

Agile + DevOps **EAST**

A TECHWELL EVENT

DW9

DevOps Case Study

Wednesday, November 7th, 2018 2:45 PM

Climbing the Mountain of Continuous Deployment

Presented by:

Michele Campbell

Lucid Software Inc

Brought to you by:



350 Corporate Way, Suite 400, Orange Park, FL 32073
888-268-8770 · 904-278-0524 - info@techwell.com - <http://www.starwest.techwell.com/>

Michele Campbell

Michele is a quality assurance manager and release coordinator at Lucid Software Inc., where she has been working on improving the testing process for over three years. Lucid has two web-based platforms that help people work visually, called Lucidchart and Lucidpress. She has been the lead software tester on both products and has performed mobile and API testing roles as well. Lately, her focus has been on improving the production release process and training new team members on common testing methodologies. In her free time, she enjoys board games, playing with her guinea pig, and trying out new recipes.



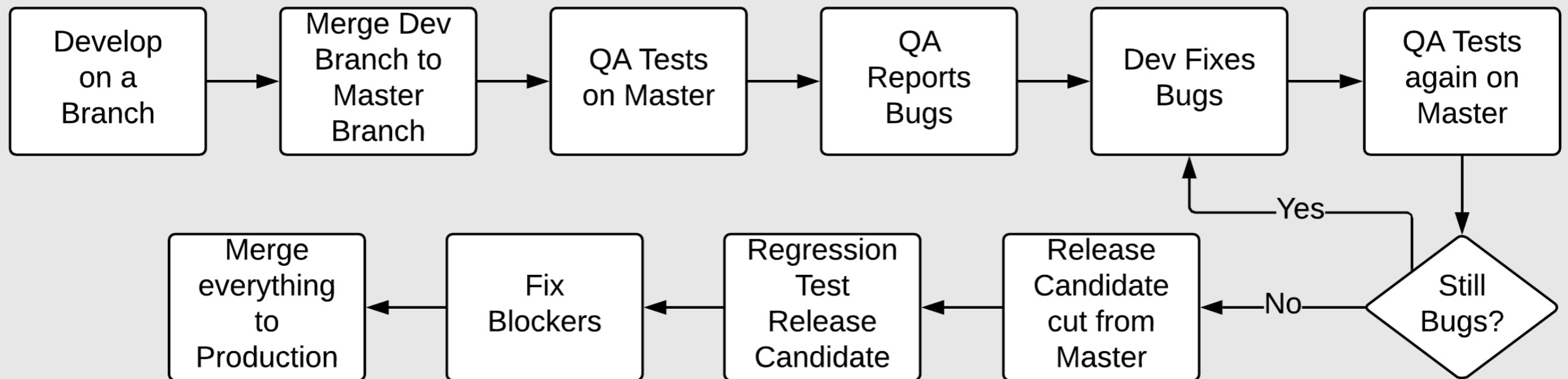
CLIMBING THE MOUNTAIN OF CI/CD: A TESTER'S TALE

MICHELE CAMPBELL

Where We Were

	Monday	Tuesday	Wednesday	Thursday	Friday
Week One	<p>Cut Release Candidate at 9AM</p> <p>Regression Testing</p> <p>Fix Blockers</p>	<p>Regression Testing</p> <p>Fix Blockers</p> <p>Develop on Branches</p>	<p>Regression Testing</p> <p>Fix Blockers</p> <p>Develop on Branches</p>	<p>Release EVERY SINGLE THING at 2PM</p> <p>Develop on Branches</p>	<p>Functional Testing</p> <p>Develop on Branches</p>
Week Two	<p>Functional Testing</p> <p>Develop on Branches</p>	<p>Functional Testing</p> <p>Fix Bugs</p> <p>Develop on Branches</p>	<p>Functional Testing</p> <p>Fix Bugs</p> <p>Develop on Branches</p>	<p>Functional Testing</p> <p>Fix Bugs</p> <p>Develop on Branches</p>	<p>Functional Testing</p> <p>Fix Bugs</p> <p>Develop on Branches</p>

