

## *Contents*

Background

Management Experience

Project Experience

- Enterprise Identity Service
- User Onboarding Experience
- Application Development Bootstrapping

**Michael Long**  
Product Designer + Manager



# I've lived and worked in...



San Francisco



Seattle



New York



Austin



Dallas

# I've worked for and with...

ThoughtWorks®



Southwest®

TOYOTA

neo

CHASE



Bloomberg

★macy's

Adobe®

2009 – 2014

# I've worked for and with...



2009 – 2014

# I've worked for and with...



**Southwest**<sup>®</sup>



**neo**

**CHASE**

**Bloomberg**

**macy's**

**Adobe**<sup>®</sup>

2009 – 2014

# I've worked for and with...



**Southwest**<sup>®</sup>



**neo**



**Bloomberg**



2009 – 2014

# I've worked for and with...



**Southwest**<sup>®</sup>



**neo**

**CHASE**

**Bloomberg**

**macy's**

**Adobe**<sup>®</sup>

2009 – 2014

# I've worked for and with...



2009 – 2014

# I've worked for and with...



2009 – 2014

# I've worked for and with...



2009 – 2014

# I've worked for and with...



2009 – 2014

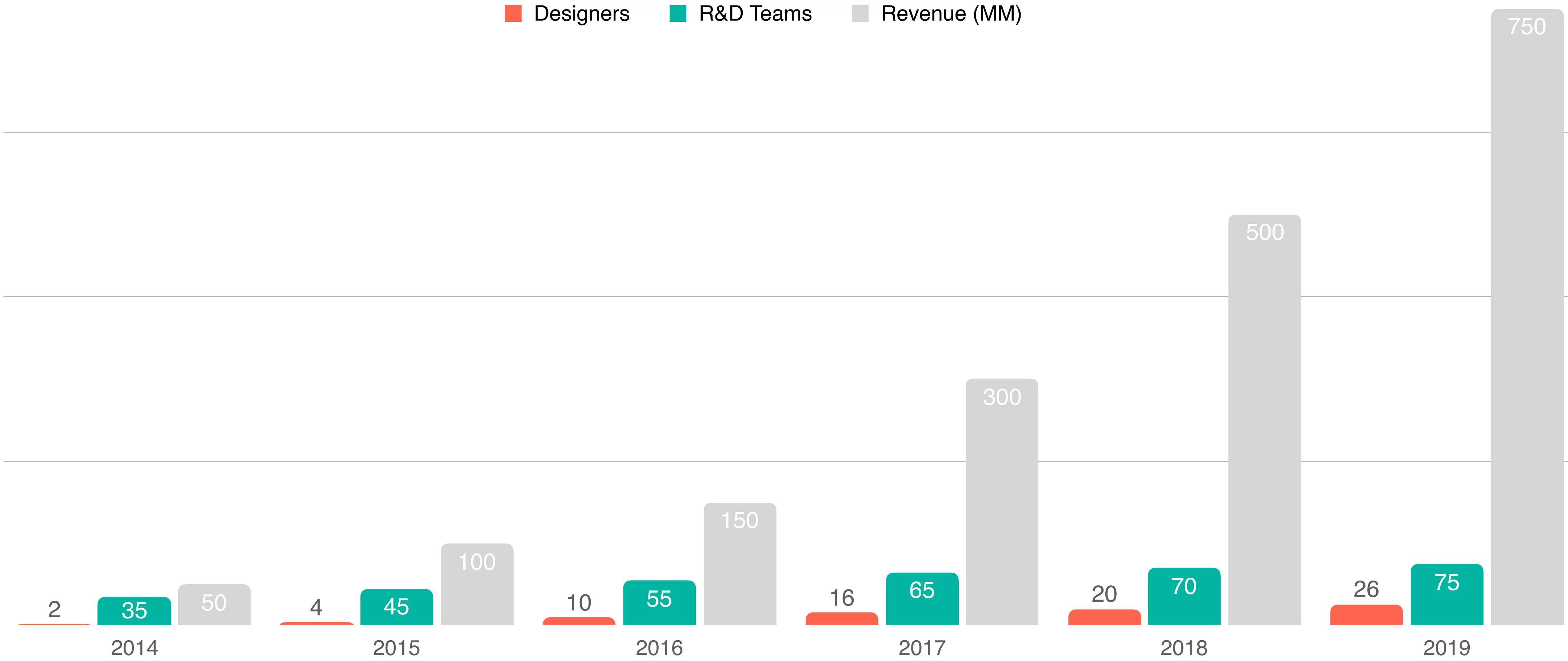
# I've worked for and with...

