STAR EAST
A TECHWELL EVENT

T14
Agile Testing
2019-05-02 13:30

Testing in Production

Presented by:

Talia Nassi

WeWork
‘

Brought to you by:

TECHWELL™

888-268-8770 ·· 904-278-0524 - info@techwell.com - http://www.stareast.techwell.com/

Talia Nassi

Talia Nassi is a quality-driven Test Engineer at WeWork with a passion for breaking and rebuilding software to be the highest possible quality. She started interning in QA when she was studying at UC San Diego and immediately knew that she had found her calling. From UCSD she was recruited to work at Visa, where she tested the payment processing system for the Prepaid Cards. After Visa, Talia started at WeWork, where she continues to innovate and do what she lovesâ€”deliver high quality software!

Testing in Production

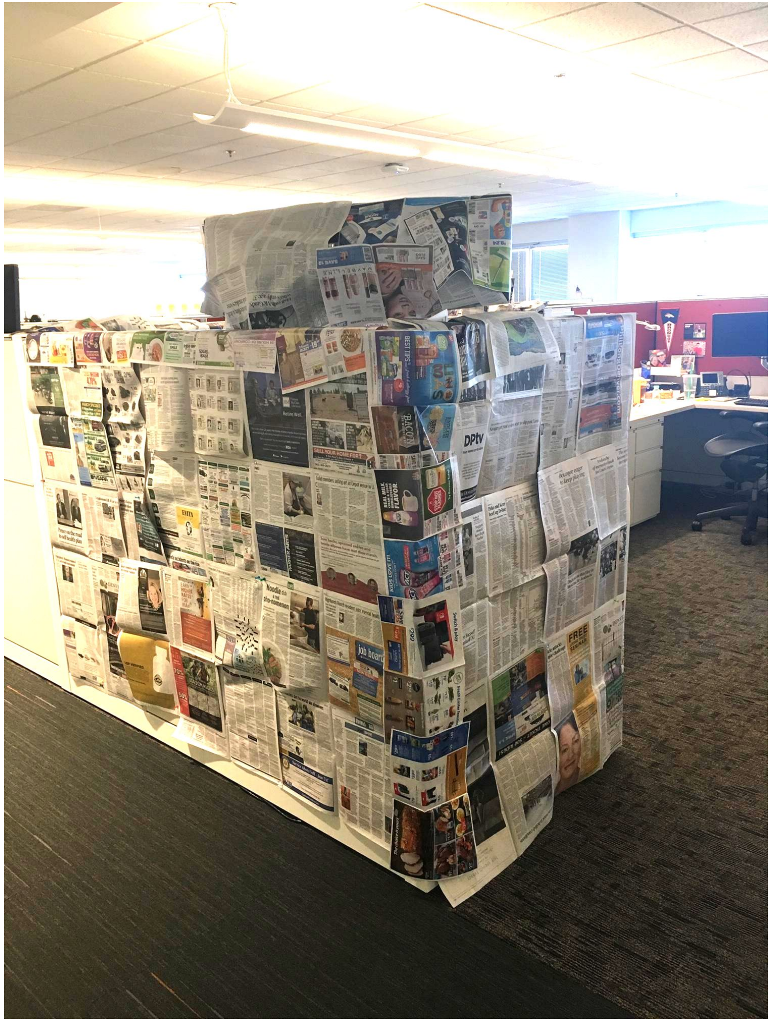
Talia Nassi

Who Am I?

- Test Engineer at WeWork
- Previously I worked at Meetup, and before that Visa
- Founder of Women Who Test Tel Aviv
- My Superpowers:
    - Turning product requirements into test cases
    - Breaking features prior to launch
    - Testing my coworkers

EXIT
Helping governments deliver greater economic development
Attack launched by U.S. linked to civilian deaths
Tar Heels hold off Ducks in thriller
EVITA
Advice on advisers
Alexa, Apple's Siri take on Marriott's hotel rooms
Summit County worried by cats
Creating a legacy
Hacking comes to Main Street
FREE
70% OFF CLEARANCE

EXIT
18-day struggle to salvage
Attack launched by U.S. linked to civilian deaths
Tar Heels hold off Ducks in thriller
EVITA
DON'T HIDE YOUR FAVORITE SIDE


BEST TIPS
BACON
DPtv
job board
FREE

Creating friction-free transactions, no matter where you are
GOLDEN STATE WARRIORS
GO WARRIORS!
GOLDEN STATE WARRIORS
GO WARRIORS!
GOLDEN STATE WARRIORS
GO WARRIORS!
GO WARRIORS!
VISA
VISA
PRESTIGE CORPORATE RELOCATION
303-469-5599

What is Testing in Production?

Testing in Production means testing your features in the environments where end users will use the features. It means your code is continuously tested to work *for real*, not in approximate environments like staging.

I know what you’re thinking.

You already do it

What’s the first thing you do right after you deploy a feature?

You go to production and test it.

My Meetup Interview

the ONLY thing WORSE you could have told me...