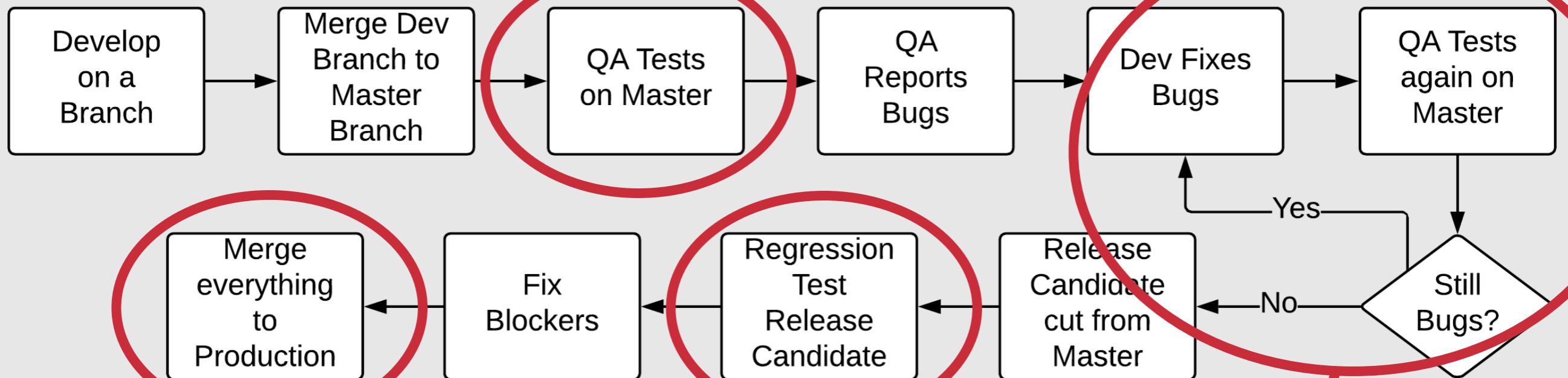
Where we were



Where we were

We can still be in this cycle when we cut RC branch

No testing before this meant problems down the line



Over 70 developers worth of work for 2 weeks

It took 7 testers 3 full days to do. Way too long!

Waiting for things to build took ages. This would be a really long feedback cycle

The Goal

- ▶ We have been doing scheduled production deploys every other week since I started four years ago
- ▶ In the last four years, we have grown from a ~50 person company to ~450 people
- ▶ **We want to get to continuous deployment**



How to Achieve it

We needed to consider the following:

