in the Design of Everyday things
  - Mapping is the notion that the connection from inputs to functionality should be easy to understand

Bad Mapping

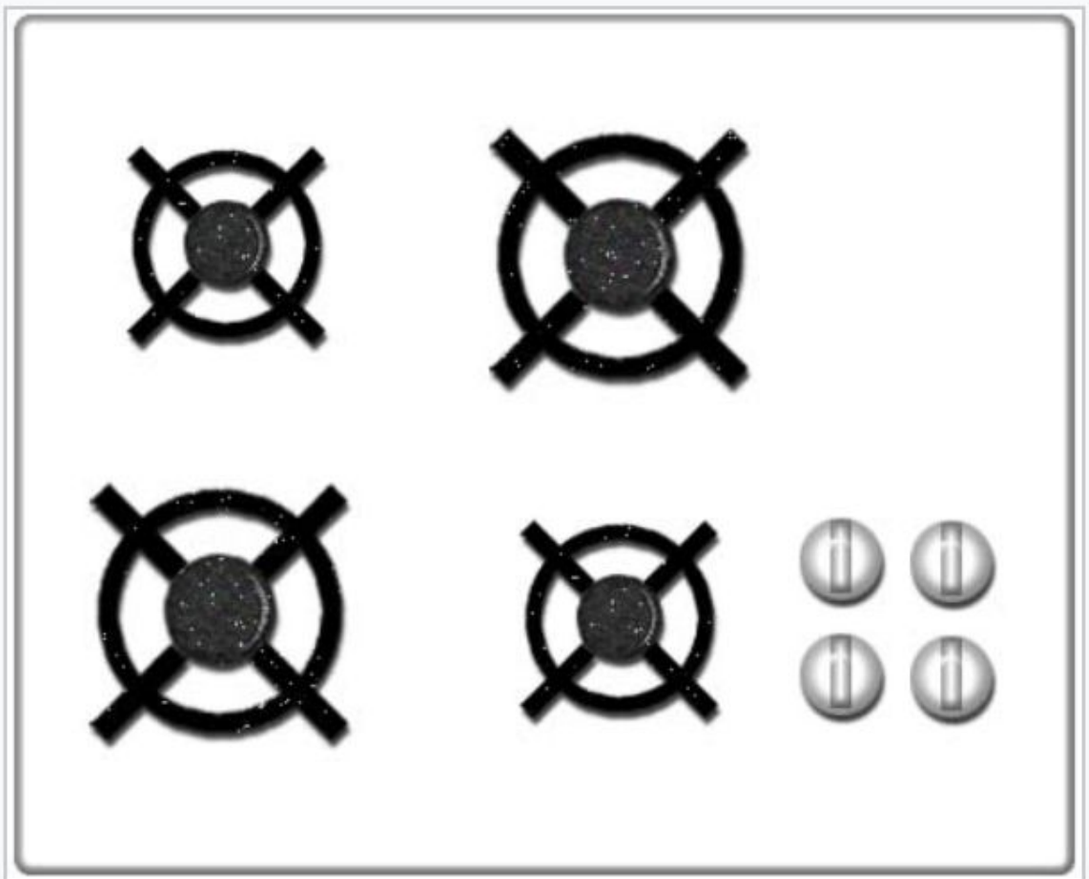
- Which control controls which burner in this stove? image credit wikipedia



Kitchen stove with arbitrary controls in a row

Natural mapping

- Which control goes with which burner here?
  again wikipedia credit



Kitchen stove with full natural mapping of Controls

Think about mapping

- When designing think about mapping
  - And feedback, when something happens/is happening make it clear to the user

Next: testing always

- Once your program is usable now we need to make sure your tests are always run before any commits of the software
- And once the tests pass and code is committed – immediately deploy it to production
- Continuous Integration (CI)/Continuous Delivery (CD)

CI: Intro

- Continuous Integration:
  - Basic idea: every time you commit/check in your code a script runs to do some work
  - Typical work:
    - Run automated tests
    - Run linters (pylint sonarlint for java etc)
    - Run formatters (black, rustfmt, gofmt etc)
    - For compiled languages, actually compile
  - If any of these fail then the check in/commit fails.
  - Frequently build software to shorten the feedback cycle

Why CI

- Don’t break the build
  - Assuming a Git-like model
  - If Developer A merges changes that break the build
    - No one else can do work