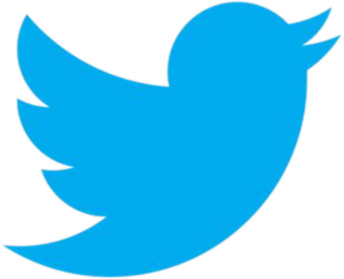
Risks

- Can affect real end users
- Can affect data and analytics > business decisions
- Can affect third parties that your software is integrated with
- You can’t create test data in production
- What if your tests end up breaking things in production?

NETFLIX

amazon

Meetup



Staging environments are expensive to maintain

FACT:
Companies spend millions of dollars on staging environments every year

Staging test results do not always match production test results

FACT:
Staging data does not match production data

The load in staging does not match production - comparatively, no one uses staging

No one cares if staging is broken, it’s not a priority

FACT:
No one is going to get a call in the middle of Thanksgiving dinner if staging is down.

The production environment includes “garbage” data that staging doesn’t have that can cause issues

What’s wrong with staging environments?

1. Staging environments are expensive to maintain

2. Staging test results do not always match production test results

3. No one cares if staging is broken, it’s not a priority

4. The load in staging does not match production

5. Production environment includes “garbage” data that staging doesn’t have that can cause issues

“I love my staging environment”

-no one ever

All valid points, but I was still hesitant.

I did my homework

“The standard solution is not the standard because it’s bad, it’s the standard solution because it’s good at delivering a certain effect and once you start saying you want to go beyond standard and you want to **raise the bar** and you pile on more demands, suddenly the standard solution doesn’t work anymore. By trying to make more people happy, you make the standard solution insufficient and you **force something better to the table**.”

-Bjarke Ingels
Chief Architect, WeWork

Here’s what else I learned.

Testing in Production only works when...

the whole team owns product quality.

Testing in Production only works when...

you test early and often.

Testing in Production only works when...

you trust your team and your product.

FEAR

Think about the last feature your team deployed.

Is it working?
Right now?
In production?

How do you know?

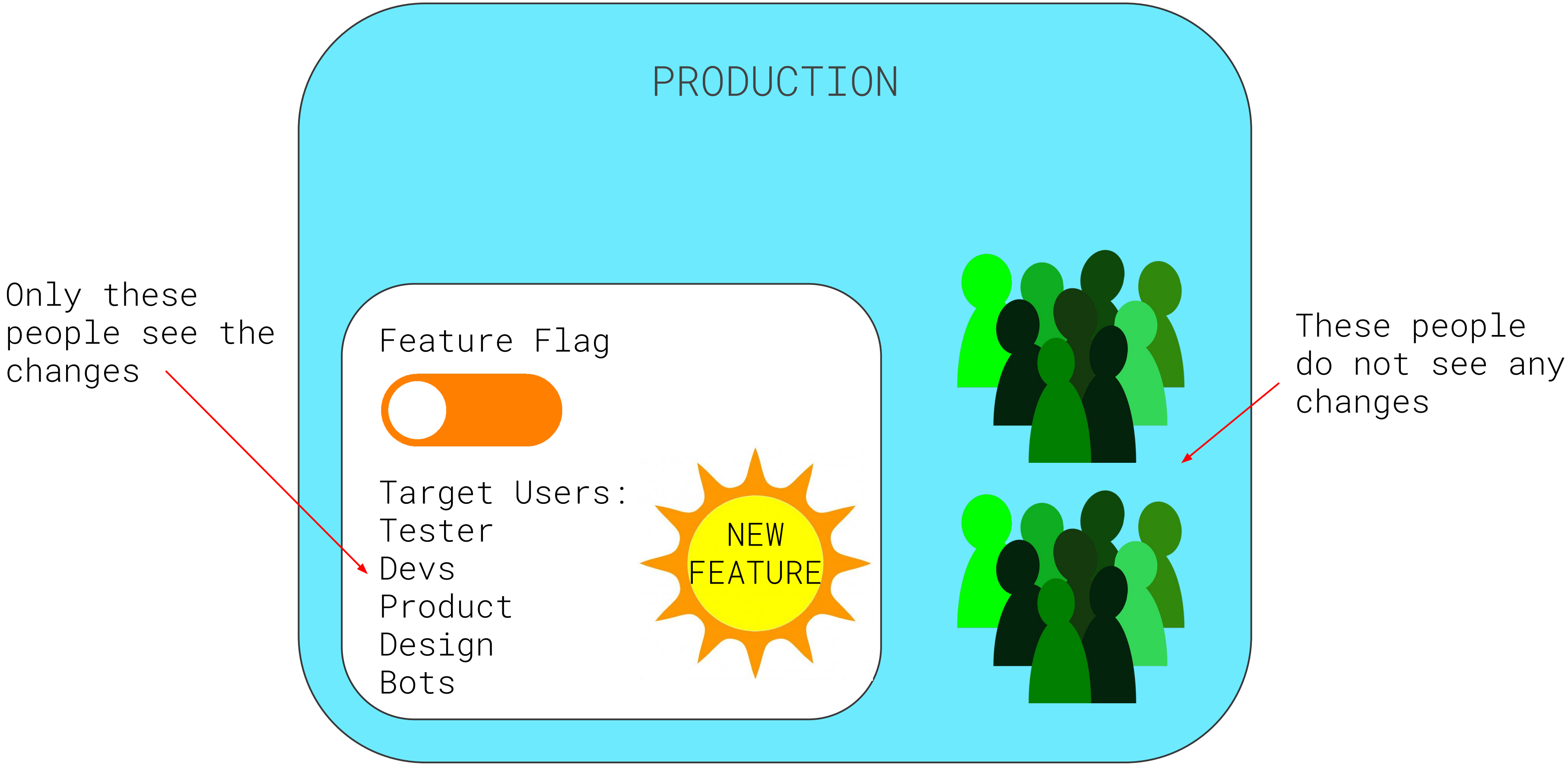
Testing in Production is the only way to know that your features are **working in production right now**