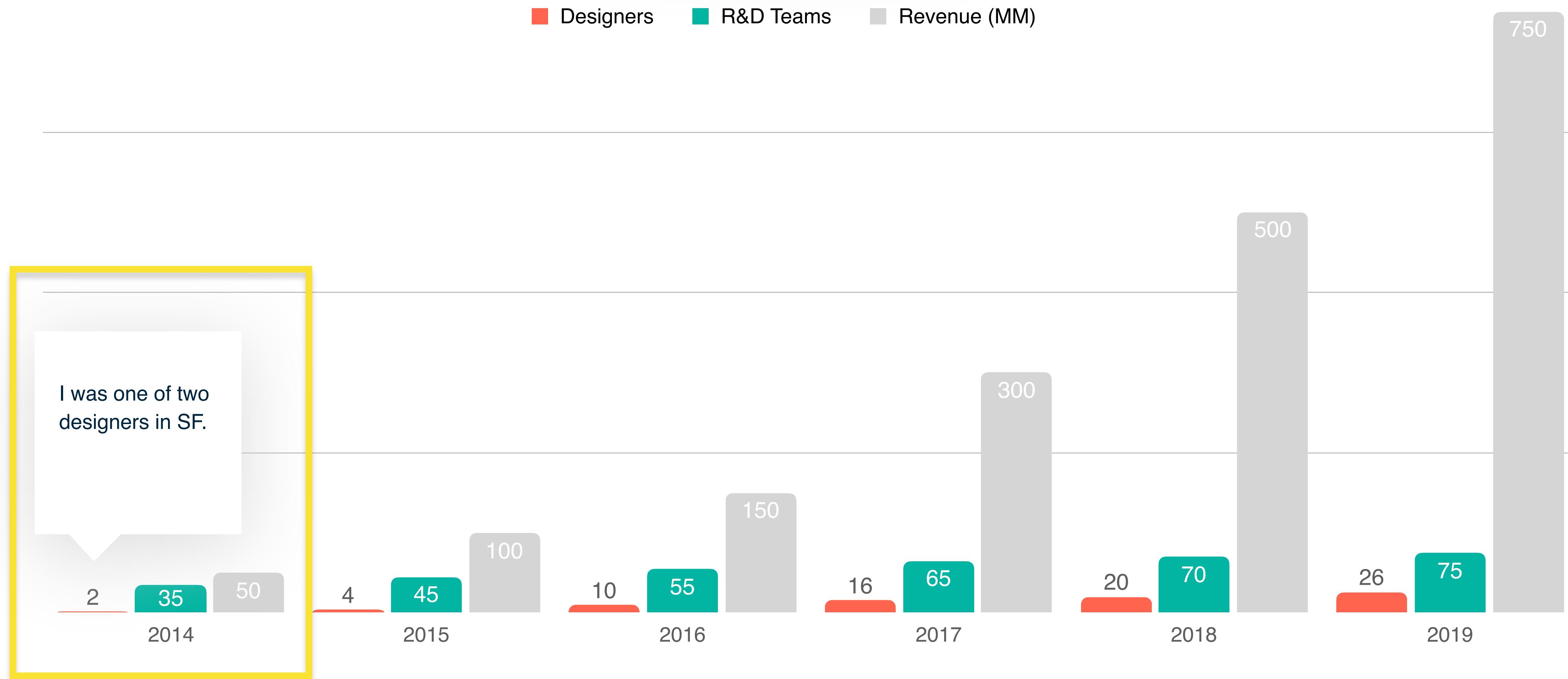


Pivotal<sup>®</sup>

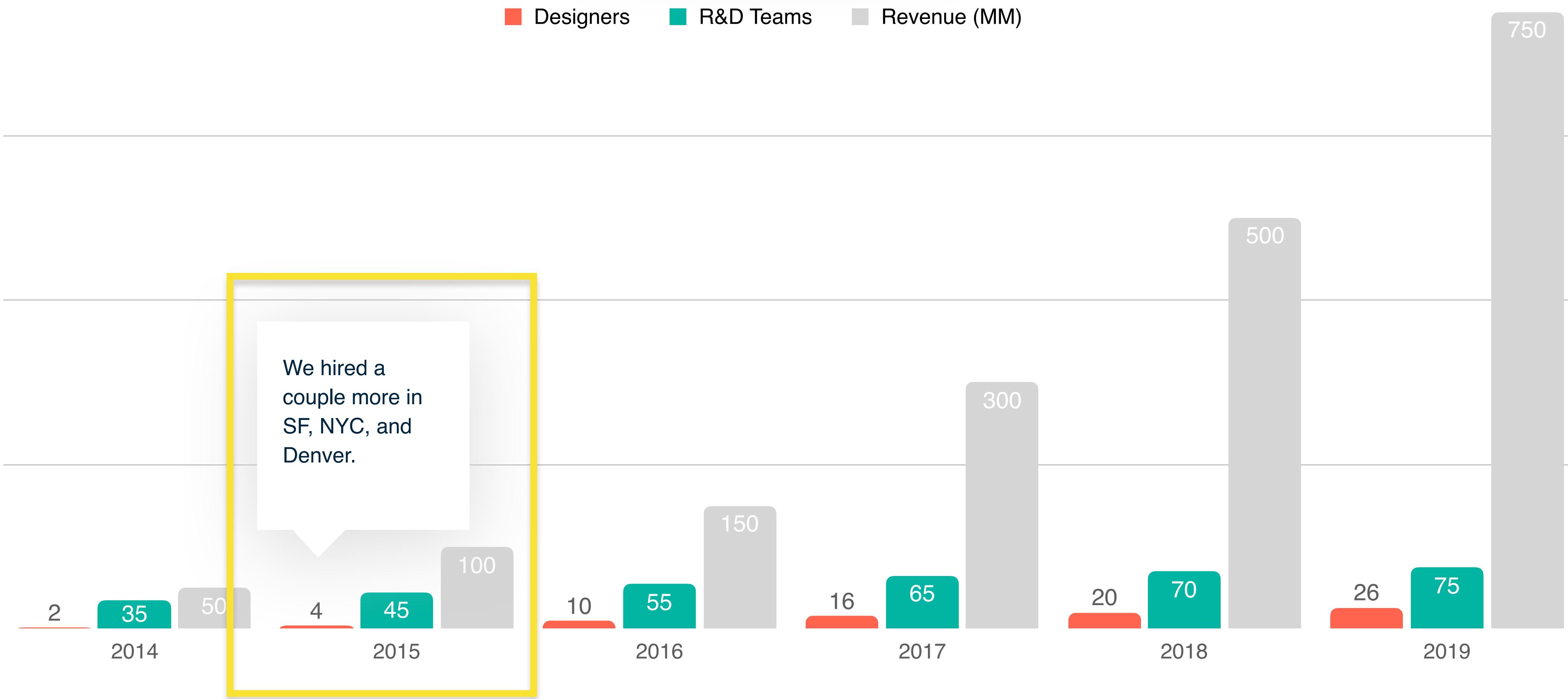
2014 – 2019