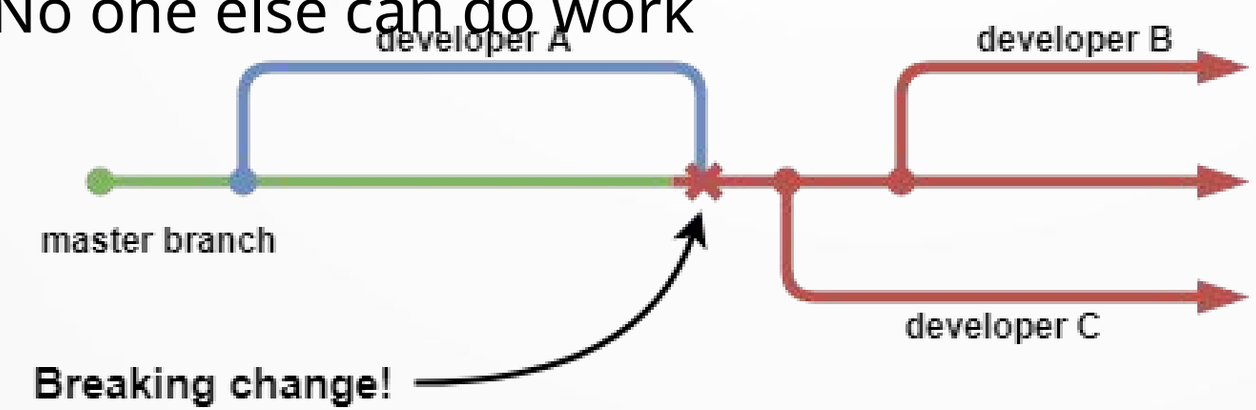


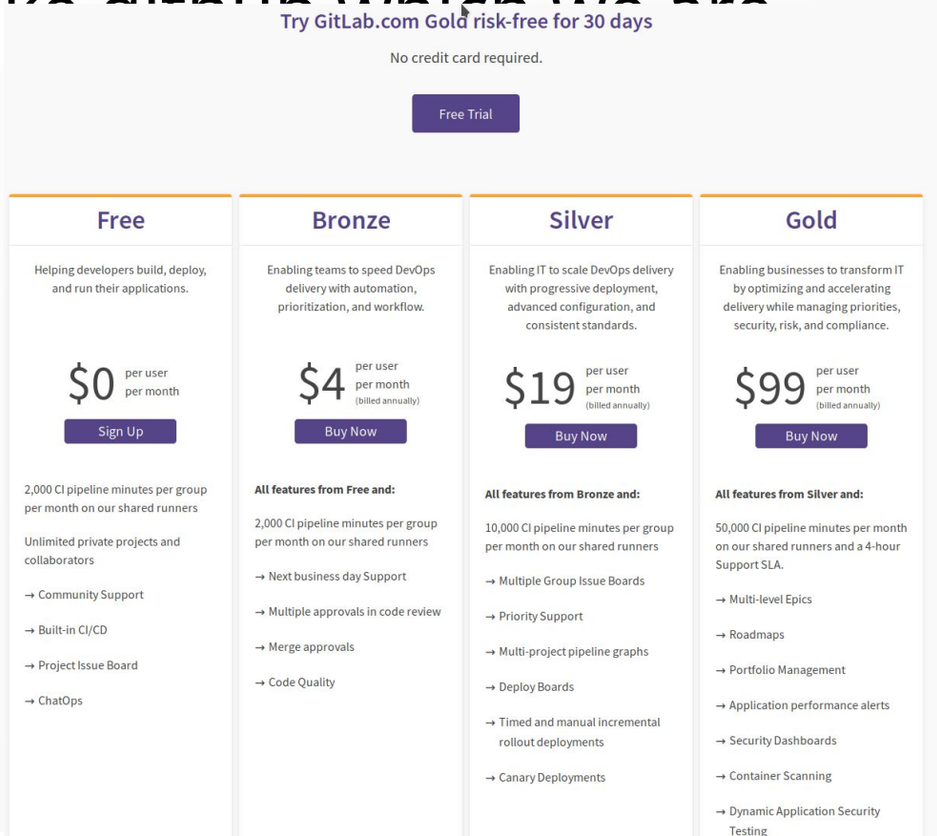
Image credit: Realpython.com

Lots of CI choices

- Today there are a number of options for CI
  - Jenkins: granddaddy of them all
    - New version out recently
    - Originally needed to run on-prem server
  - TravisCI
    - Gained traction because of github integration
  - CircleCI
    - Runs as a service – hosted on their servers
  - GitLab
    - Gitlab launched as a github competitor
    - Added build in CI as a competitive option
  - And more (Atlassian etc)

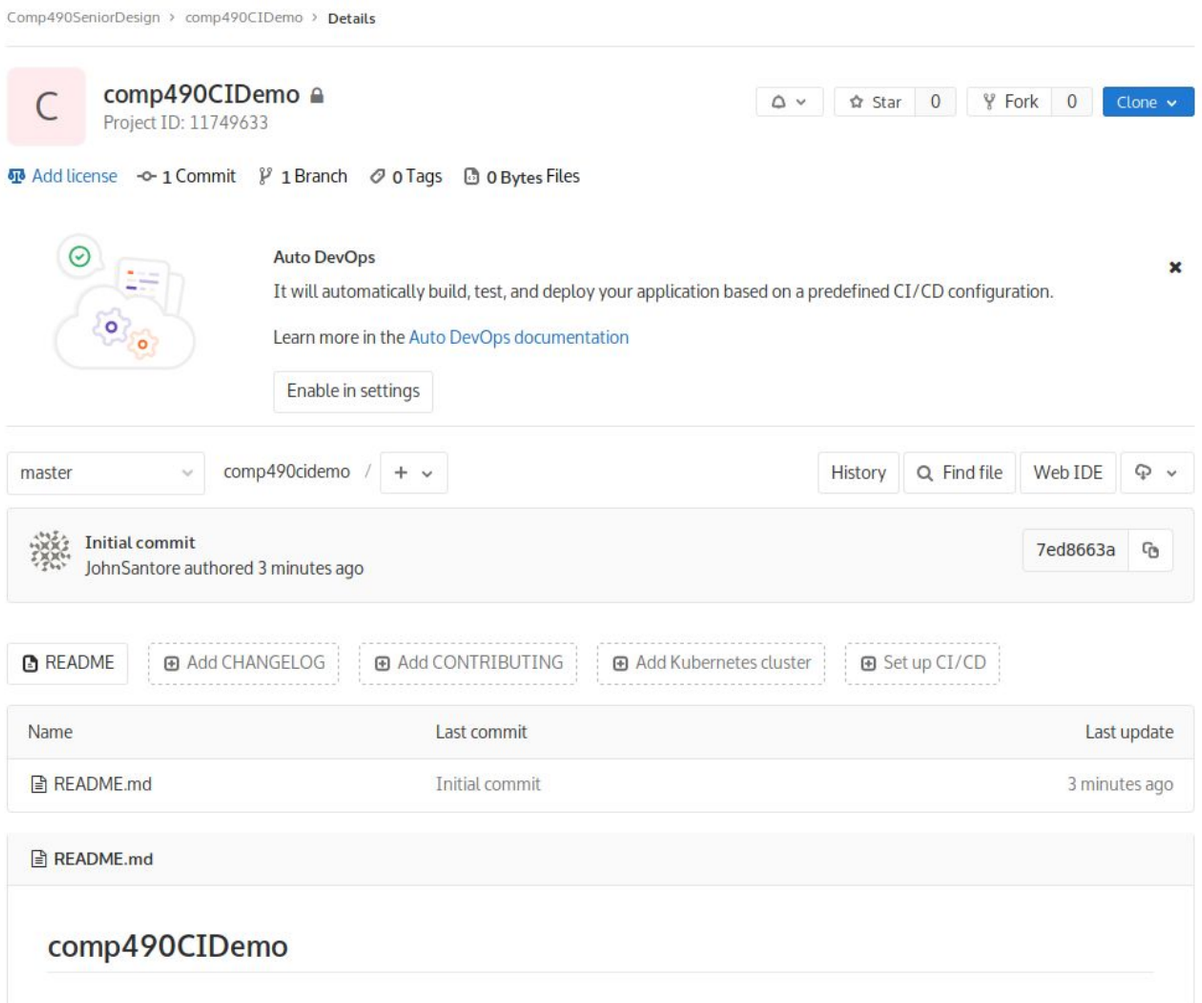
Gitlab CI

- I’m going to use Gitlab CI for most of my examples
  - Available on their free tier
  - Another cloud hosted git repository like github which we are already using.
  - Gitlab is a github competitor in the cloud git repository space
  - Distinguishing itself with this sort of offering



