

# Welcome to A100

## *Fall 2015 Cohort*



# A100 History

*How did we get here?*





Derek Koch

Julio Mansilla

Krishna Sampath

Chris Impastato

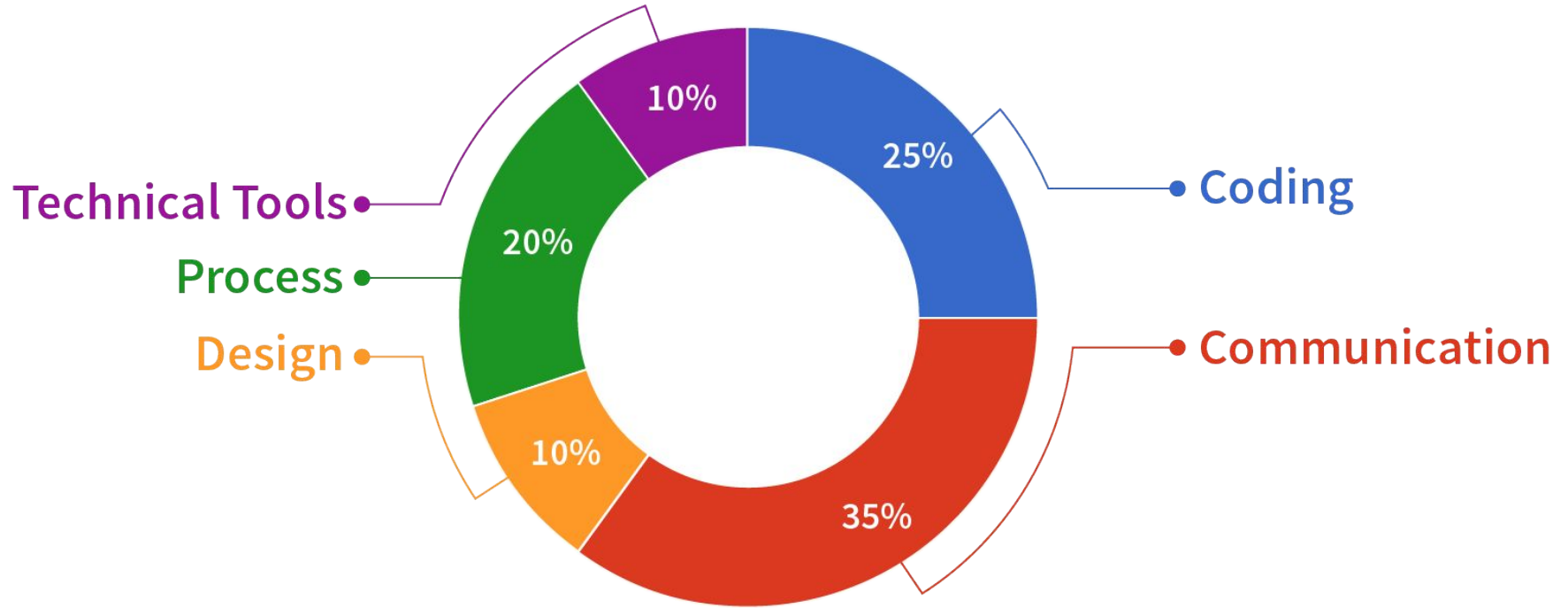
Senior Apprentices

## Introductions: Linked List!

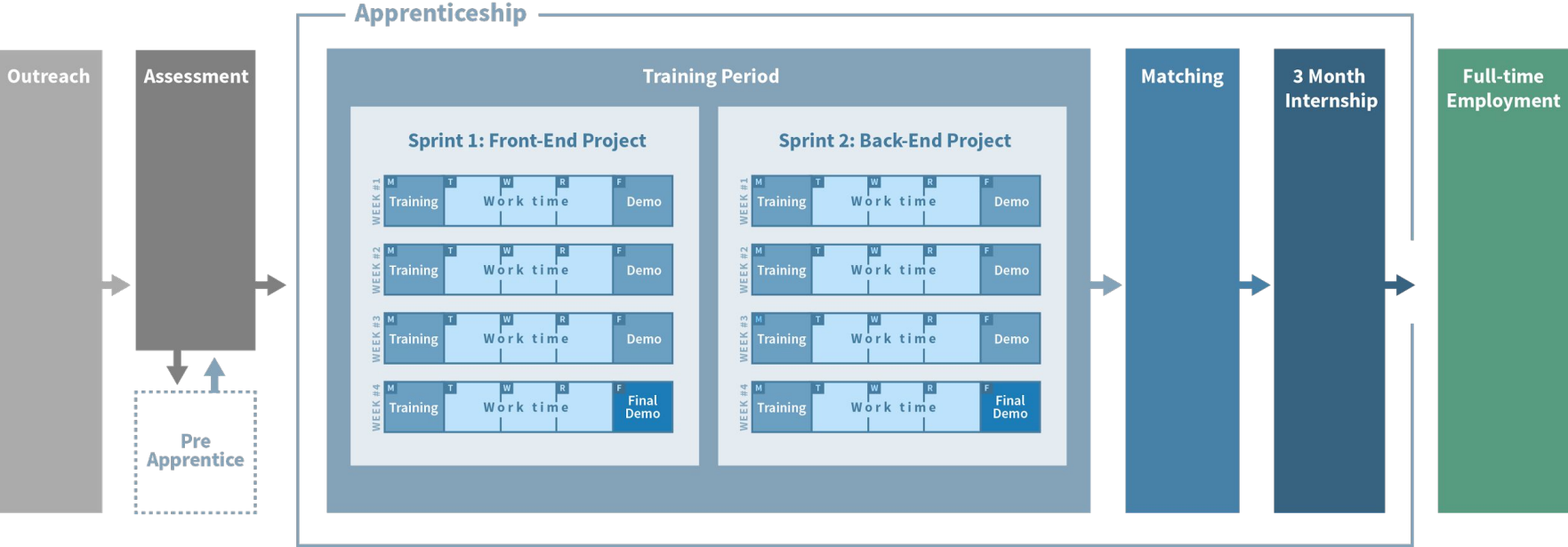
- [Thank previous person]
- Your Name
- Your Major & School (or your background)
- One Thing You'd Love to Learn
- Point to next person

**Your Goals?**





# Apprentice100 Program Timeline





## Program Expectations

Show up for all sessions on time and prepared.

Ask questions. Questions are mandatory.

Respond to all emails from A100 Staff, your team, Partner Companies within one day.

Apply to and Interview with Partner Companies exclusively.

*CC Site Manager on ALL communications with Partner Companies.*

### 3-Strike Rule.

Strikeable Offenses:

Breaking one of **The 5 Pillars of A100**:

1. Be present and prepared at all sessions.
2. Pull your own weight on the Taskboard.
3. Contribute quality code to your team via git.
4. Address all issues affecting your team's performance on a timely and appropriate manner.