Step 1: Install Necessary Tools

Feature Flagging

It’s a way to decide who sees which features

It’s used to hide, enable or disable the feature during run time



PRODUCTION

Feature Flag

Target Users:
Tester
Devs
Product
Design
Bots

NEW FEATURE

Only these people see the changes

These people do not see any changes

WHILE FEATURE FLAG IS OFF:

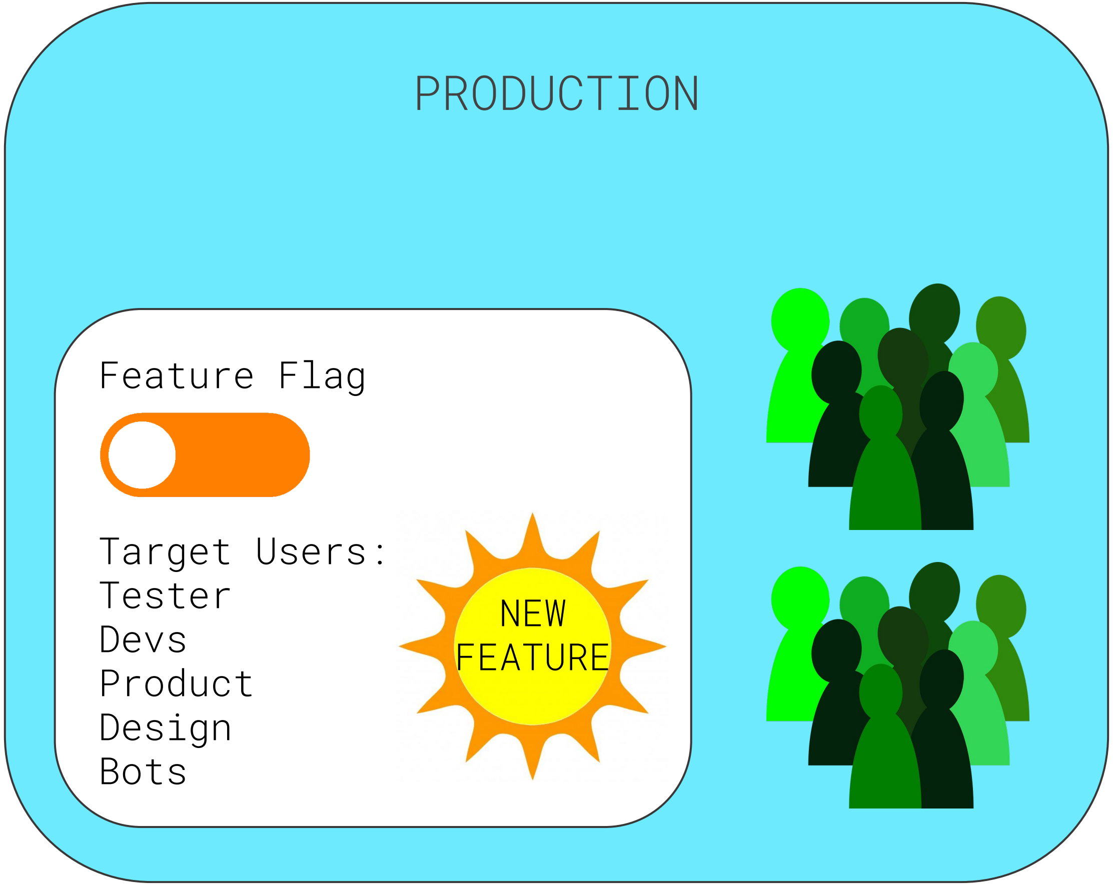
TEST REQUIREMENTS

OPEN DEFECTS

WRITE AUTOMATION SCRIPTS

VERIFY DESIGN

VALIDATE PROPER FUNCTIONALITY



PRODUCTION

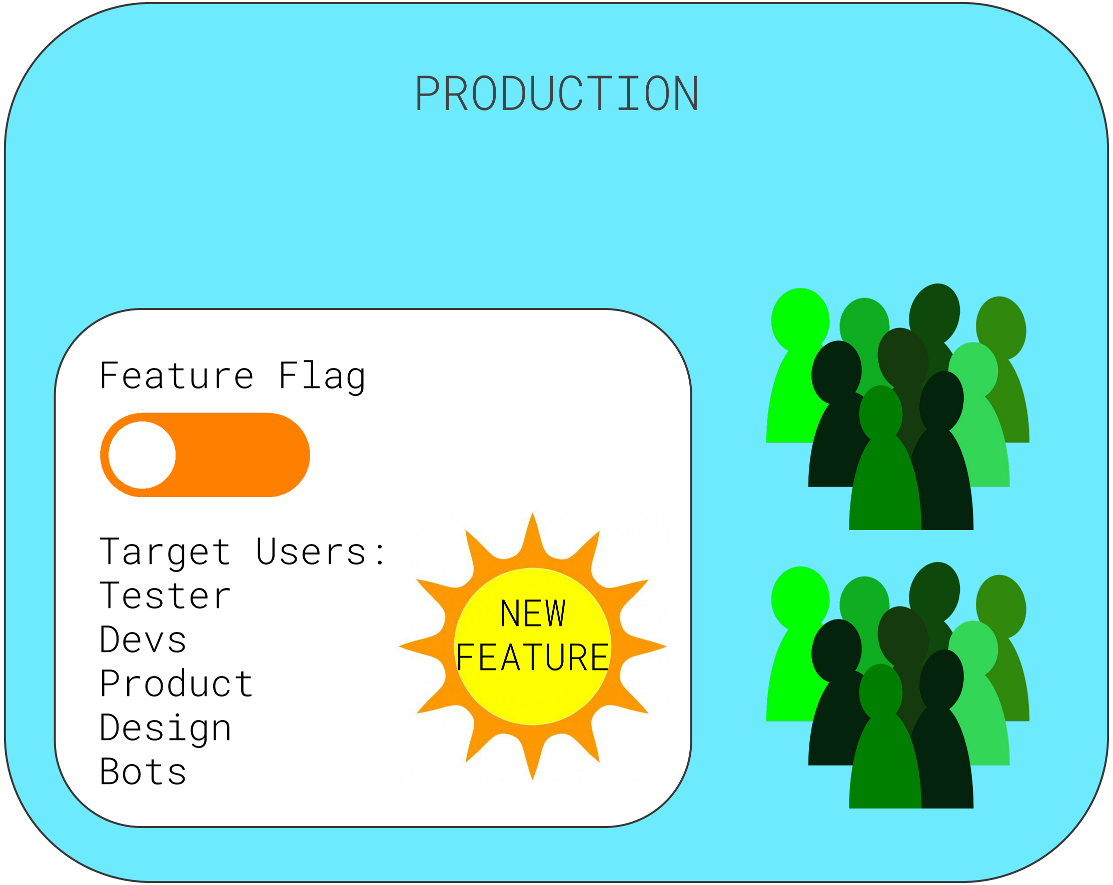
Feature Flag

Target Users:
Tester
Devs
Product
Design
Bots

NEW FEATURE

If there were bugs in our new feature, no end users would be affected because they are not targeted in the feature flag.

The bugs were fixed before the flag was turned on and before the users ever saw anything wrong