GitLab: new project

- A newly created project in gitlab:
- Lets look at what is different than github
- What follows is an edited diary of my learning so you can too



CI Pipelines

- Central to most CI workflows is the notion of a CI pipeline
- Multiple steps possible in the pipeline
  - Each step can have multiple parallel jobs running
  - Each step can use the output from earlier steps
  - But each job in a step can’t assume output from its peers

Pipelines

Example Here:

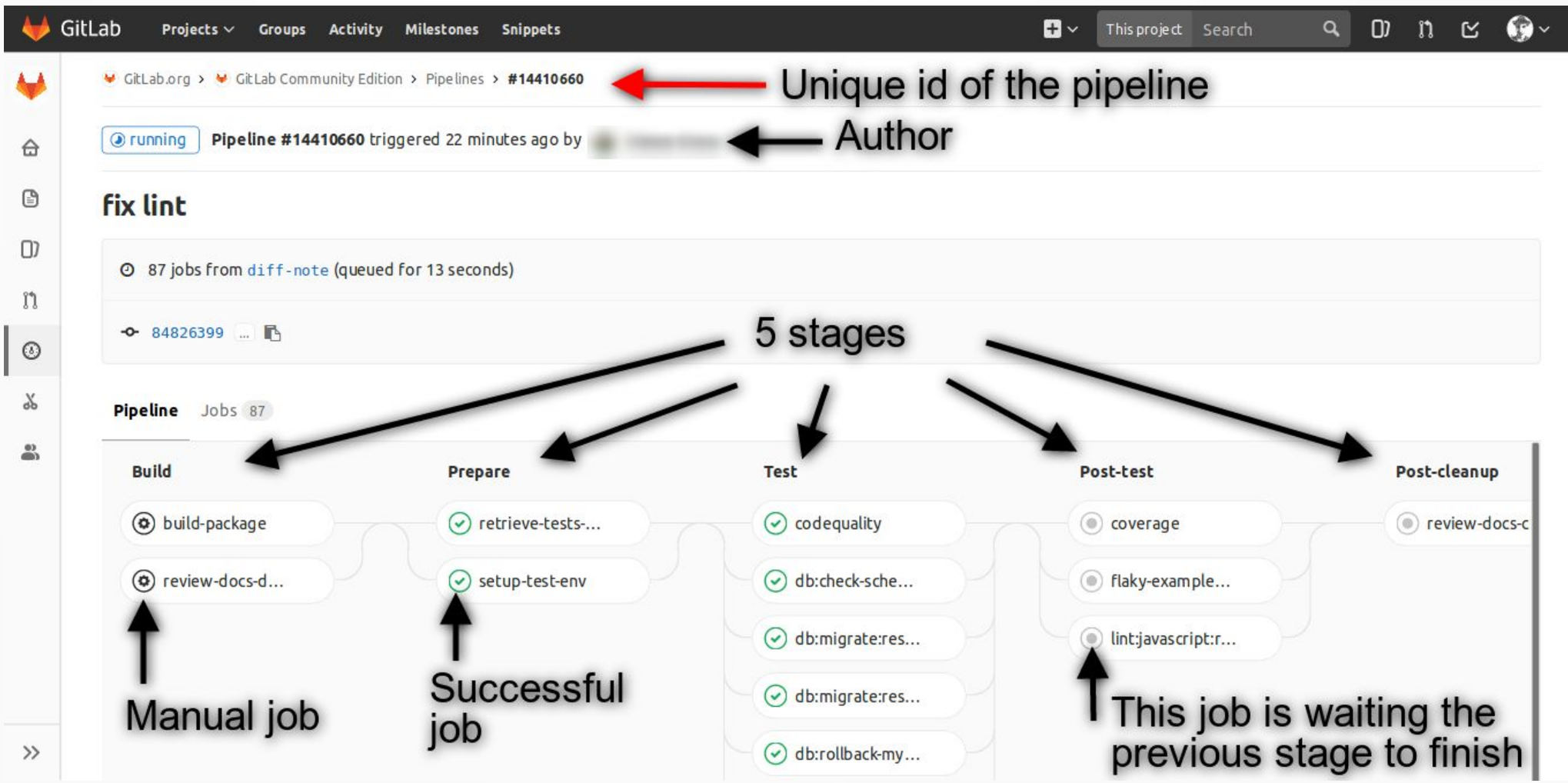


image credit:
https://about.gitlab.com/2018/01/22/a-beginners-guide-to-continuous-integration/

What sort of pipelines

- Based on our discussions so far
  - Or your own experience
  - What sorts of things should CI do for us?

