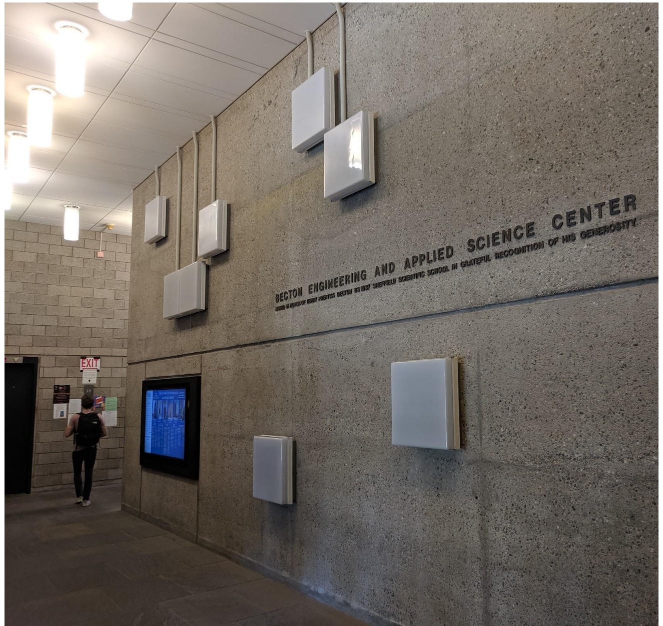
# Creative Embedded Devices: Class 3

Getting started:
Pi on your Pi

## Today:

- Homework questions
- Processing Demo
- Dev vs Prod
- Activity

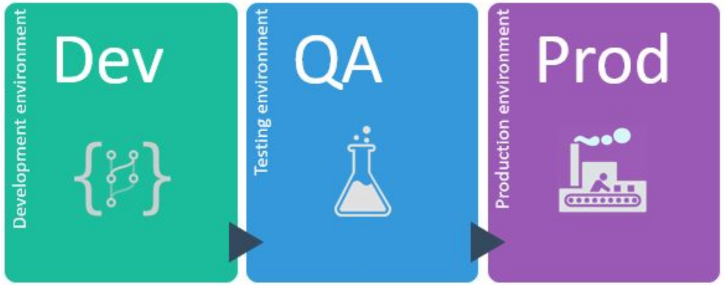
YALE CENTER FOR ENGINEERING
INNOVATION AND DESIGN
TOBACCO FREE

EXIT
EXIT
WINCHESTER HALL

BECTON ENGINEERING AND APPLIED SCIENCE CENTER
EXIT

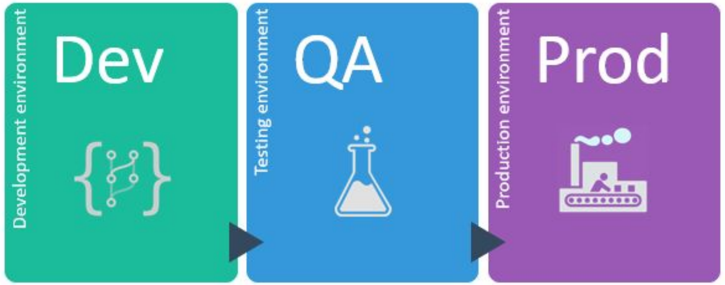
## Processing Demo

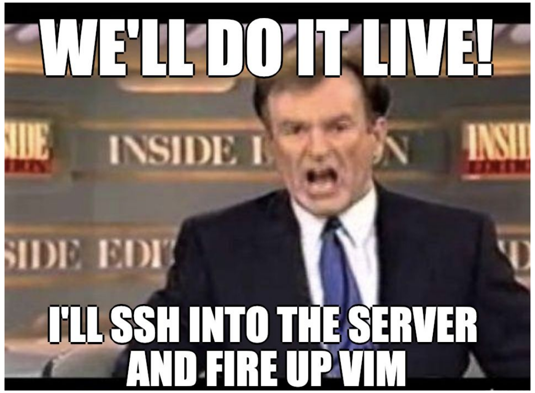
- Homework questions
- Processing Demo
- Dev vs Prod
- Activity