- ▶ The People
- ▶ The Obstacles
- ▶ The Timeline
- ▶ The Tools

The People



The Head of Engineering



The Engineers



Quality Assurance



The Release Coordinator/Manager



Devops



Product Managers

The Obstacles



Arduous Deploys



The Obstacles

Habits



The Obstacles



Regression Testing



The Timeline

- Specific
- Measurable
- Achievable
- Relevant
- Time bound

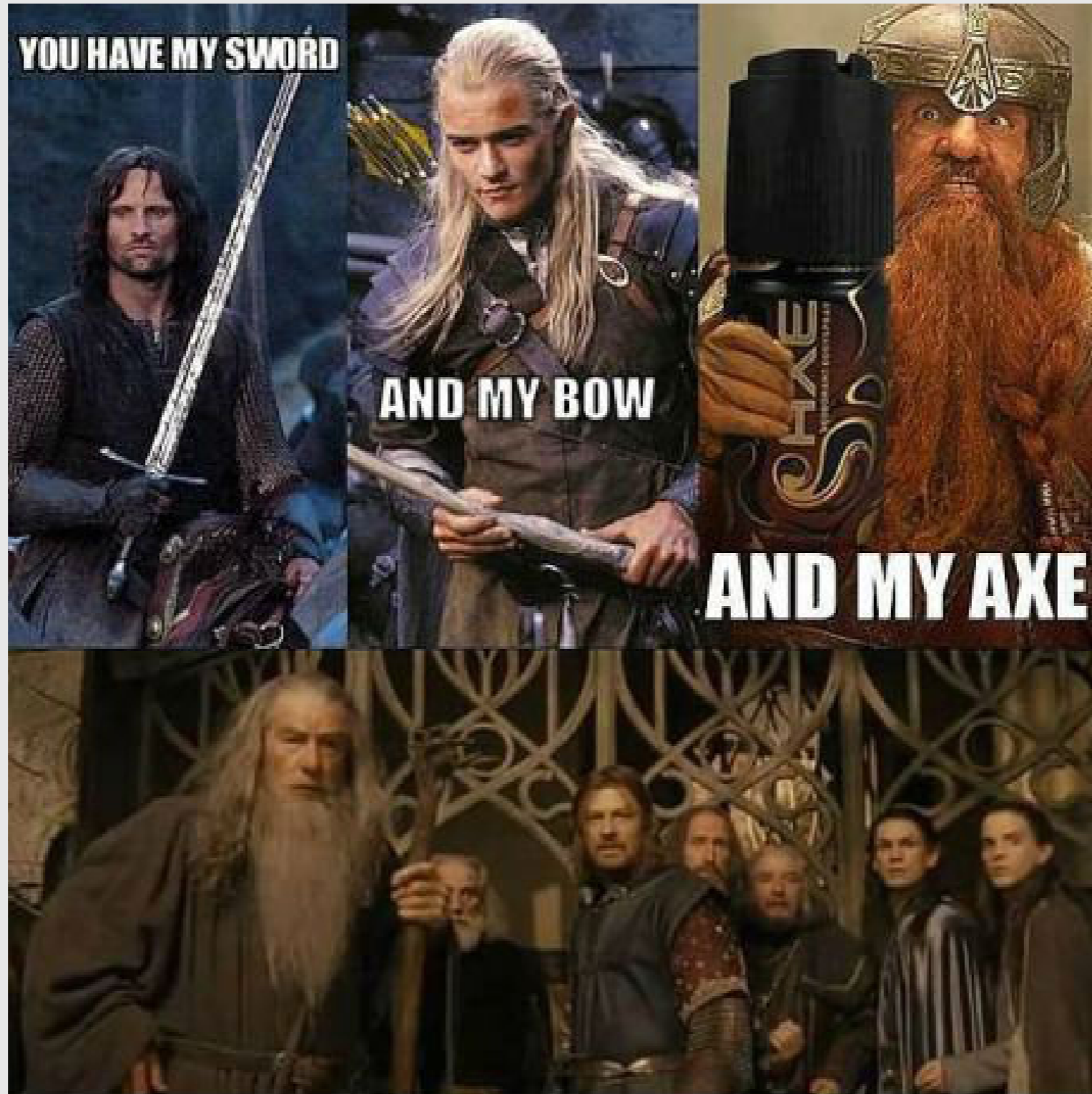


The Timeline

Be at weekly deploys by the end of the year with service A, B, and C in automatic continuous deployment