What sort of pipelines

- Based on our discussions so far
  - Or your own experience
  - What sorts of things should CI do for us?
  - Likely answers include
    - Running a linter (or formatter)
    - Compiling (or running code through interpreter)
    - Running automated tests
    - Sending changes to deployment

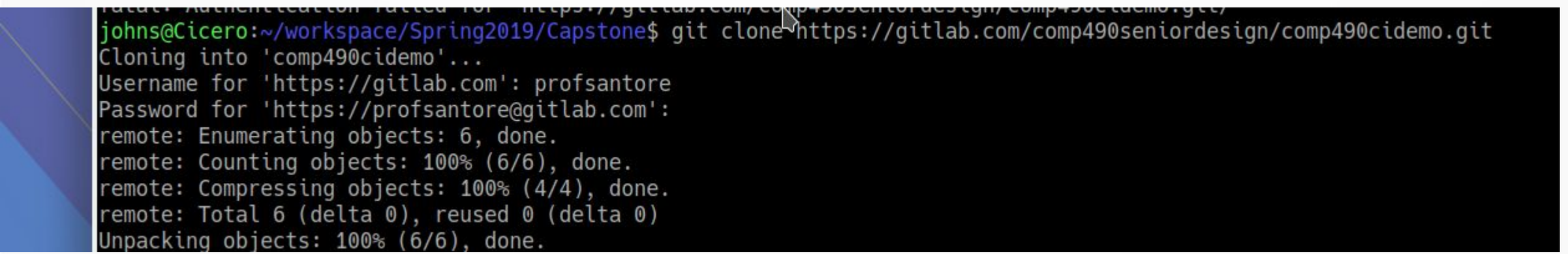
Setting up CI/CD

- Setting up GitLab CI/CD uses a yaml file
- To right is simple example from Zuri Hunter's medium writeup

```
image: node:10.15.0

stages:
  - build
  - test
  - deploy

before_script:
  - npm install

build-min-code:
  stage: build
  script:
    - npm install
    - npm run minifier

run-unit-test:
  stage: test
  script:
    - npm run test

deploy-staging:
  stage: deploy
  script:
    - npm run deploy-stage
```

A Deeper look

- So lets continue with our example
  - I have the floodfill from the makeup project
  - I need to work on that



  - Then I opened it in pycharm
    - Created a test folder and a first test
    - I used pycharm's vcs integration to add it to git and commit
    - Then I pushed the changes to the gitlab project

Like this

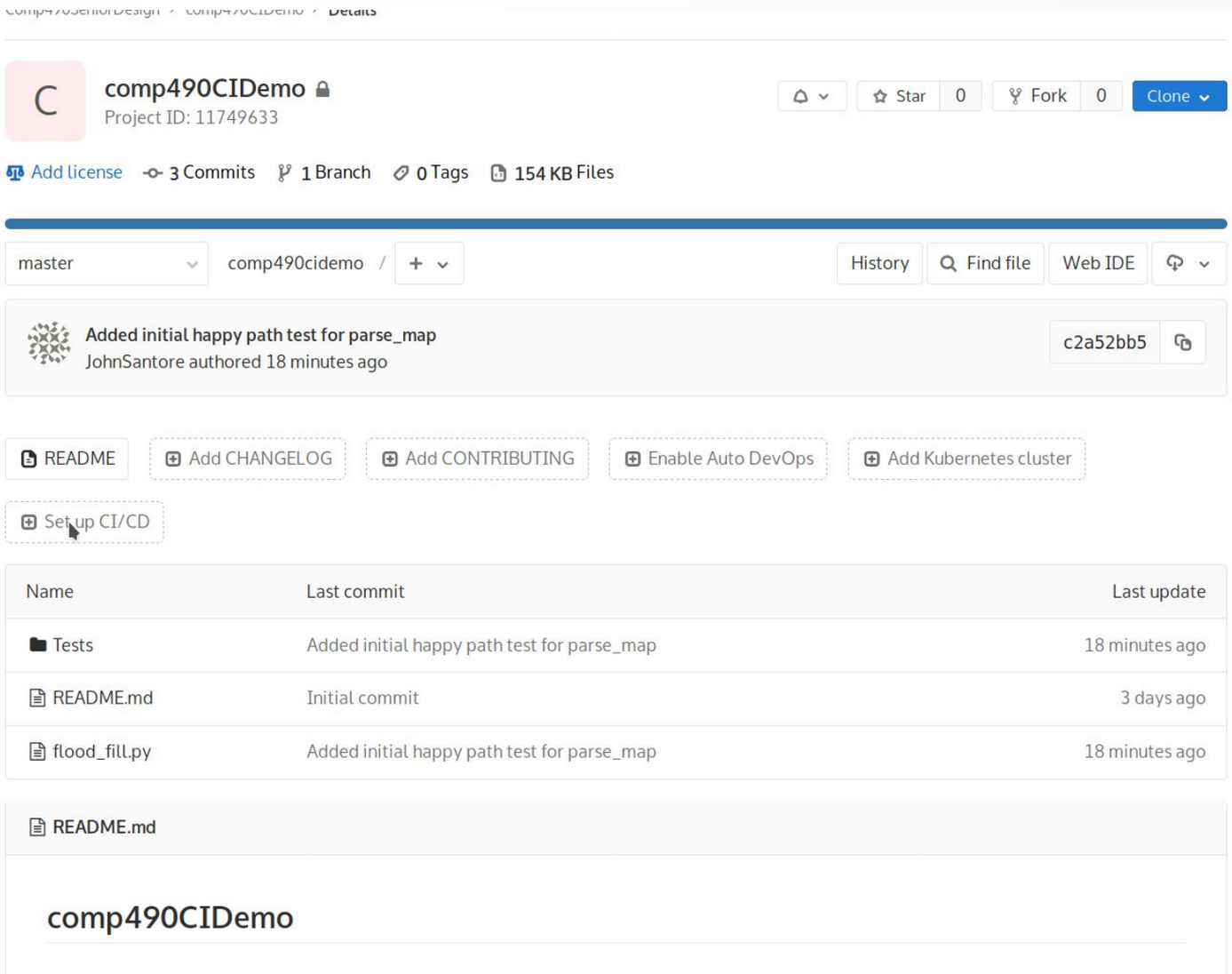
- Here was my update:

```
johns@Cicero:~/workspace/Spring2019/Capstone/comp490cidemo$ git push https://gitlab.com/comp490seniordesign/comp490cidemo.git
Username for 'https://gitlab.com': profsantore
Password for 'https://profsantore@gitlab.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 670 bytes | 670.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
To https://gitlab.com/comp490seniordesign/comp490cidemo.git
   ac18c2f..c2a52bb  master -> master
```

- Caveat, disclaimer, best practices, merge/pull requests, fast and beginner, etc.
- Ok – now we have a test, furthermore, maybe I want to run a linter (flake8 is what I'll use here)

