

# Using Feature Boxes For Faster Development

Mike Waud and Eric Palakovich Carr



# Who are we?



Mike



Eric

Software Developers at the Schroeder Institute at Legacy

# One of Our Apps - UbiQUITous



First ever evidence based quit smoking Facebook app  
~8000 users

Built With

- Django
- Celery
- GitHub
- Amazon RDS
- Heroku
- Heroku Addons

# How a Site Gets Made

Step 1 - Develop Code

Step 2 - ???

Step 3 - Deploy Code to Production Server

The ??? is important

# How We All Started





# After the Site Crashed a Few Times



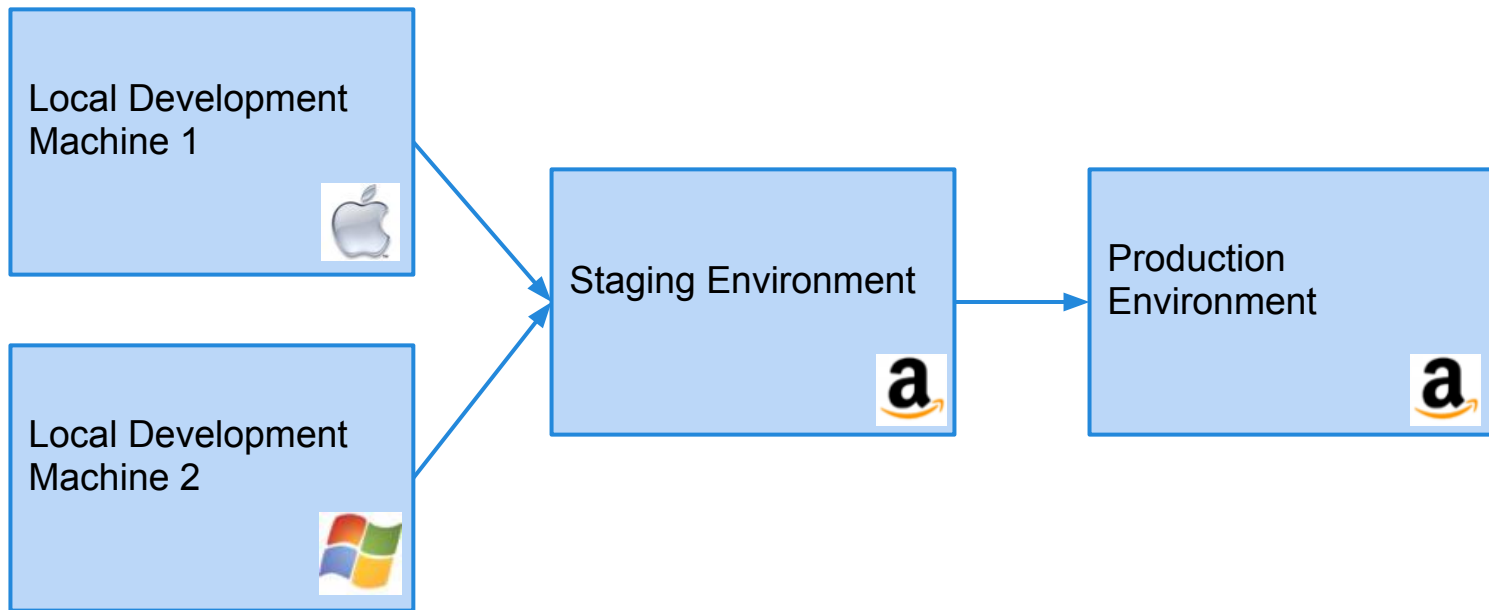
**I DON'T ALWAYS PUT MY  
CODE ON STAGING**

**BUT WHEN I DO, IT  
LIKES TO BREAK WHEN  
MY BOSS LOOKS AT IT**

quickmeme.com



# With 2+ Developers



**I DON'T ALWAYS PUT MY  
CODE ON STAGING**

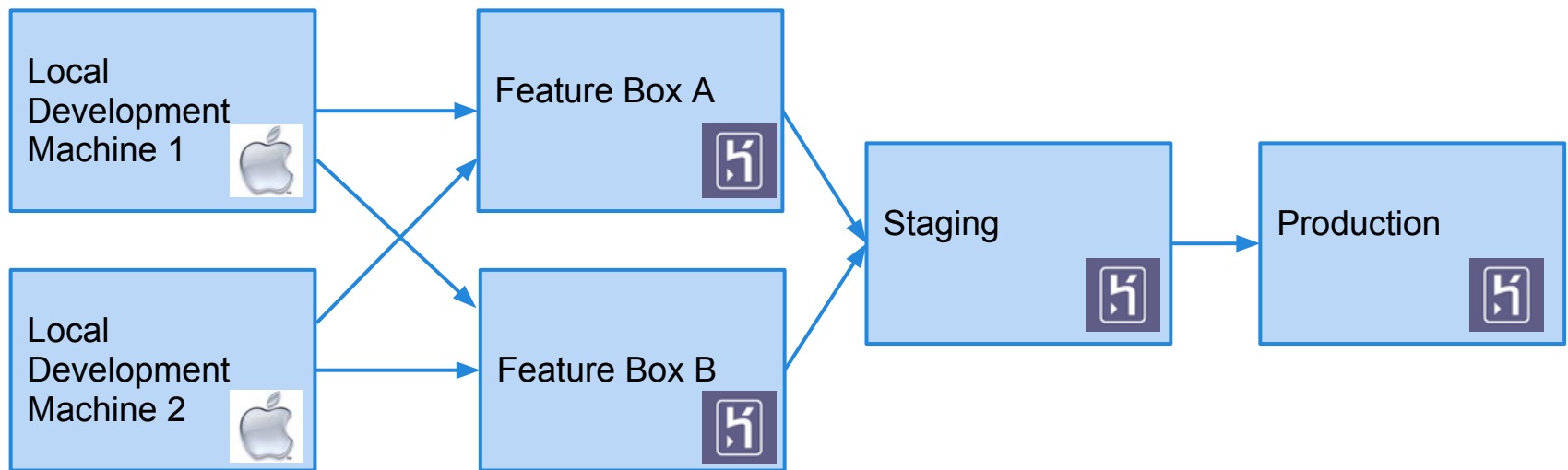
**BUT WHEN I DO, I  
MESS UP SOMEONE  
ELSE'S TESTING**