# Pivotal®





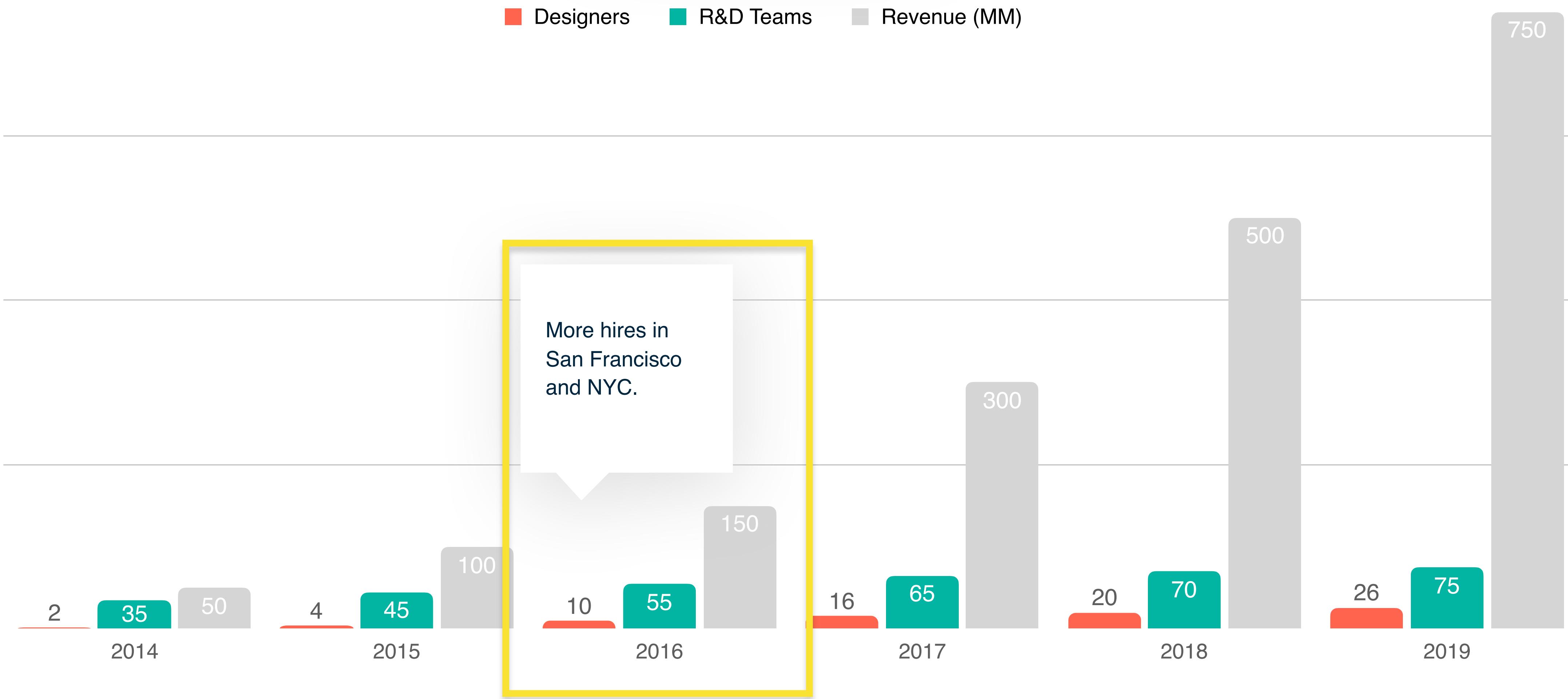
# Pivotal®



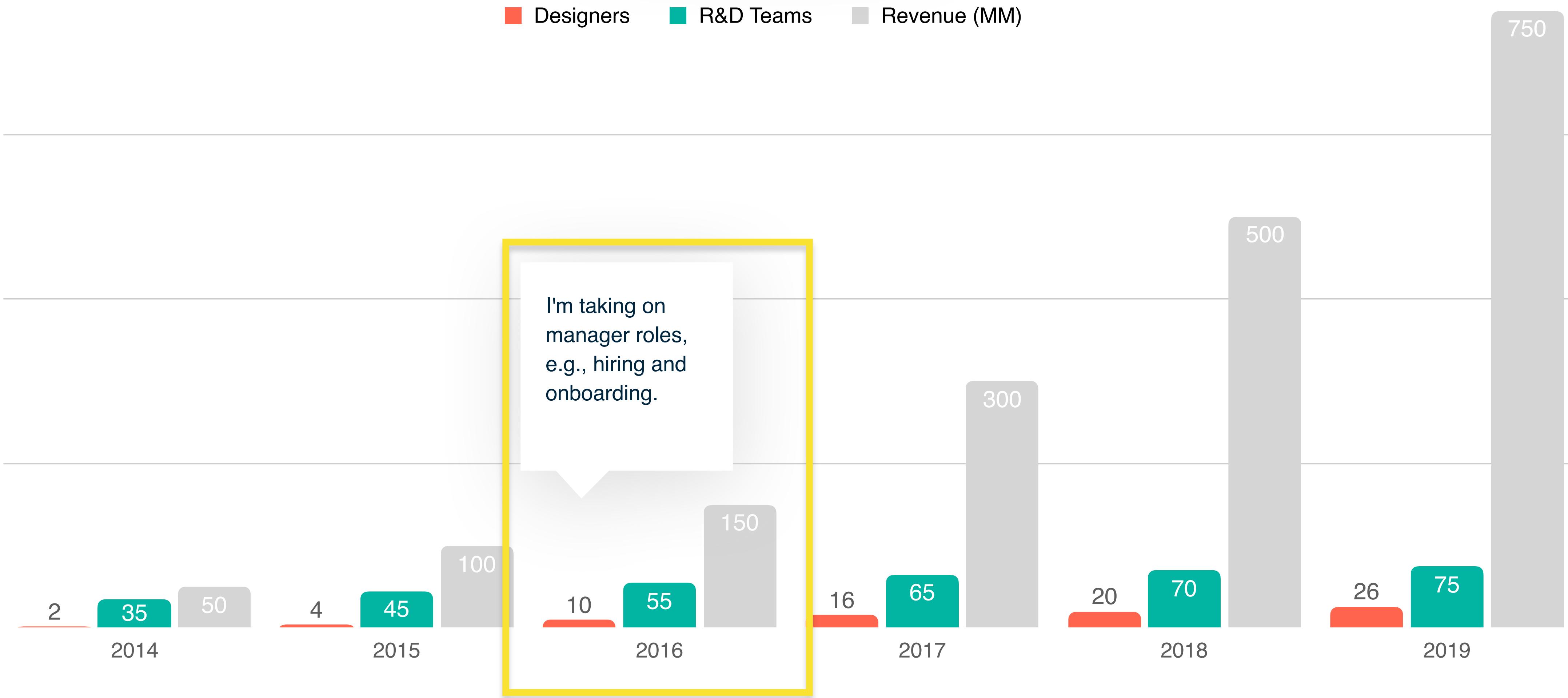