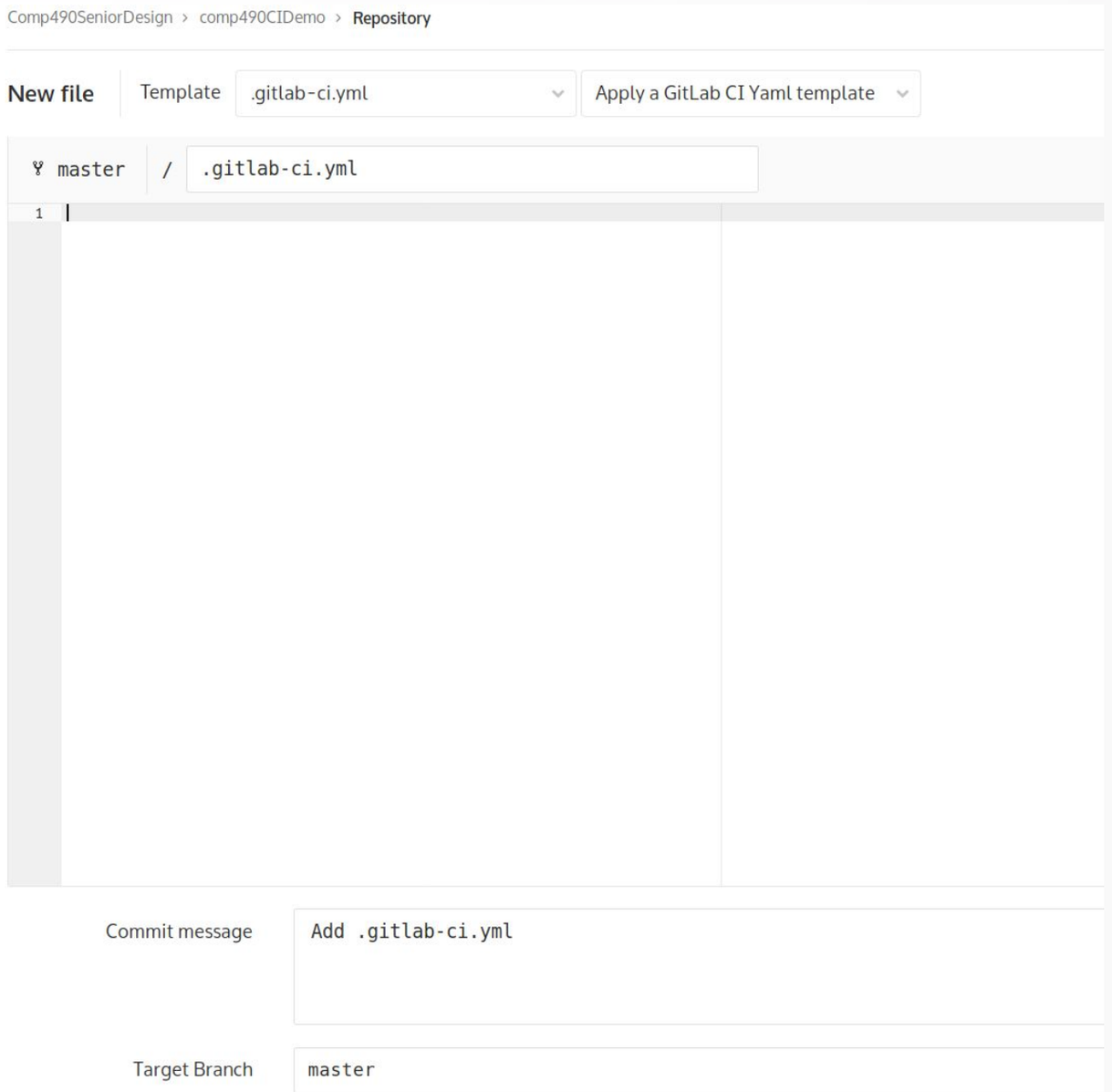
Setup CI on Gitlab

- So lets setup CI directly on Gitlab



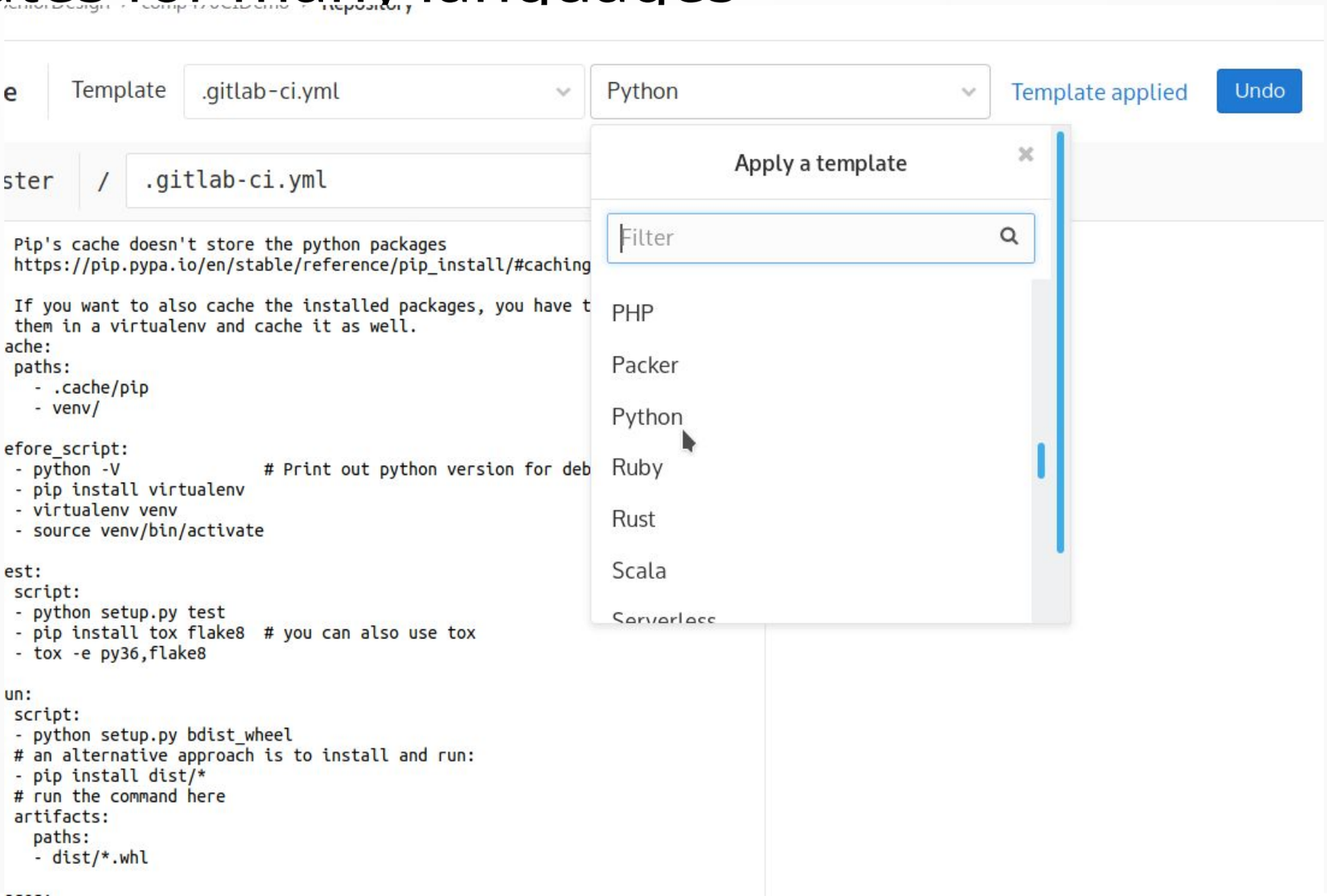
Creates the gitlab-ci.yml

- Using the online ide



Templates

- Gitlab has templates for many languages



Example

- So I used the python template and removed the packaging stuff

```
image: python:latest

# Change pip's cache directory to be inside the project directory
# since we can only cache local items.
variables:
  PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache/pip"

# Pip's cache doesn't store the python packages
# https://pip.pypa.io/en/stable/reference/pip_install/#caching
# If you want to also cache installed packages, you have to
# install them in a virtualenv and cache it as well.
cache:
  paths:
    .cache/pip
    venv/

before_script:
   python -V               # Print out python version for debugging
   pip install virtualenv
   virtualenv venv
   source venv/bin/activate

test:
  script:
   python setup.py test
```

Image

- Image specifies a docker image
  - Anything on docker hub should work (I only tried some python images)
  - https://hub.docker.com/
  -

Before_script

- before_script:
  - This section runs before each job
  - In our case setting up a python virtual env
    - Discuss if needed

Test:

- Test:
  - Section is our only current job
  - The script had more
  - If any item here fails, the whole pipeline fails.
- Can have an arbitrary number of jobs

At this point I went home

- And when I got there
  I had mail

✖ Your pipeline has failed.

Project	Comp490SeniorDesign / comp490CIDemo
Branch	master
Commit	68f1473c Add .gitlab-ci.yml - started from the gitlab py...
Commit Author	JohnSantore

Pipeline #56412394 triggered by JohnSantore
had 1 failed build.

Logs may contain sensitive data. Please consider before forwarding this email.

✖ test — test

```
$ virtualenv venv
Using base prefix '/usr/local'
New python executable in /builds/comp490seniordesign/comp490cidemo/
venv/bin/python
Installing setuptools, pip, wheel...
done.
$ source venv/bin/activate
$ python setup.py test
python: can't open file 'setup.py': [Errno 2] No such file or direc
tory
ERROR: Job failed: exit code 1
```

Simplify

- So I simplified the Ci/CD pipeline

```
# Change pip's cache directory to be inside the project directory
#since we can only cache local items.