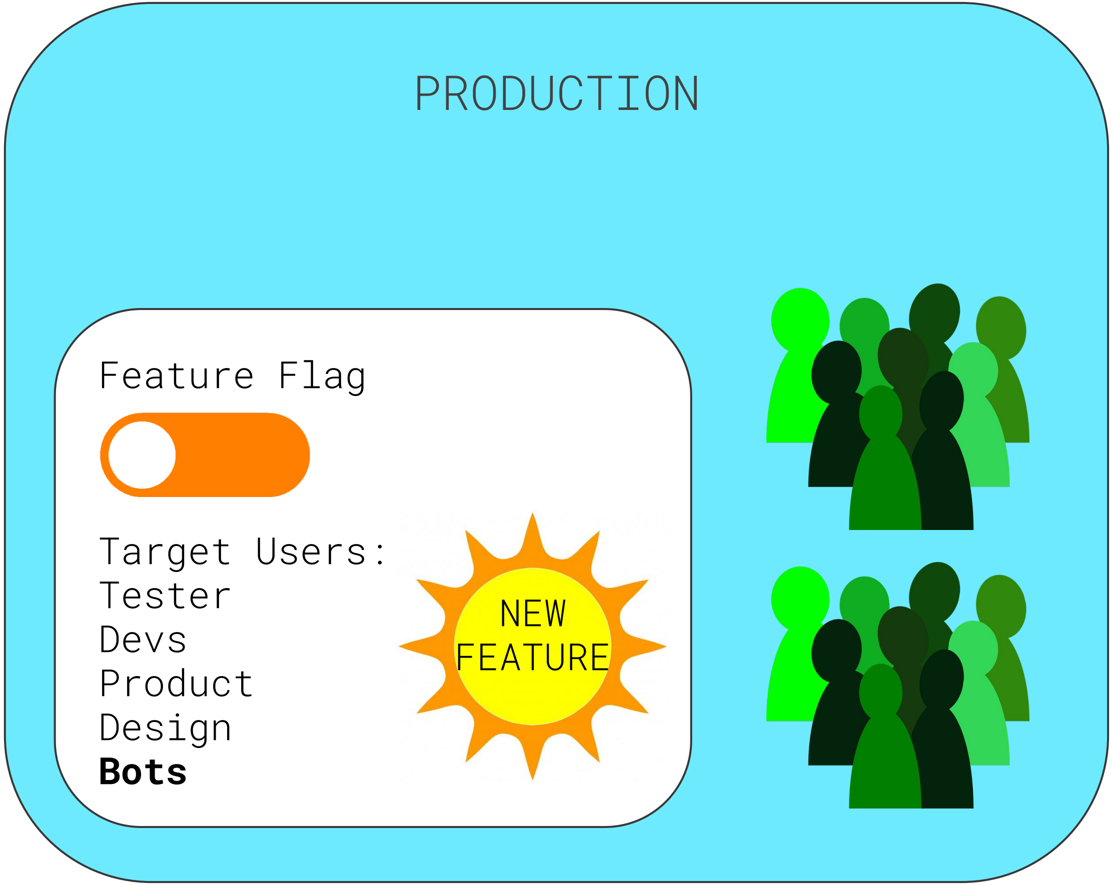


PRODUCTION

Feature Flag

Target Users:
Tester
Devs
Product
Design
Bots

NEW FEATURE

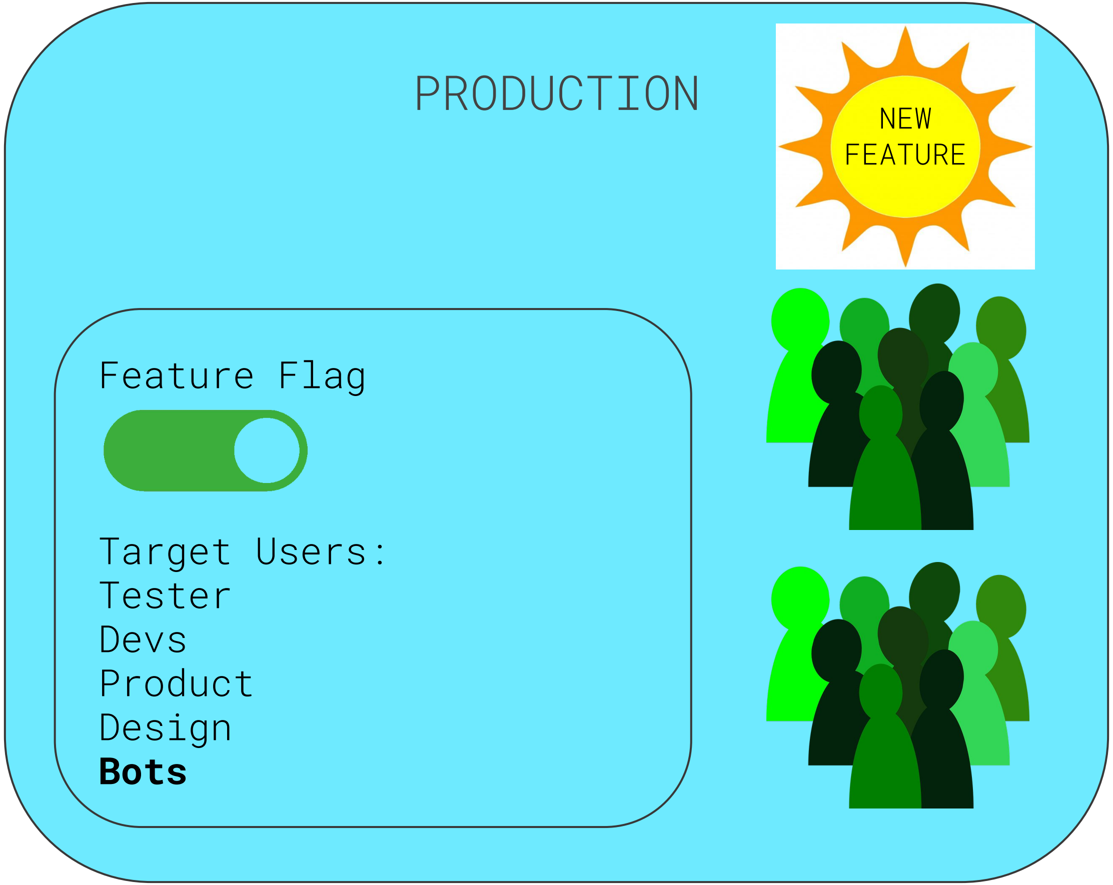


PRODUCTION

Feature Flag

Target Users:
Tester
Devs
Product
Design
Bots

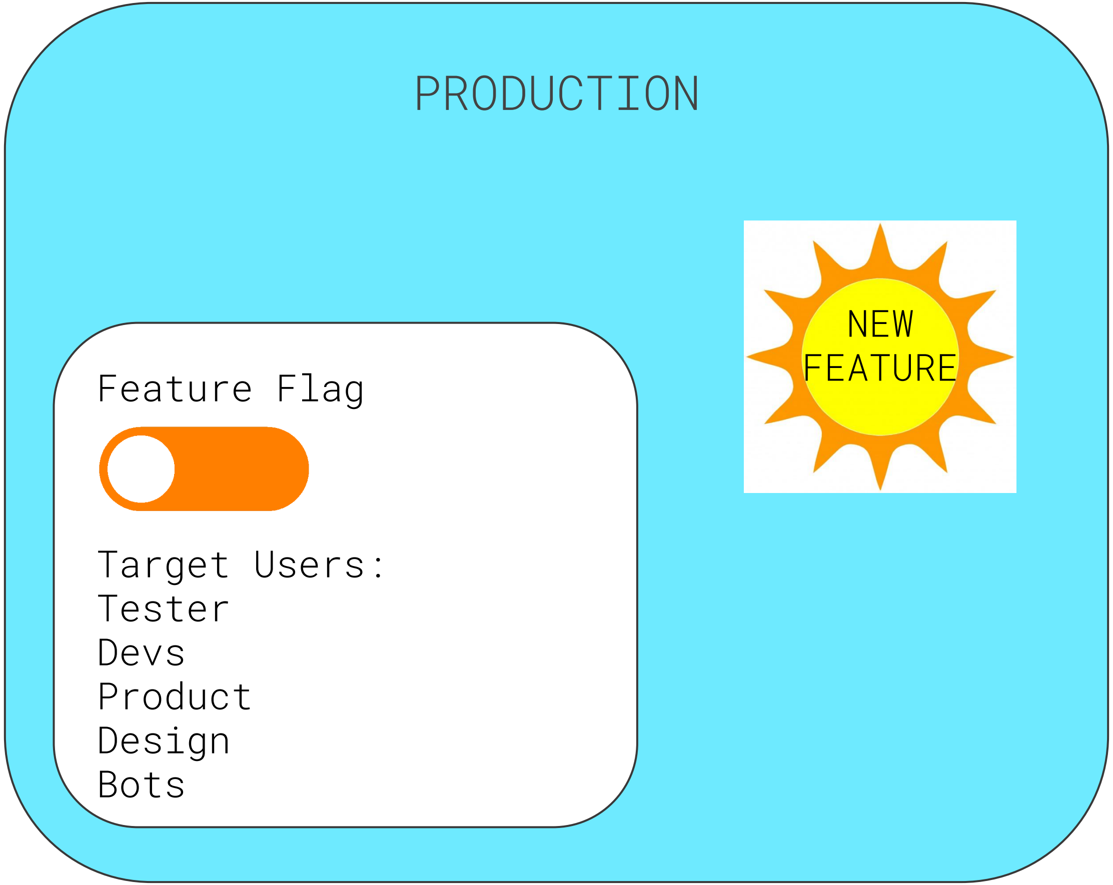
EVERYONE SEES THE NEW FEATURE



PRODUCTION

Feature Flag

Target Users:
Tester
Devs
Product
Design
Bots



PRODUCTION

Feature Flag

Target Users:
Tester
Devs
Product
Design
Bots