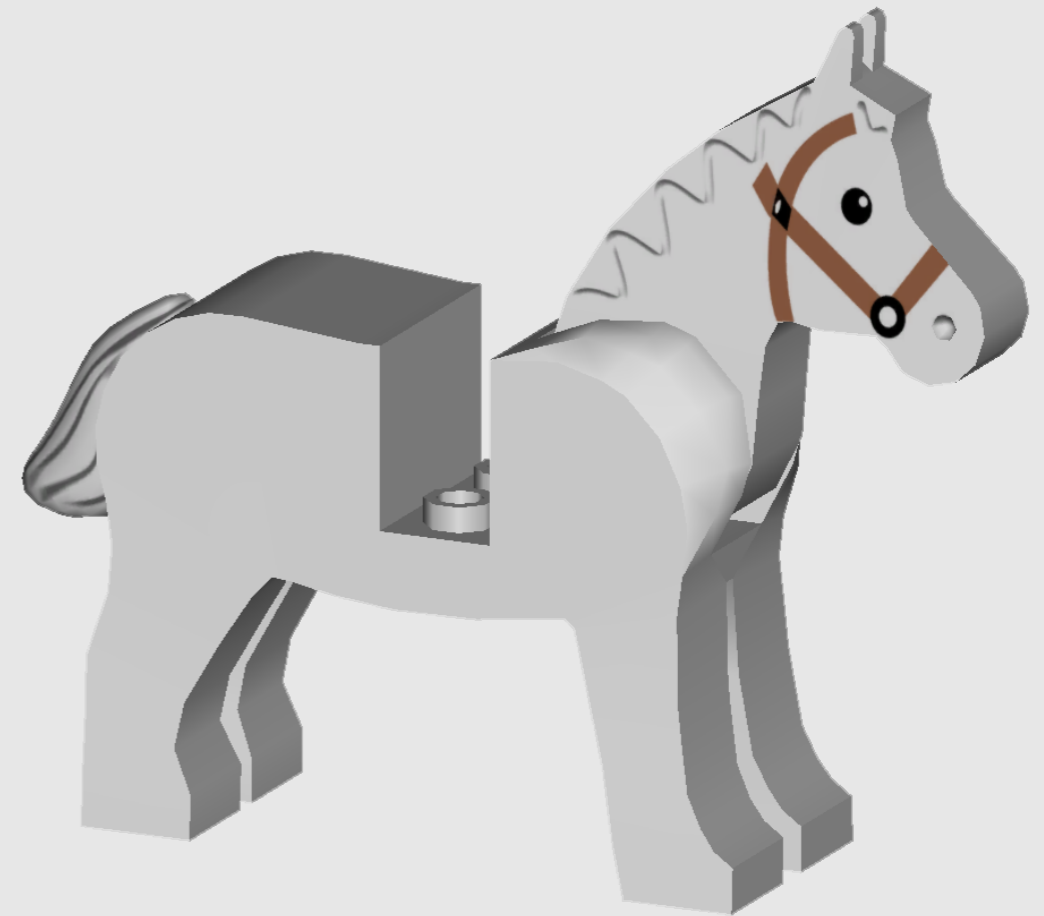


The Tools

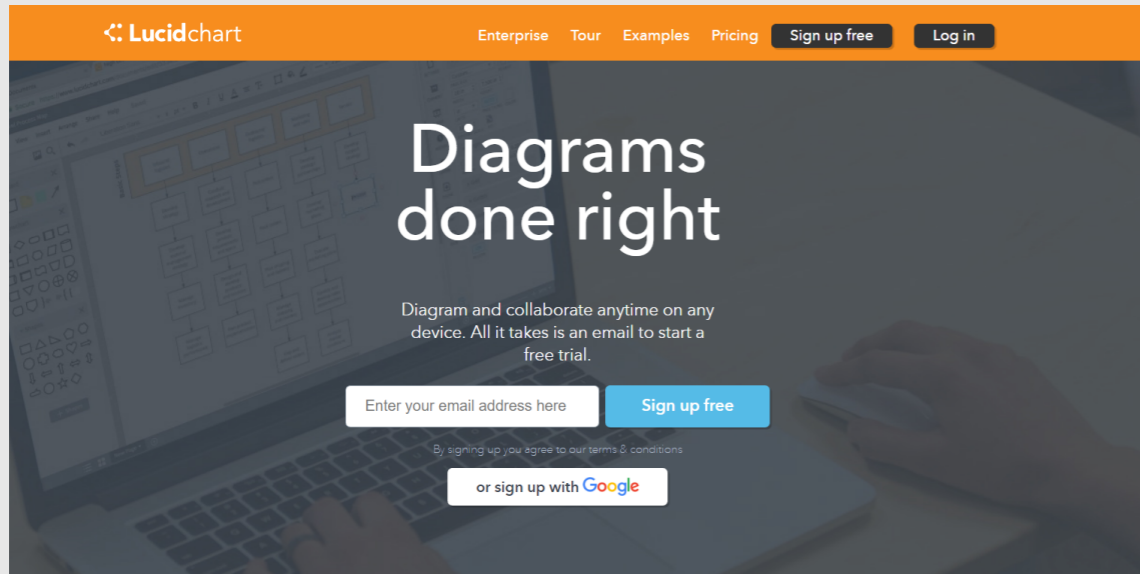


The Tools

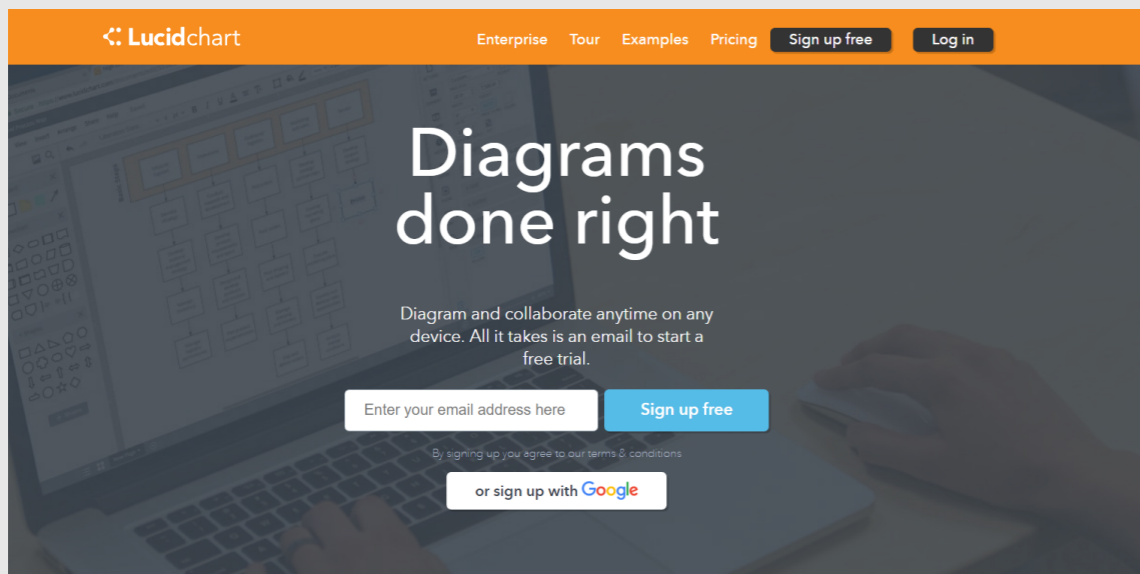
Moving Team Machines
to AWS



The Tools



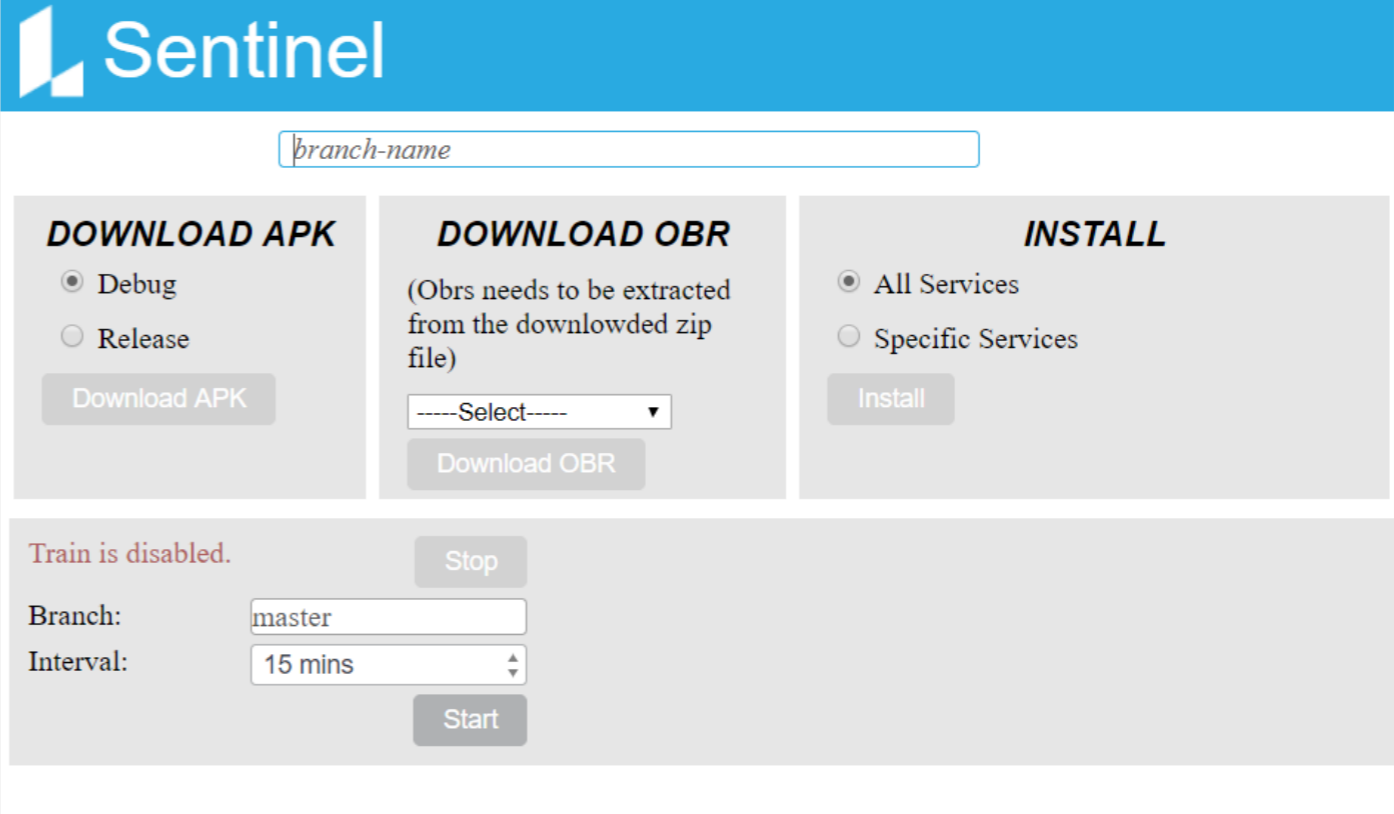
Production
=
Staging



Production-like Staging
Environment
&
Lots & Lots of
Environments

The Tools

Sentinel: A custom built user interface to quickly deploy any service from any branch to any environment



The screenshot displays the Sentinel web interface. At the top, there is a blue header with the Sentinel logo and a search bar containing the text "branch-name". Below the header, there are three main sections: "DOWNLOAD APK", "DOWNLOAD OBR", and "INSTALL".

- DOWNLOAD APK:** Includes radio buttons for "Debug" (selected) and "Release", and a "Download APK" button.