variables:
  PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache/pip"

# Pip's cache doesn't store the python packages
# If you want to also cache the installed packages, you have to
#install them in a virtualenv and cache it as well.
cache:
  paths:
    - .cache/pip
    - venv/

before_script:
  - python -V               # Print out python version for debugging
  - pip install virtualenv
  - virtualenv venv
  - source venv/bin/activate

test:
  script:
  #- python setup.py test
  - pip install tox flake8 pytest # you can also use tox
  #- tox -e py36,flake8 # look at this more enterprise way of  later
```

Code not following style

- So my CI fails the build because I'm not following the style guide – again let the computer check it:

```
$ flake8 flood_fill.py
flood_fill.py:1:1: E265 block comment should start with '# '
flood_fill.py:1:80: E501 line too long (109 > 79 characters)
flood_fill.py:3:1: E265 block comment should start with '# '
flood_fill.py:4:1: E265 block comment should start with '# '
flood_fill.py:13:31: E231 missing whitespace after ':'
flood_fill.py:19:28: E261 at least two spaces before inline comment
flood_fill.py:19:29: E262 inline comment should start with '# '
flood_fill.py:20:11: E111 indentation is not a multiple of four
flood_fill.py:25:11: E111 indentation is not a multiple of four
flood_fill.py:77:24: E711 comparison to None should be 'if cond is not None:'
flood_fill.py:106:55: E712 comparison to False should be 'if cond is False:' or 'if not cond:'
flood_fill.py:111:80: E501 line too long (80 > 79 characters)
flood_fill.py:127:51: E712 comparison to Fa
flood_fill.py:131:80: E501 line too long (8
flood_fill.py:131:80: E502 the backslash is
flood_fill.py:132:69: E712 comparison to Fa
flood_fill.py:255:41: E251 unexpected spaces around keyword / parameter equals
flood_fill.py:255:43: E251 unexpected spaces around keyword / parameter equals
flood_fill.py:259:3: E111 indentation is not a multiple of four
flood_fill.py:260:3: E111 indentation is not a multiple of four
flood_fill.py:260:34: E231 missing whitespace after ','
flood_fill.py:261:3: E111 indentation is not a multiple of four
flood_fill.py:262:3: E111 indentation is not a multiple of four
flood_fill.py:266:11: W292 no newline at end of file
ERROR: Job failed: exit code 1
```