# Pivotal®



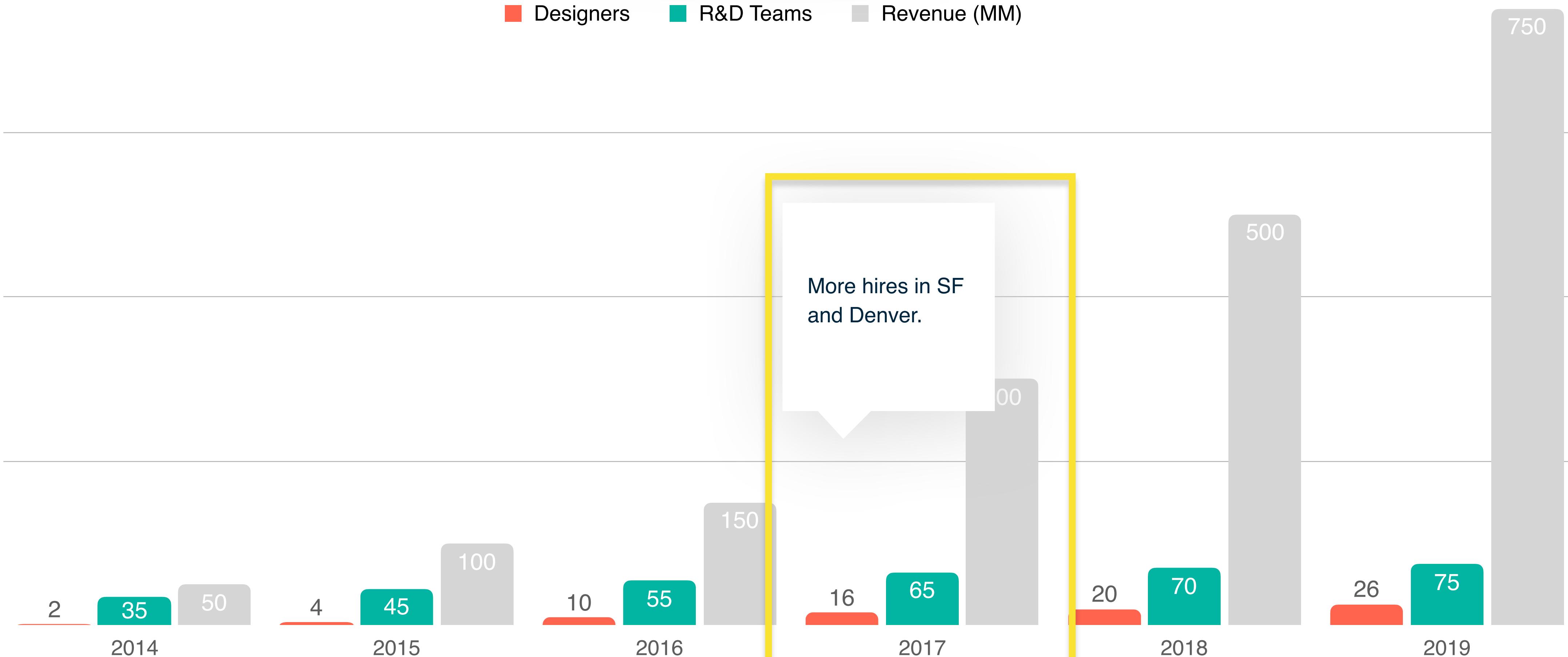
# Pivotal®



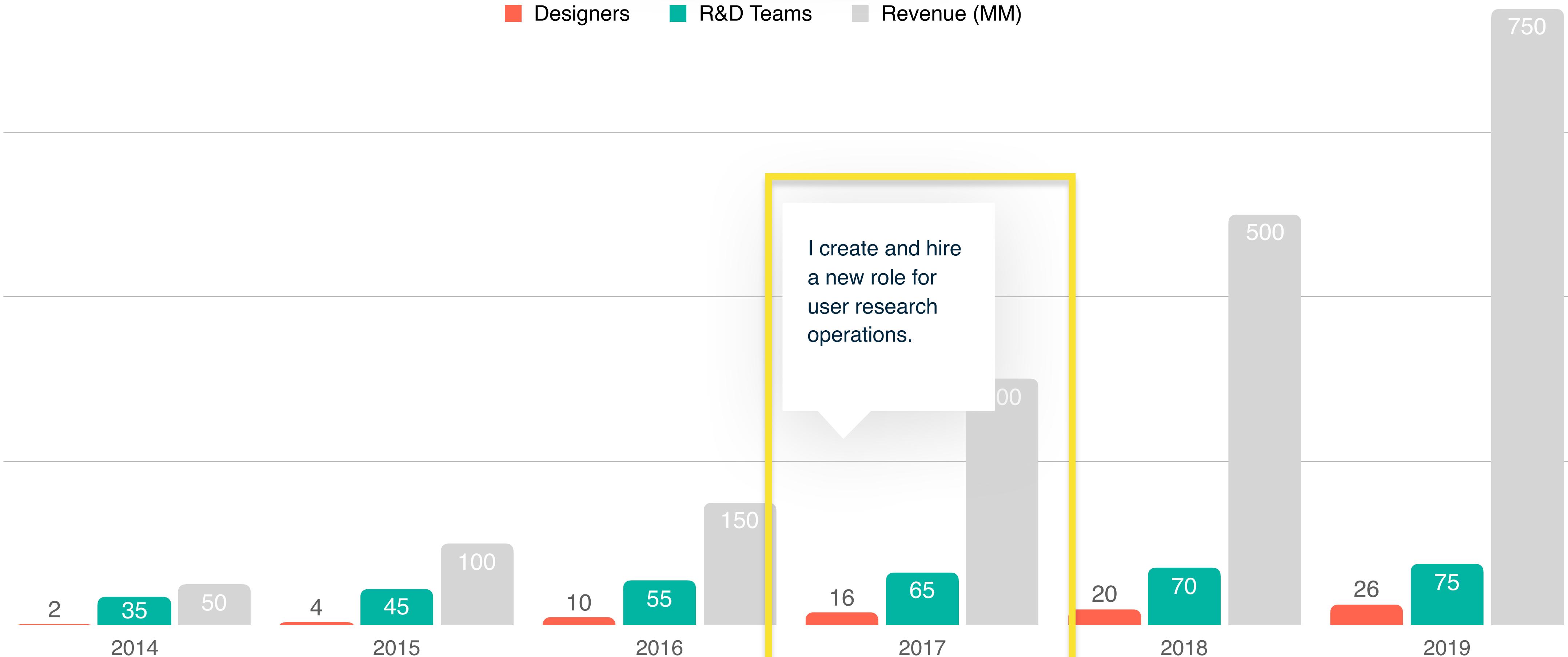