Automation Framework

1. Easy to adopt
2. Easy to debug
3. Good reporting
4. Support community

Automation Framework

```
pip install robotframework
```

```
npm i puppeteer
```

```
npm install -g @angular/cli
```

Job Scheduler

I will run the tests every 30 minutes for you

Job Scheduler

Why not just run the tests in a loop?

- It’s messy
- Creates garbage data
- Reduces too much load

Tools

Feature Flagging	LaunchDarkly split
Automation Framework	Puppeteer +
Job Scheduler	circleci

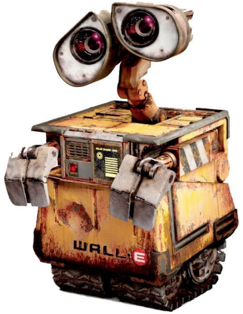
Step 2: (Carefully) Create Test Data

Problem:

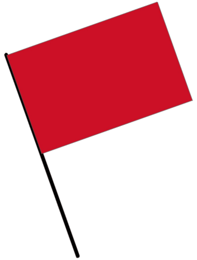
We needed a way to create and manipulate test data in production without affecting real end users or any data and analytics.

Solution:

- Consistent naming convention for test users

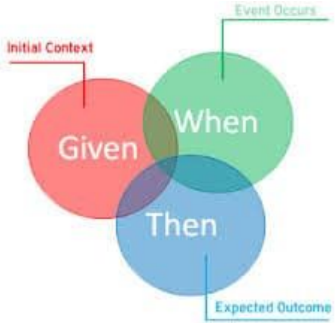


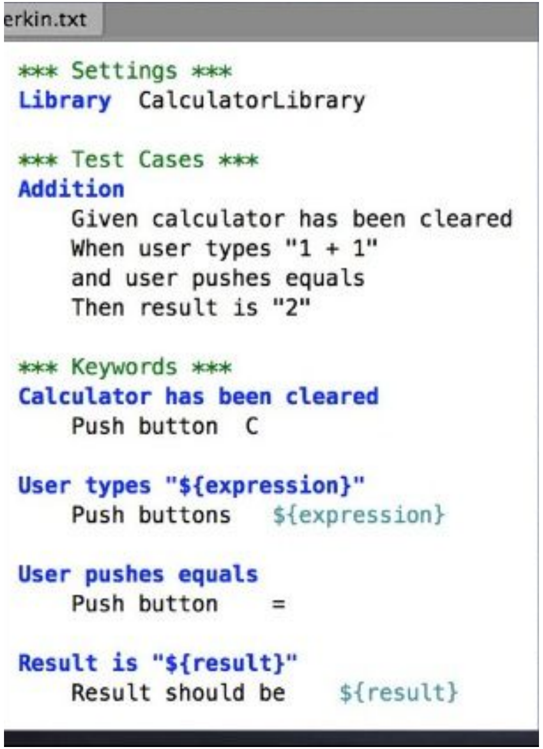
- Backend flagging system used to identify test groups