Dot dot dot

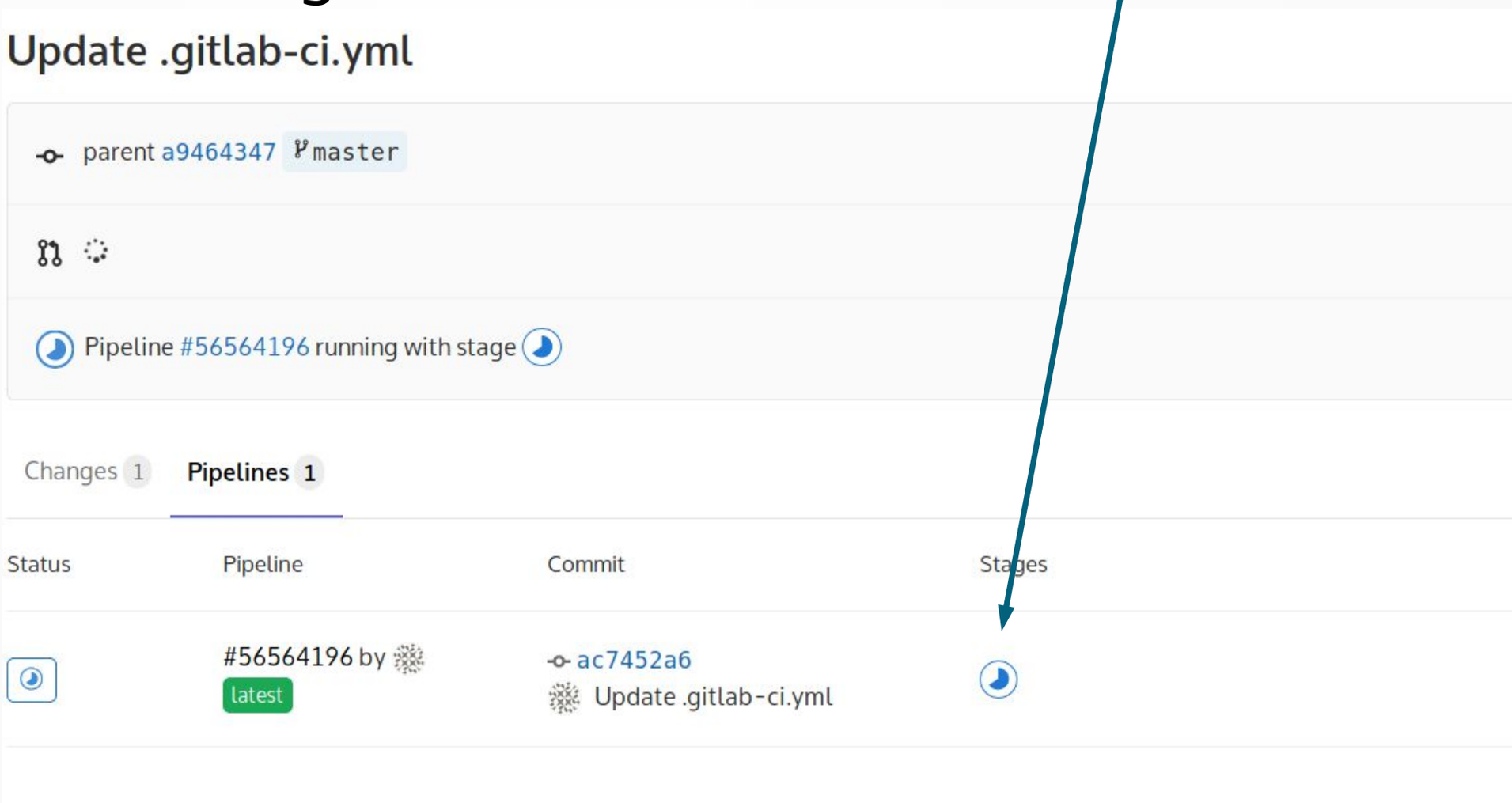
- I did a bit of trial and error learning the ins and outs of these tools on the command line that I've used from pycharm locally
  - I edited .gitlab-ci.yml to run flake8 –max-line-length=100
  - And turned the pytest line to python -m pytest
    - pytest Tests/test_flood_fill.py
    - Became
    - python -m pytest
    - Reason: when run as python -m pytest then current working directory is root of tests (for import statements) also pytest will look in directories named Test for files for the sort test_something.py

Style guides

- So the code I extended wasn't following the style
  - And I introduced two style issues myself
  - Pycharm autofixed about 80% of it
  - I changed all the
    - if expression == False:
    - To
    - if not expression:
  - And fixed my doc string ('→ “)
  - And checked the code back in.

Check in – pipelines running

- Check in – running CI



And Success!

- Finally I don’t have email from gitlab

```
virtualenv-16.4.3
$ flake8 --max-line-length=100 flood_fill.py
$ python -m pytest
============================= test session starts ==============================
platform linux -- Python 3.7.3, pytest-4.4.0, py-1.8.0, pluggy-0.9.0
rootdir: /builds/comp490seniordesign/comp490cidemo
collected 1 item

Tests/test_flood_fill.py .                                               [100%]

=========================== 1 passed in 0.03 seconds ===========================
Creating cache default...
.cache/pip: found 193 matching files
venv/: found 1737 matching files
Uploading cache.zip to https://storage.googleapis.com/gitlab-com-runners-cache/project/11749633/default
Created cache
Job succeeded
```

Any questions

- Anyone not quite there on your version of the project?
- Please help your neighbors and I’ll be around

Now lets do real work

- We have our project in the basics of shape
  - But we only have one happy path test.
  - Lets do a more complete test
  - Uncomment the commented test in the test_flood_fill file. It tries to load a non-existent file. That should do something sane, like return an empty list – not throw exception
  - Commit to gitlab and the tests will automatically run and show the error

The failed test

As those of you who did the makeup project discovered, the parse _map function isn't very robust

```
Tests/test_flood_fill.py:20:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

file_name = 'lkj nc qgfwcaelkjh.txt'

    def parse_map(file_name: str):
        """
        this function might have problems, but testing will find that.
        """
>       map_file = open(file_name, encoding='UTF-8')  # the encoding-'UTF-8' is only needed for windows
E       FileNotFoundError: [Errno 2] No such file or directory: 'lkj nc qgfwcaelkjh.txt'

flood_fill.py:242: FileNotFoundError
============================================= 1 failed in 0.03 seconds ==========================================
```

TDD after all

- Now let’s do us some test driven development
  - We have a failing test, how do we make it pass?