- DOWNLOAD OBR:** Includes a note "(Obrs needs to be extracted from the downloaded zip file)", a dropdown menu with "----Select----", and a "Download OBR" button.
- INSTALL:** Includes radio buttons for "All Services" (selected) and "Specific Services", and an "Install" button.

Below these sections, there is a status indicator "Train is disabled." with a "Stop" button. Further down, there are input fields for "Branch:" (containing "master") and "Interval:" (containing "15 mins"), along with a "Start" button.



The Tools

Formalizing Everything and Anything Possible for the Release

1. Scheduled Release Meeting
2. Two dedicated ops members to release (on rotation with on call)
3. An engineer from every team available and watching the release
4. Tracking the responsible team for each blocker to find trends
5. Release Manager tracks go/no-go status of release leading up to it
6. Release Manager pesters appropriate people about anything blocking the release
7. A dedicated Slack channel that is only used to update on status of release



The Tools

Extra Assistance in
Testing



The Tools



Automatic Git Hooks

- Remind people of services they are using that are CD
- Who to add to certain pull requests



The Tools

Improved Test Automation
Practices
&
Creating Reliable Automated Tests

Blog Link: [FLAKY TESTS: THE TESTER'S
F WORD](#)



The Tools

Communication, Communication, Communication

(weekly update meeting like sprint reviews, make sure each person has defined assignments, entire company including business side knows your goal)