# Premise

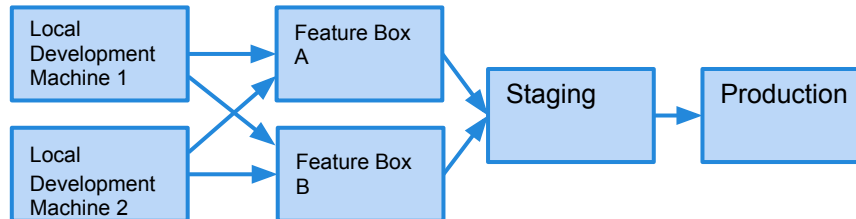
- testing should not be difficult
- upcoming features should be viewable by others at any time (QA, clients, etc.)
- developers should not compete for resources
- the testing environment should mirror production environment as closely as possible

# What We Recommend

## "Feature Boxes"



# Pros & Cons



## Pros

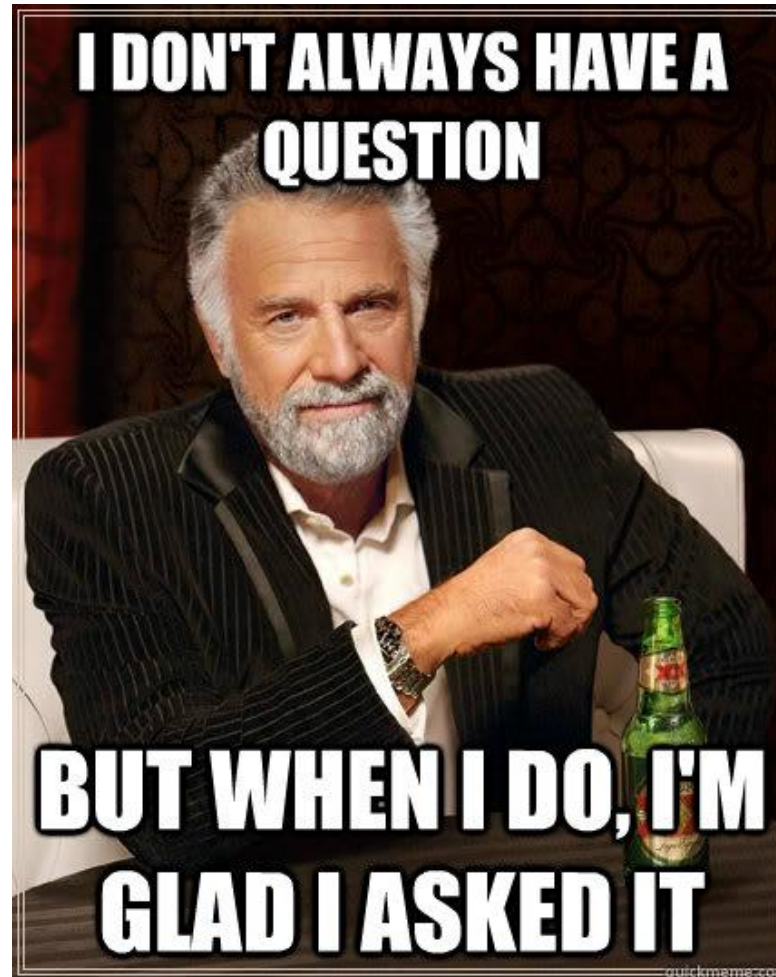
- encourages testing of features in isolation
- always available and accessible by 3rd parties
- can more closely match production environment
- good for teams that share a codebase

## Cons

- cost
- some administration required, but this can be minimized
- overkill for small or fast moving projects

# Tips, Tricks, and Gotchas

- name your feature boxes in a way that is memorable, but doesn't imply ownership
- environment variables are your friend
- add the git branch and commit to the environment variables (automated with Fabric)
- watch out for database schema conflicts when running migrations



# Questions?

We are hiring!

[mwaud@legacyforhealth.org](mailto:mwaud@legacyforhealth.org)  
[epalakovichcarr@legacyforhealth.org](mailto:epalakovichcarr@legacyforhealth.org)