# Pivotal®



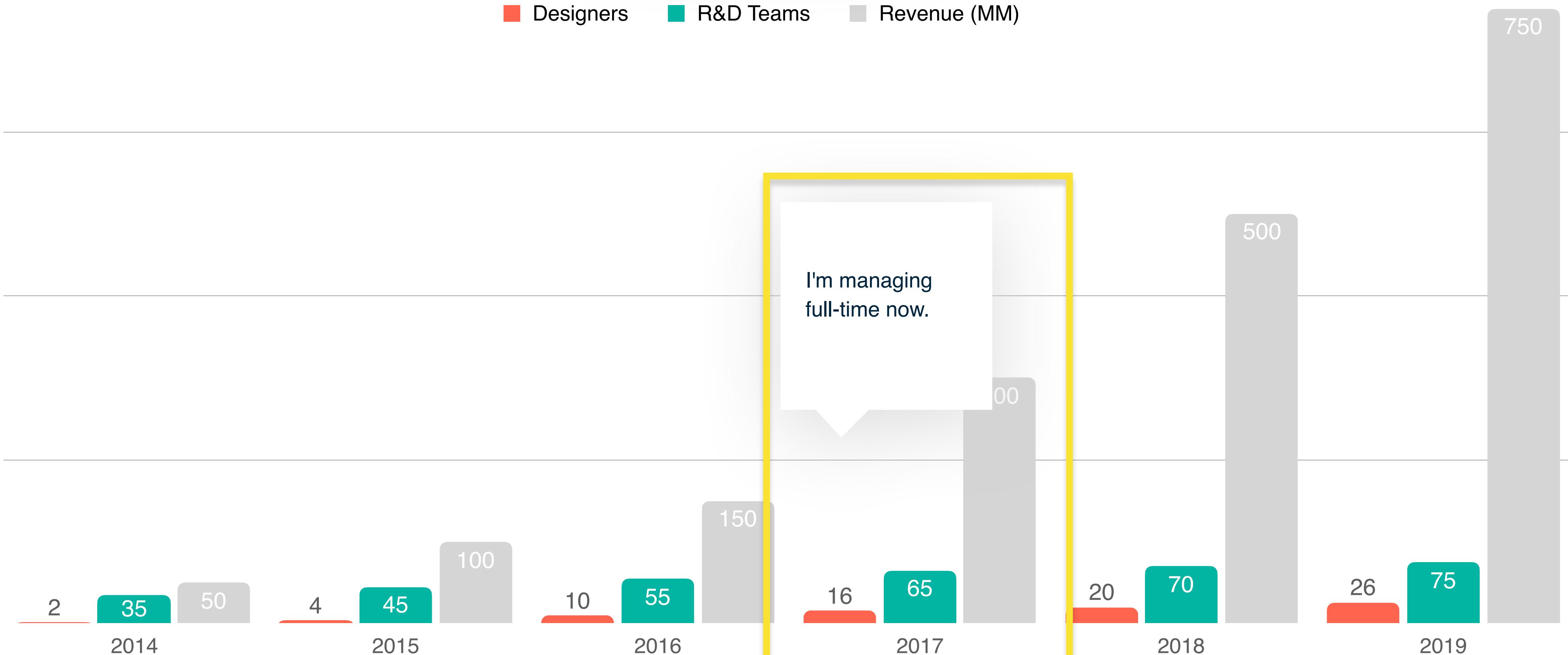
# Pivotal®



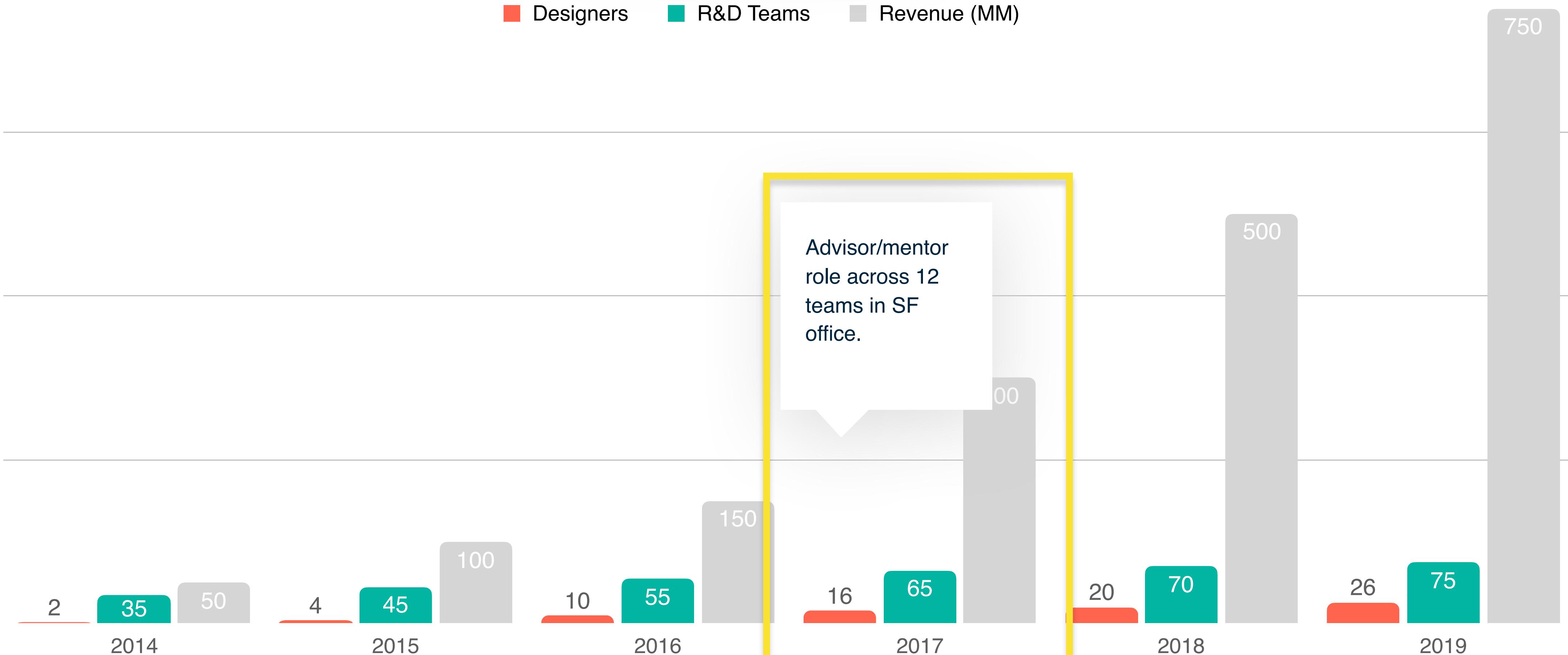