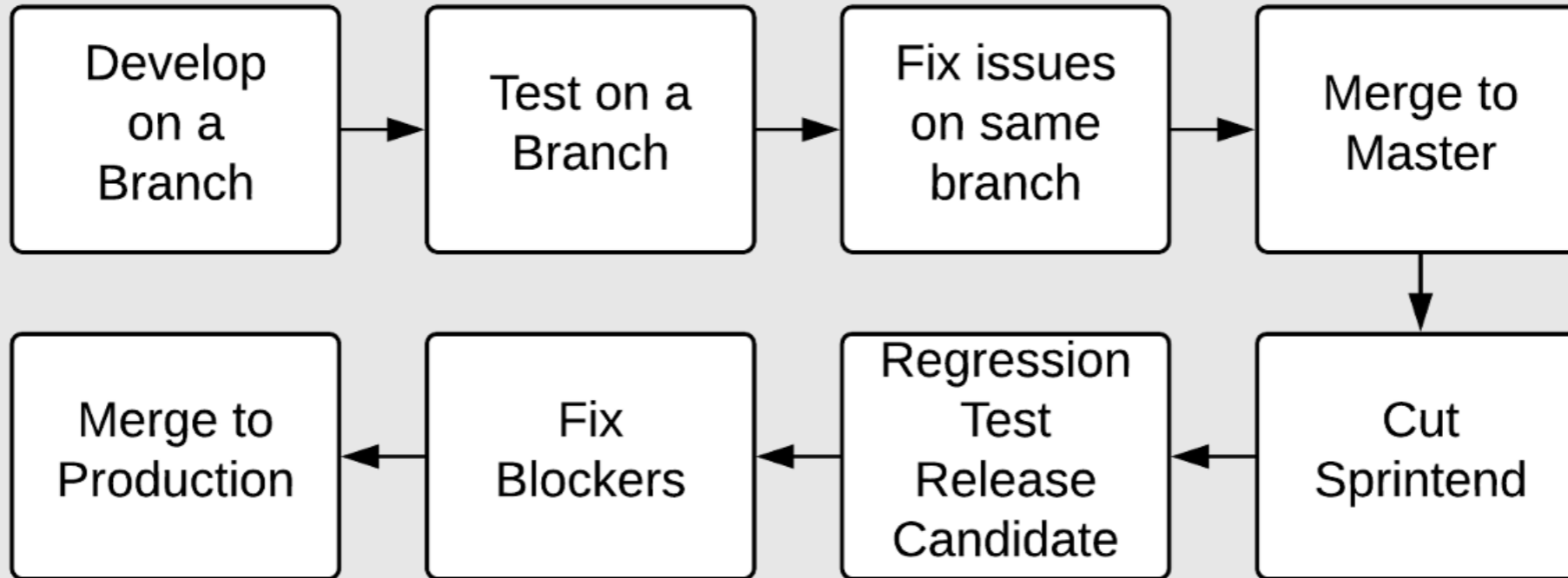
Where We Are

We have had a release of all services to production every single week for the last 10 months.

15 out of 20 services are deployed continuously

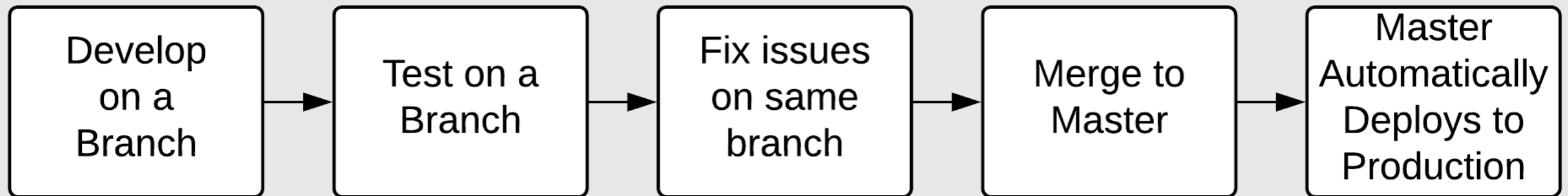


Where We Are



So What's Next?

Breaking up the Monoliths!
Release More than Once a Week?



ONE DOES NOT SIMPLY

DEPLOY CONTINUOUSLY

www.linkedin.com/in/mcampbell22