5. Ensure your MVS performance is on track.

Dismissal can happen at any point during the Program.

**Grounds for immediate dismissal:**

Any inappropriate behavior or violation of our Code of Conduct.

## Program Schedule

- Training Sessions:
  - Hartford: Mon, 5:00 - 7:30 pm
  - New Haven: Tue, 5:00 - 7:30 pm
  - Make-up sessions (Hartford): Wed, 10:00 am - 12:30 pm
- StudyHall: A Weekly Hack Night
  - Hartford: Wednesday, 5:30 - 8:30 pm
  - New Haven: Thursday, 5:30 - 8:30 pm
  - MUST RSVP here: <http://meetup.com/a100-dev-community>
- Demo Day:
  - Hartford: Fri, 10:00 am - 12:00 pm
  - New Haven: Fri, 12:00 - 2:00 pm
  - Arrive early for tech check and set up

***Bio Break (5 minutes)***

# Introduction to Agile



**What is a *Moment of Truth*?**

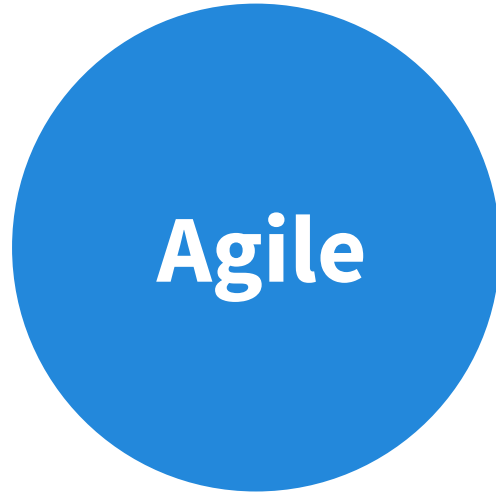


***Communication***

---

***Code***

Problem



Solution

# Exercise

## *Moving Day*

## Ground Rules

- Each team acts on its secret instructions
- You cannot speak to each other during this exercise
- You may not share your goal with the other teams

**Feedback**



A sustainable pace of development & environment of learning.