Step 3: Write Your Tests

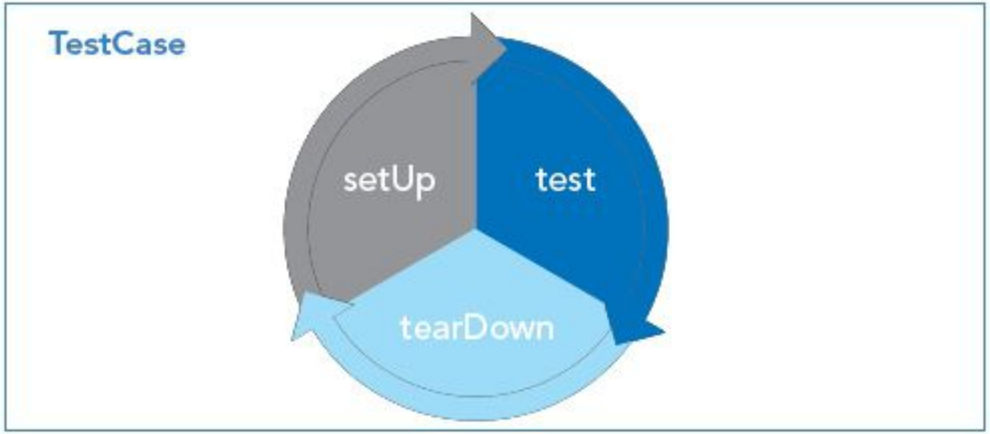
BDD





```
*** Settings ***
Library  CalculatorLibrary

*** Test Cases ***
Addition
    Given calculator has been cleared
    When user types "1 + 1"
    and user pushes equals
    Then result is "2"

*** Keywords ***
Calculator has been cleared
    Push button  C

User types "${expression}"
    Push buttons   ${expression}

User pushes equals
    Push button    =

Result is "${result}"
    Result should be    ${result}
```

Setup/Teardown



Step 4: Deploy to Production Canary

What’s a Production Canary?

It’s when you slowly roll out the change to a small subset of users before rolling it out to the entire infrastructure to minimize impact if something goes wrong

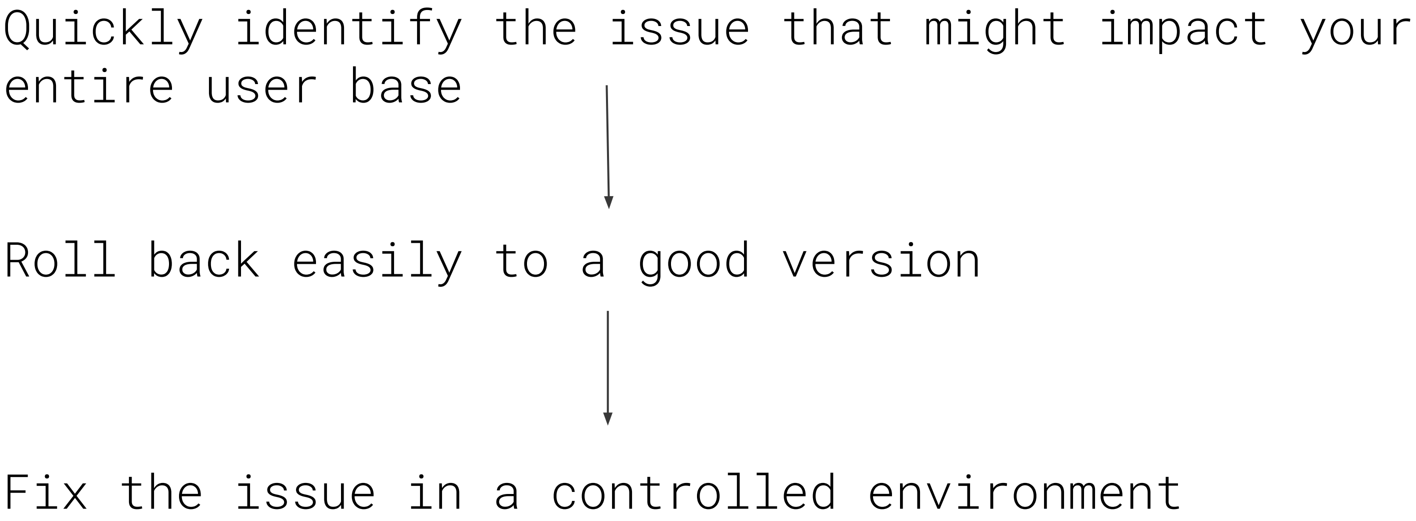
Why use a production canary?

Why use a production canary?

Canary launches provide Risk Mitigation

Do you want 100% of your users to encounter the issue or 1%?

Why use a production canary?