# Pivotal®



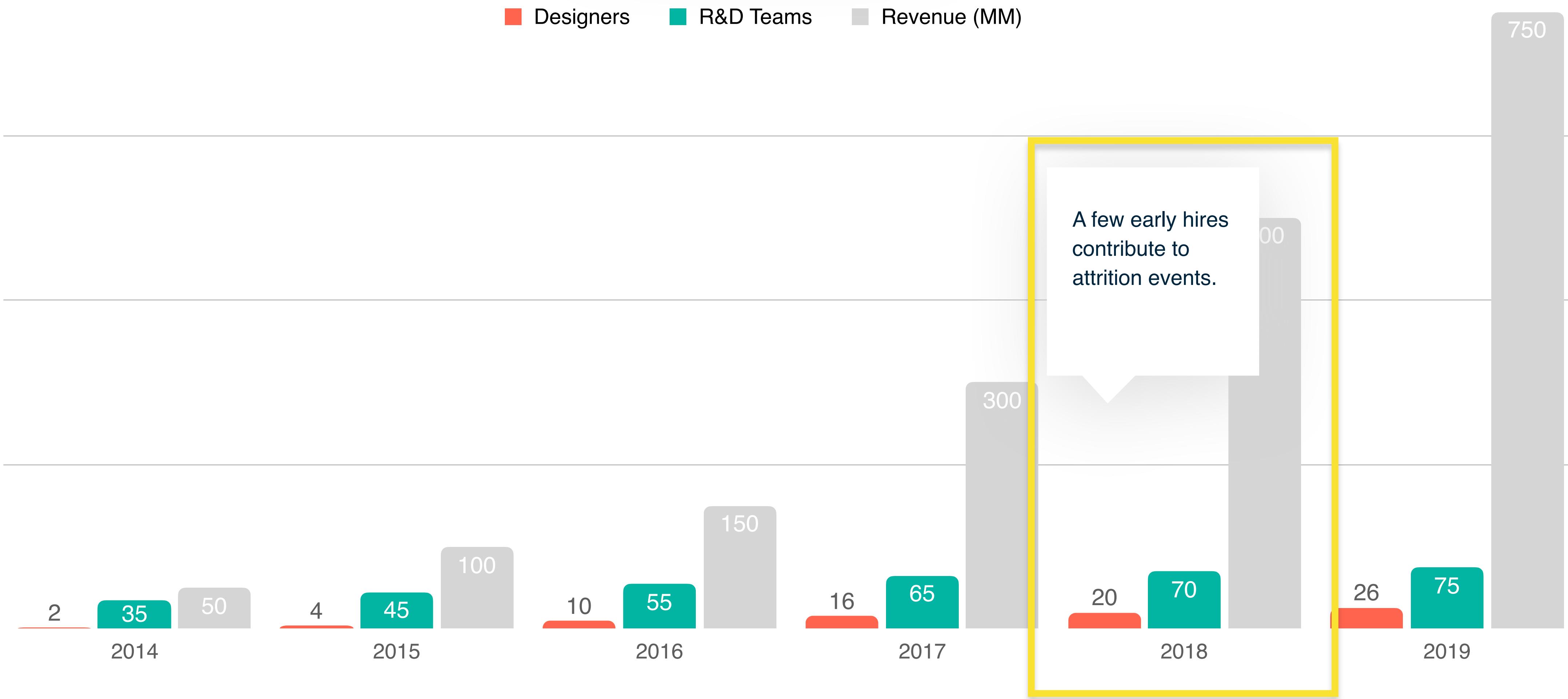
# Pivotal®



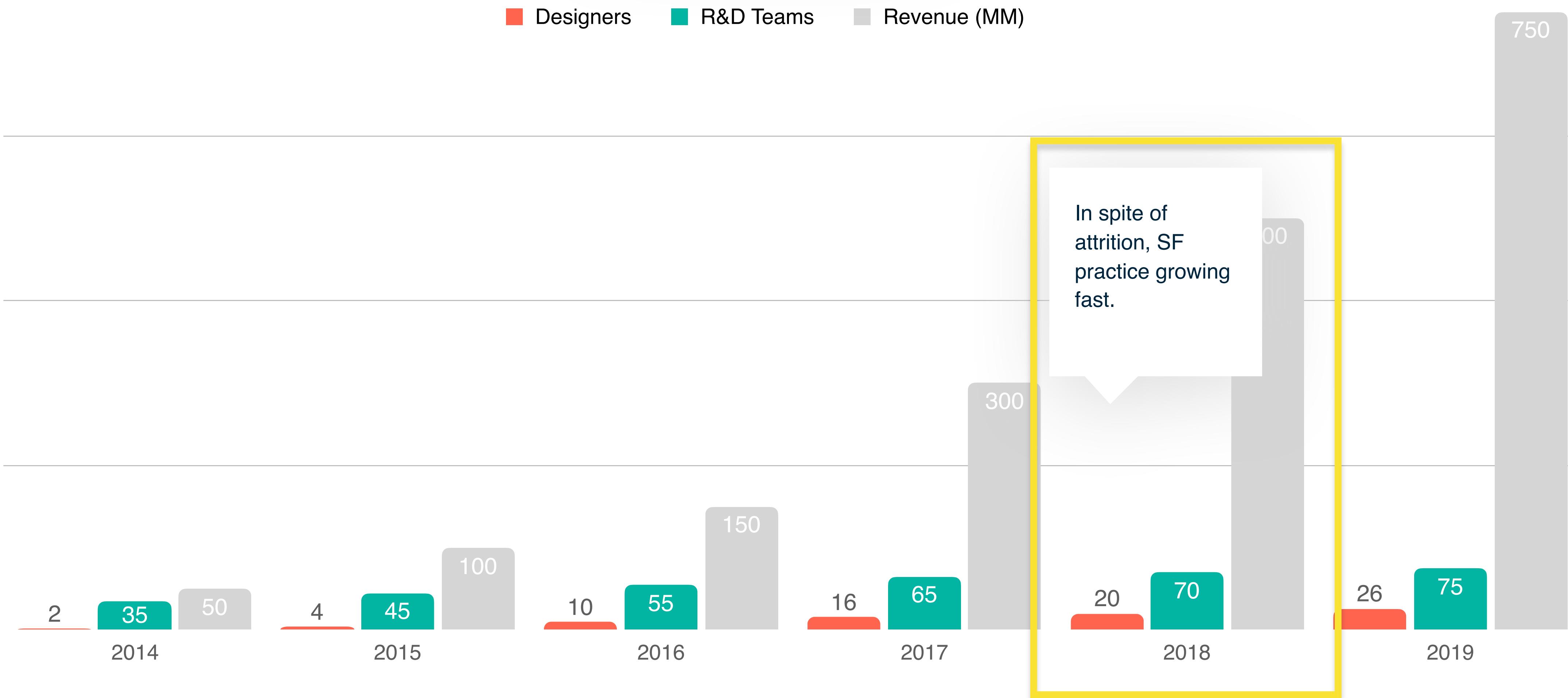