**Individuals and Interactions** ...over processes and tools

**Working Software** ...over documentation

**Collaboration** ...over negotiation

**Responding to Change** ...over following a plan

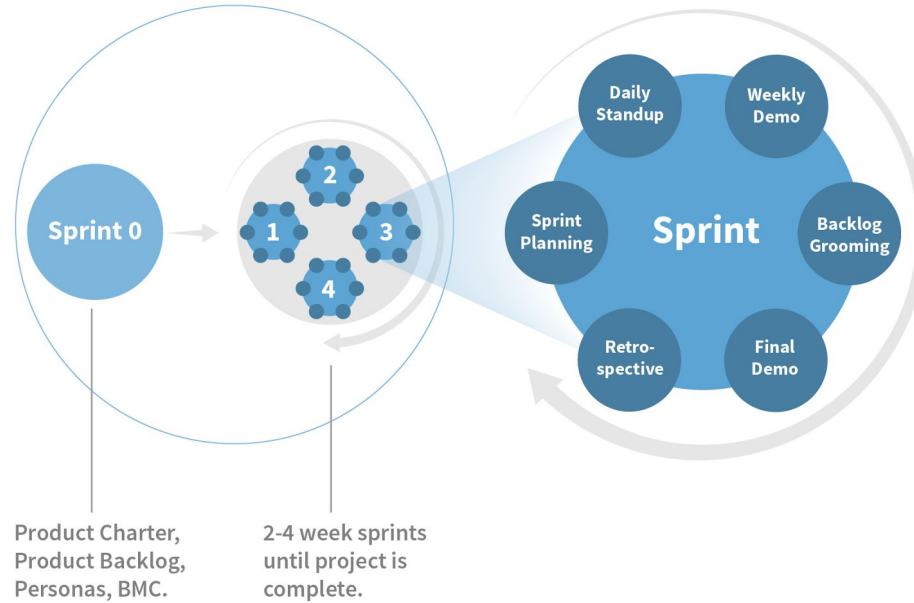


# Agile Process for New Product Development

Roles	Product Owner Scrum Master Team Member
Ceremonies	Product Planning Meeting Sprint Planning Meeting Daily Standup Weekly Demo Final Demo Retrospective
Tools	Project Charter Business Model Canvas Persona Backlogs Planning Poker Taskboard Template Agendas
Concepts	Releases Sprint Stories

# Agile Process for New Product Development

## Project Cycle (Product Planning)





**Why Bother?**

## Why Bother?

Can't the **Product Owner** just tell me exactly what they want?

Can't I just **build** what I know will satisfy the need?

We can pick up the pace and **cram in more work** at the end, right?

## Triple Constraint



**Space**

# Version Control I: Intro to Git





## What is Version Control?

### **Have you ever ...**

saved a file named *essay\_version2b\_May42015(edits-10AM).doc*?

lost files JUST when you needed them?

forgotten what changes you made?

needed to collaborate on a coding project?

broke a program JUST before a deadline.

**Version Control is the answer.**



**git**

## Terminology

**Git** A version control system developed by Linus Torvalds.  
Initially used during development of Linux.

**repo** Short for *repository*.

Directory in which files are stored and tracked as part of a project.

**GitHub** A popular online host for git repositories.  
NOT the same as git.

**Command line** A text-based interface for systems/programs.  
On Windows: *PowerShell*, *git Bash*. On OSX/Linux: *Terminal*.

## Git Commands: New Repositories

`git`                      Start every command with this

`git init`                Initialize a new repo in current directory

`git clone`              Create a clone of an existing repo

`git status`            Show current status of the repo

## Git Commands: Staging and Committing

`git add [file]` Add [file] to the staging area

`git add .` Add all files (.) in the current directory

`git commit -m "[Message]"`

Commit all tracked files/changes, including a [message]

`git log` Show commit history

## Git Commands: *reset* or *rm*?

`git reset [file]`      Unstage [file]

`git reset --soft`      Unstage everything

`git reset --hard`      Roll back to previous commit,  
losing all changes

`git rm [file]`      Delete [file] and tells git about it



Specify a location for remote repo on [server]:

```
git remote add origin [server]
```

Usually, this [server] will be something like:

```
https://github.com/user/repo.git
```

Check your remote's location:

```
git remote -v
```

Get changes from a remote repo called `origin`:

```
git pull origin master
```

Push your (committed) changes to a remote repo, `origin`:

```
git push origin master
```

### **Official Documentation**

### **Cheat Sheets**

[Git cheat sheet](#)

### **Free Ebooks**

[Learn Version Control with Git](#)

[Command Line 101](#)

[Pro Git](#)

# Individual Assignment 1 (a & b)



### **Individual Assignment 1a**

Setup a repo for your Personal Website.

### **Individual Assignment 1b**

Begin your first Individual Project - Personal Website.

Create a page and add your current resume in some form.

**Be ready to demo your work at this week's Demo Day.**

# See you at StudyHall!