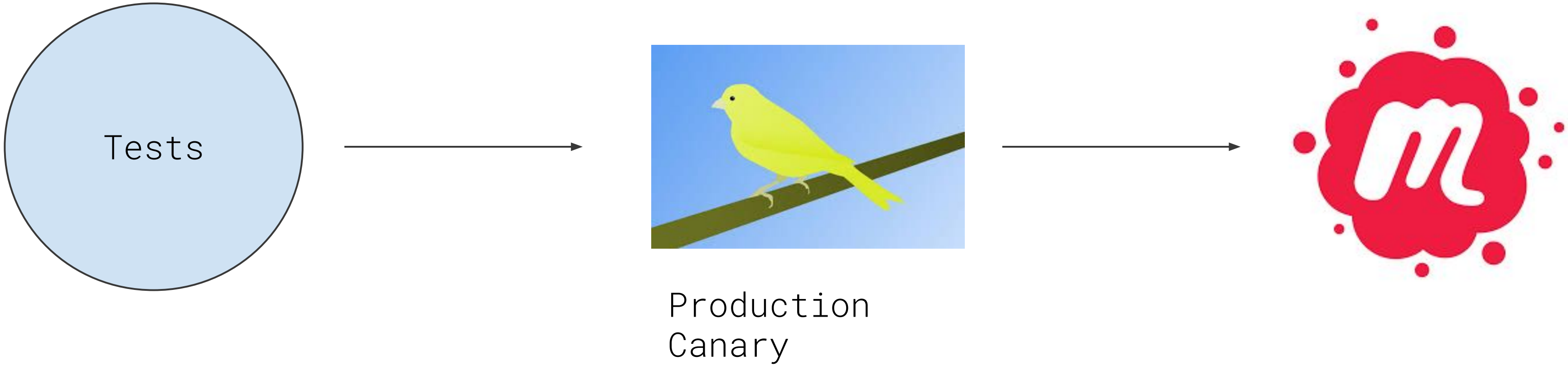


Quickly identify the issue that might impact your entire user base

Roll back easily to a good version

Fix the issue in a controlled environment

Production Canaries



Tests

Production Canary

Risk Mitigation

1. Production Canaries to limit audience
2. Feature Flagging to target users

OUTCOME = OUTSTANDING

HIGHER CONFIDENCE

INCREASED DEVELOPER VELOCITY

~~Reactive~~
Proactive

Long-Term Effects

FAIL

Tests would fail

Immediately get alerted

Analyze the issue right away

Resolve it ASAP

Long-Term Effects

⛔ Minimize user interaction with bugs and defects

✅ Ensures a great user experience

Shifting Your Company’s Testing Culture

Shifting Your Company’s Testing Culture

1. Explain why you think the pros outweigh the cons for your company

Is your staging environment unreliable?

Are there frequently issues that you think could have been caught if you were testing in prod?

Shifting Your Company’s Testing Culture

2. Use examples from the past

Do you remember when we merged xyz and it caused this issue in production?

Do you think if we tested xyz in production that that issue could have been caught?

Do you remember when we tested xyz inside and out in staging and then we deployed to prod and it broke?

Shifting Your Company’s Testing Culture

3. Propose a path forward

Have you heard about this cool thing called feature flagging? Can I take some time in the next sprint to see if we could benefit from it?

If we were to start testing in prod,which tests do you think would bring us the most value?

Shifting **MY** Company’s Testing Culture

We used to be scared of deploying new features

We used to have debates of whether or not to deploy code on Fridays

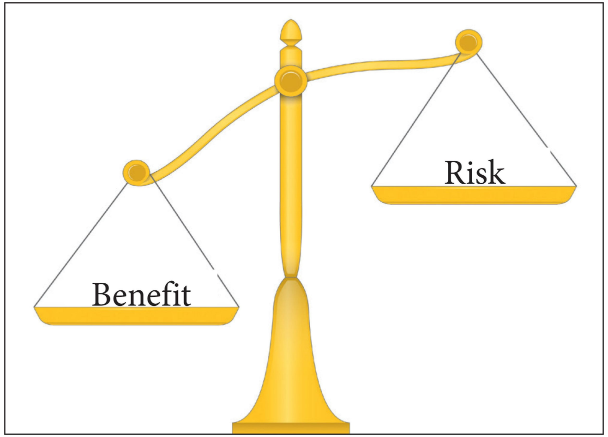
Shifting **MY** Company’s Testing Culture

Once we started moving more and more things to testing in prod, these discussions and this fear stopped.

Shifting **MY** Company’s Testing Culture

The lead time to know if something is wrong is reduced and the confidence in the release is increased

Shifting **MY** Company’s Testing Culture



The benefits outweigh the risks completely

How to deal with naysayers

How to deal with naysayers

1. Staging will never fully represent prod
2. Staging is a sunk cost
3. They’re not your target audience

Summary

No one cares if your feature is working in staging, we care if it’s working in production.

To provide risk mitigation, use feature flagging and production canaries.

The only way to know if its working in prod is to test it in prod.

Resources

Podcast by Mike Bryzek:
https://www.infoq.com/podcasts/Michael-Bryzek-testing-in-production

Saucelabs Article:
https://saucelabs.com/blog/why-you-should-be-testing-in-production

Questions?

Talia Nassi
talia.nassi@wework.com