# Dev(lopment) vs Prod(uction)



# Dev(lopment) vs Prod(uction)





# Dev(lopment) vs Prod(uction)

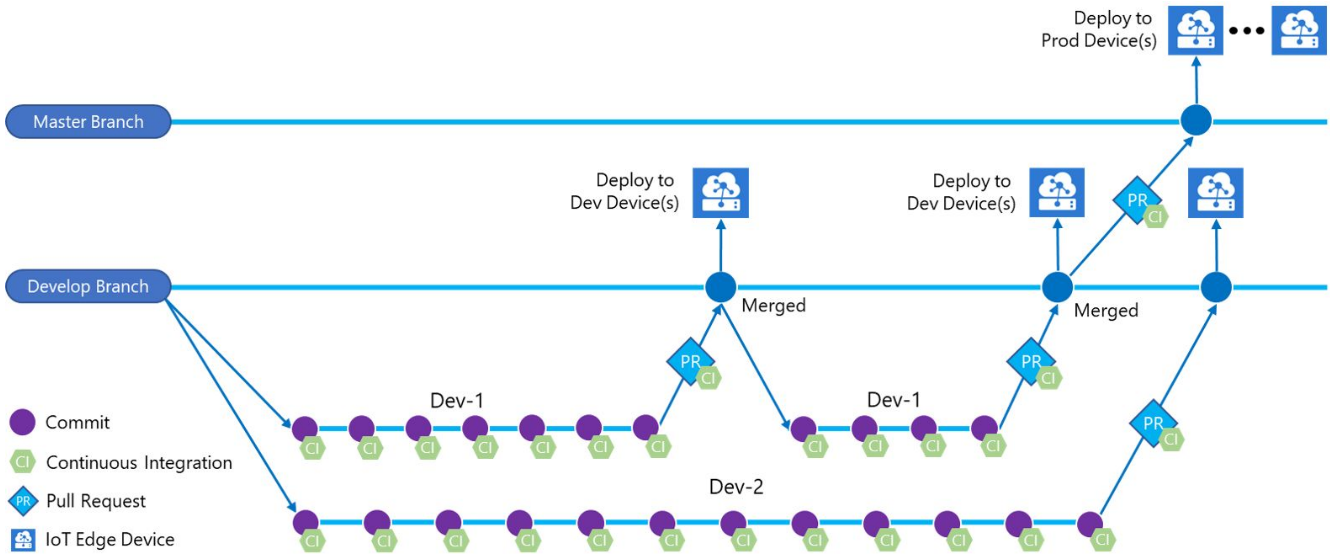
What does this mean for creative embedded systems?

- Automating deployment workflows ensures reproducibility

- Production usually doesn’t change once you “install” (eg a museum*) your system
- However, you might rely on remote service that can change (e.g. pulling weather from an API)

*bonus points for discussing the artistic implications of a museum work that can be remotely updated. Does the space of a museum exhibit allow for ‘software updates’?

# Dev vs Prod


Deploy to Prod Device(s)
Master Branch
Deploy to Dev Device(s)
Deploy to Dev Device(s)
PR
CI
Develop Branch
Merged
Merged
PR
CI
PR
CI
PR
CI
Dev-1
Dev-1
Dev-2
Commit
Continuous Integration
Pull Request
IoT Edge Device

# Dev vs Prod

What does this mean for creative embedded systems?

- Automating deployment workflows ensures reproducibility

- Production usually doesn’t change once you “install” your system
- However, you might rely on remote service that can change (e.g. pulling weather from an API)

- General principle of “*Keep Dev close to Prod*”
- Production environment may have different hardware (processing, sensors, actuators) than dev. May need to emulate this hardware.

## Dev Environments

Developing on the physical embedded device is not always practical

We need to be able to develop software on one machine and port to another.

- What is an architecture (x86 vs ARM)?

  > “The x86 was not designed; It evolved on an island, with a strange bird that ate everthing that tried to pray on it. It now looks stranger than a duck billed platypus, and would not do well if a ship-full of new animals came along.– ctrl-alt-delor

- Cross-compilation vs side-loading vs network connections

read more on http://cpsc334.cs.yale.edu/1_Background/1_Preliminaries/1_Development_Tools.html

# Activity - Pair Program your Pi for Pi

You have three tasks to complete in any order:

- Connect a screen to your Pi - you do not need a SD card running to start on this.
- Write a program to calculate the 10,000 digit of Pi on your laptop, and move that program to your Pi over the internet. Collect timing information for both machines using linux's `time` command.
- Use `/etc/rc.local`* to run your Pi program on startup and save the result in a timestamped file (just in case it changes one day). You will need to use `vi`, `emacs`, or `nano`

* `systemd` is another option, but is less portable across systems