# Pivotal®



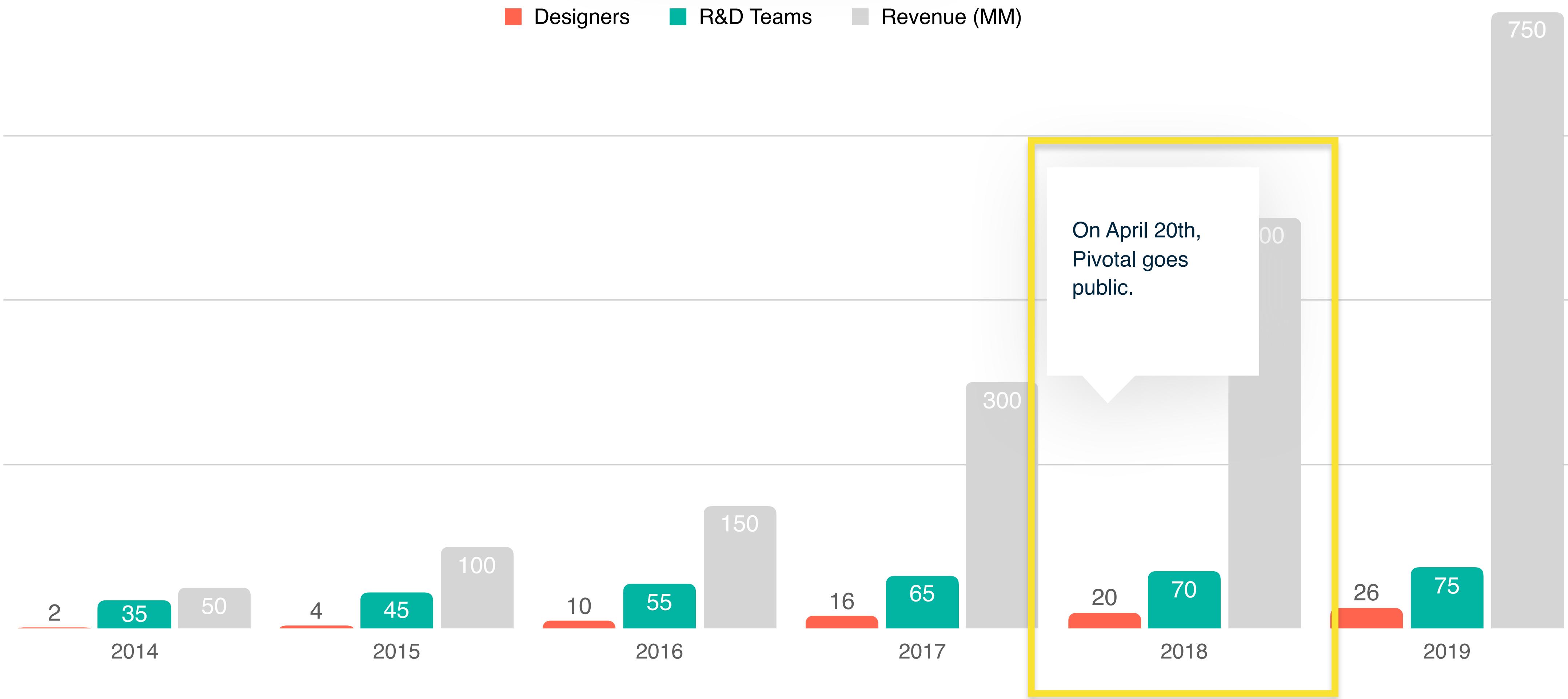
# Pivotal®



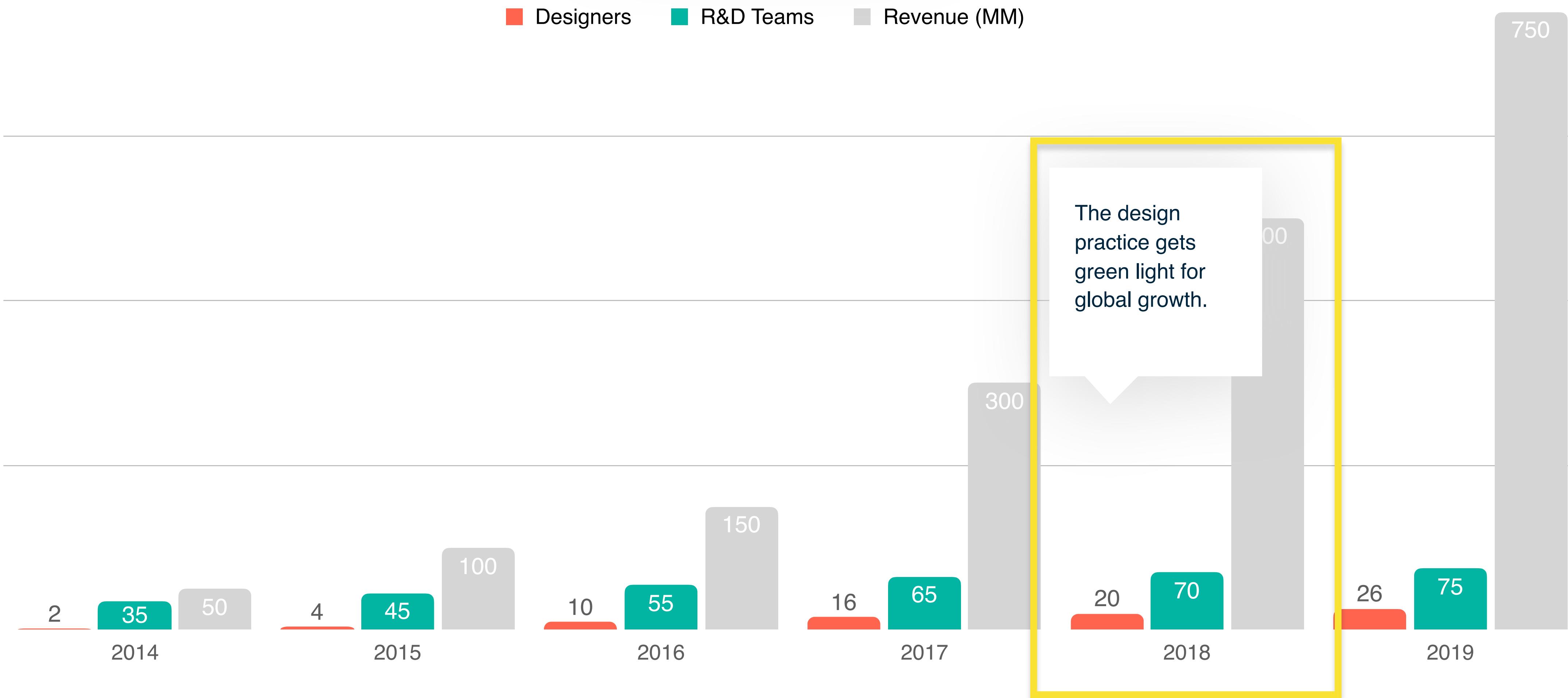
