

# Continuous Deployment with iOS

Patrick Goley

# What is Continuous Integration / Deployment?

- Continuous Integration - automated process of testing code and integrating it into the product
- Continuous Deployment - the ability to put new builds in the hands of users or tests with minimal or no effort
- Automating all possible tasks in the development process



# Why Invest in Automation?

- Consistency - Enforce a process for building, testing, and deployment
- Quality - Catch bugs before they leave the developer
- Speed of delivery - New code is delivered *as soon as it's working*, not when the developer manages to distribute it.

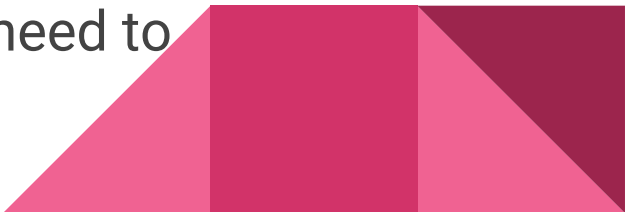


“Asking experts to do boring and repetitive, and yet technically demanding tasks is the most certain way of ensuring human error... short of sleep deprivation, or inebriation.”

– David Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*



# Consistency

- Produce builds in a predictable and reliable way
    - Always increment build number
    - Always use correct code signing
    - Always tag commit
  - Notify testers when new builds are ready
  - Notify the responsible developer when a build fails
  - Everyone's builds follow the same policy, no need to educate or document
- 

# Quality

- Don't waste testers' time catching bugs that could have been caught with tests
- Value of automated testing is only realized when the tests are run
- Enforce a certain level of code coverage



# Speed of Delivery

- Builds should be available to testers as soon as they pass continuous integration with no effort from the developer
- Builds can be pushed to the store with the click of a button
- The job is only done when features are in the hands of users



# Implementation

Build scripts - logic for how to build, test, and distribute your application

Build server - responds to code changes and executes build scripts and publishes the results






# Tools

- xcodebuild - build and archive. more verbose, pipe to xcpretty
- xctool - can run tests in parallel
- shenzhen - can build and distribute to various services

Piecing them together...

- bash
  - make
  - Fastlane!
- 



Fastlane is a robust automation toolset for Cocoa development.

Built with Ruby, it comes with a host of gems for automating all kinds of tasks like building, distributing, even managing your developer portal and iTunes Connect.

<https://krausefx.com>

# Just a few fastlane tools...

Gym - build and archive applications

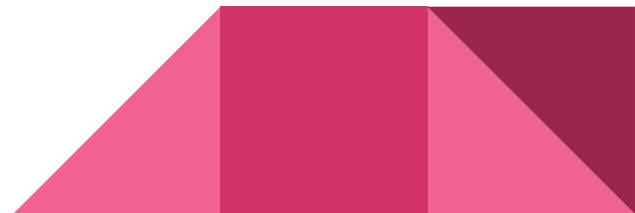
Scan - execute tests, generate reports

Produce - create applications on iTunes Connect and the Developer Portal

Match - certificate and provisioning profile management (great for teams)

Snapshot - generate localized screenshots for every supported device

145 built-in actions at time of writing



# Examples

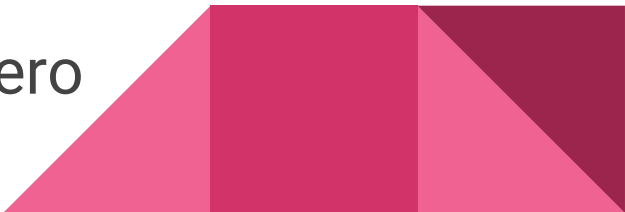
<https://github.com/patgoley/ToDo>

<https://github.com/fastlane/examples>



# Build Server

Server application to manage different jobs that execute tests or produce builds.

- Jobs typically clone a repository and then run scripts in the directory
  - Cache artifacts (builds or test results)
  - Usually triggered by git webhooks
  - Job fails if any command returns non-zero
- 

# Drawbacks

- Managing build scripts and server
- Updating existing projects to work with scripts
- Time invested only pays off after a few builds