How should we fix the issue

- ? students suggest and we try it

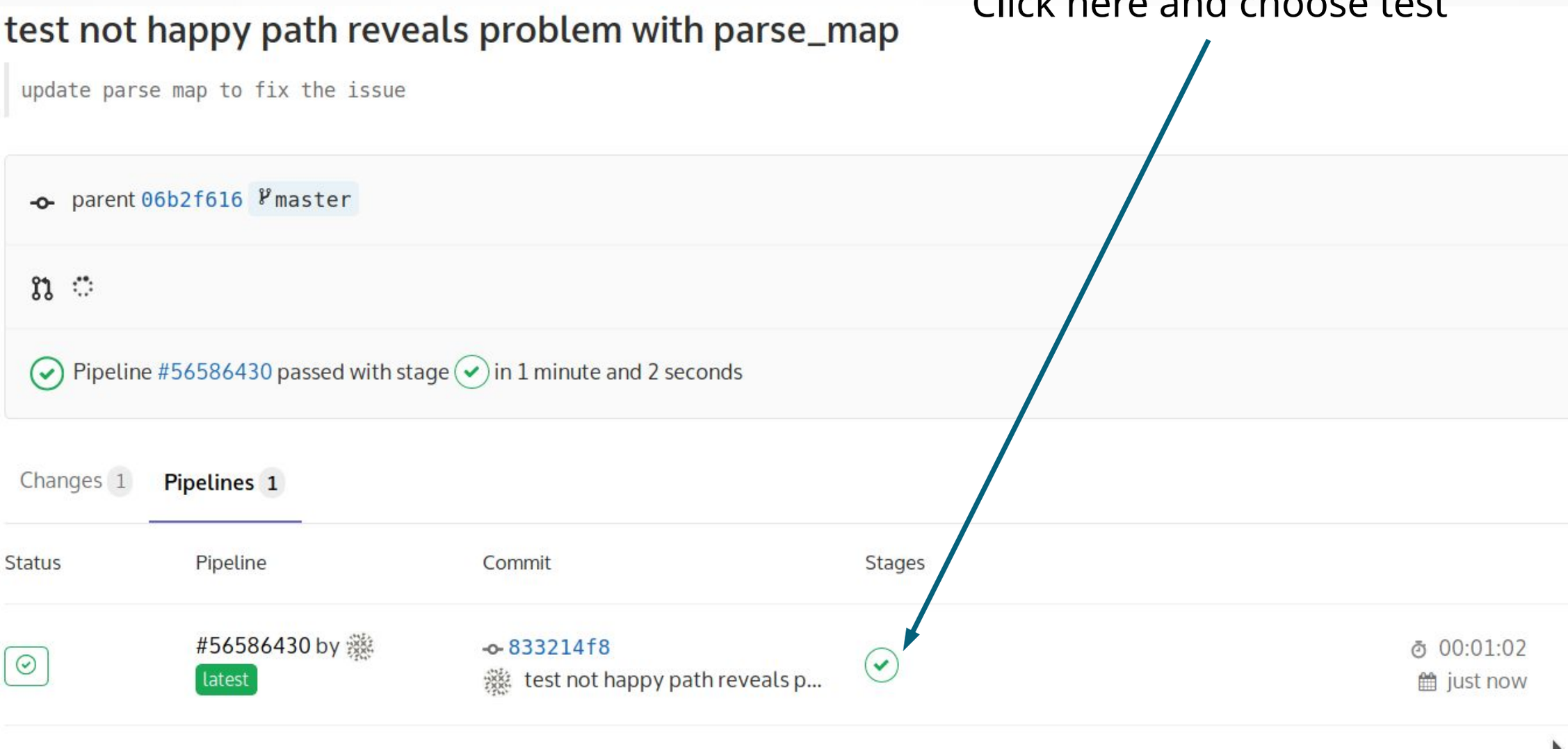
My solution

- My solution below, passes tests

```
def parse_map(file_name: str):
    """
    this function might have problems, but testing will find that.
    """
    map_representation = []
    try:
        map_file = open(file_name, encoding='UTF-8')  # the encoding-'UTF-8'
needed for windows
        lines = map_file.readlines()
    except (FileNotFoundError, PermissionError):
        return map_representation
    for line in lines:
        line = list(map(int, line.split(',')))
        map_representation.append(line)
    return map_representation
```

And commit works

- After the commit:



Yes – we fixed the error!

- Now our pipeline is working like it should
-

```
pytest) (7.0.0)
Requirement already satisfied: attrs>=17.4.0 in ./venv/lib/python3.7/site-packages (from pytest) (19.1.0)
$ flake8 --max-line-length=100 flood_fill.py
$ python -m pytest
============================= test session starts ==============================
platform linux -- Python 3.7.3, pytest-4.4.0, py-1.8.0, pluggy-0.9.0
rootdir: /builds/comp490seniordesign/comp490cidemo
collected 1 item

Tests/test_flood_fill.py .                                               [100%]

========================== 1 passed in 0.03 seconds ===========================
Creating cache default...
.cache/pip: found 193 matching files
venv/: found 1737 matching files
Uploading cache.zip to https://storage.googleapis.com/gitlab-com-runners-cache/project/11749633/default
Created cache
Job succeeded
```

Now lets continue

- I wrote that function to have an issue for the makeup assignment
- Lets go from here. Lets choose a function, write a test for it, commit and git push and let our CI pipeline run the tests automatically
- If the test fails we'll edit the real code
- Huzzah!

References

- Some references I used if you want/need more
  - https://medium.com/devopslinks/beginner-friendly-introduction-to-gitlab-ci-cd-1c80ee5ba0ae
  - https://realpython.com/python-continuous-integration/
  - https://medium.com/metro-platform/continuous-integration-for-python-3-in-gitlab-e1b4446be76b
  - https://docs.gitlab.com/ee/ci/yaml/#stages
  - https://www.tutorialspoint.com/gitlab/gitlab_ci_cd.htm
  - http://www.codingtricks.biz/ci-cd-python-project-with-gitlab/