# Pivotal®



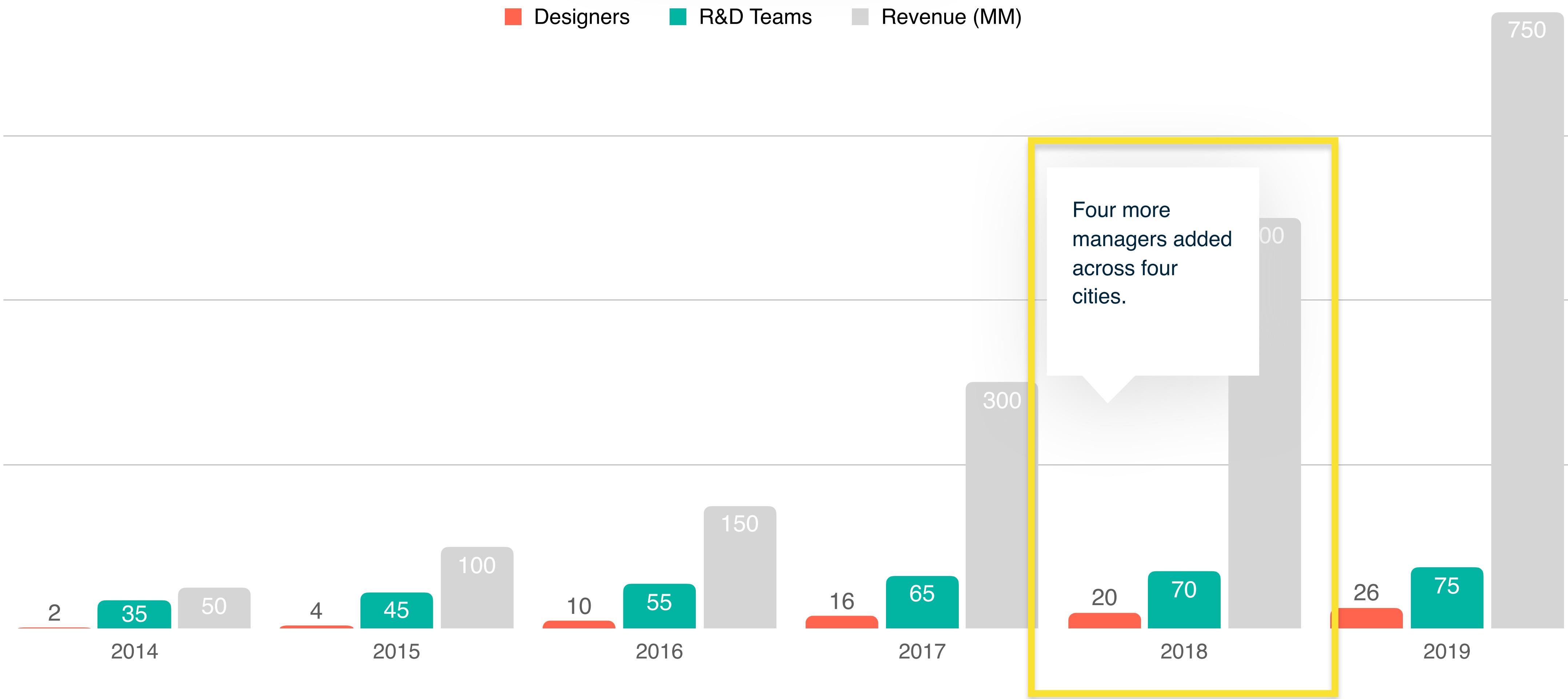
# Pivotal®



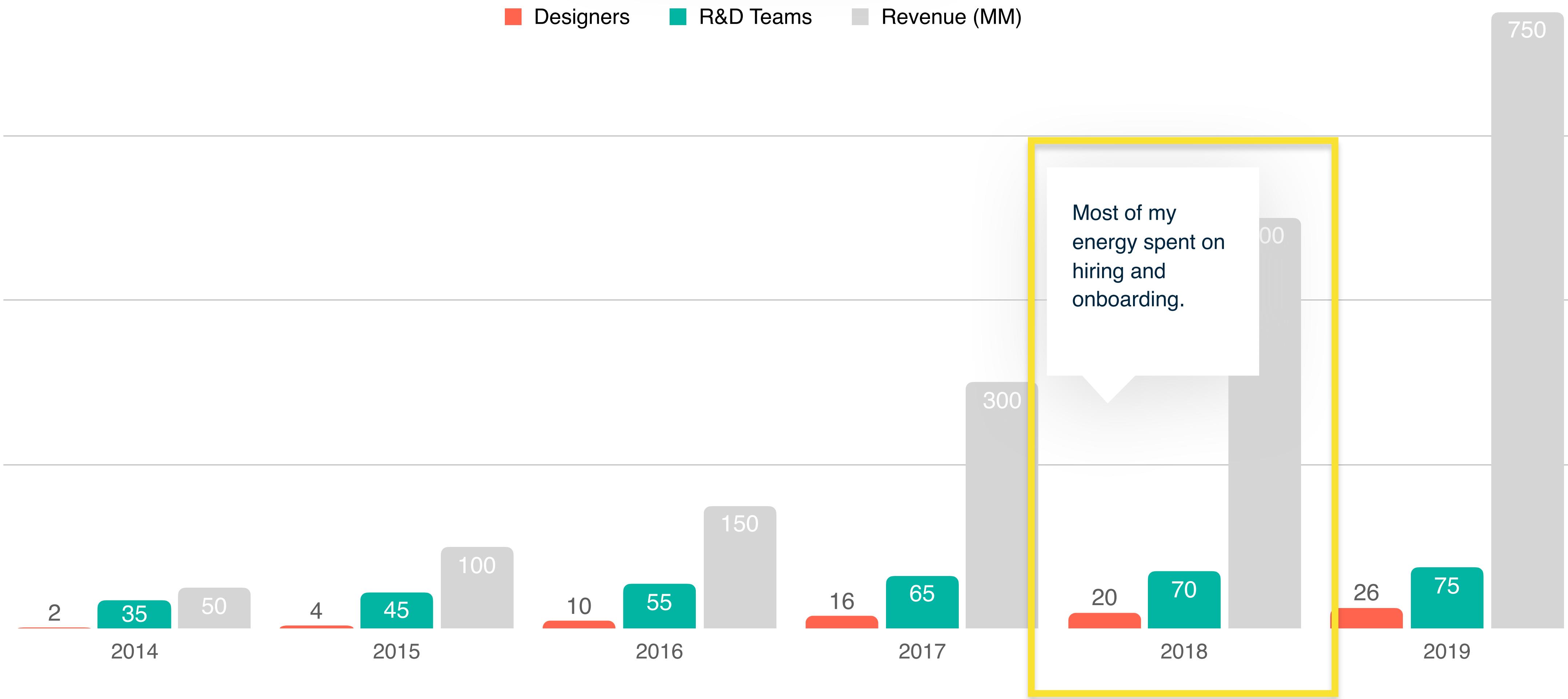
# Pivotal®



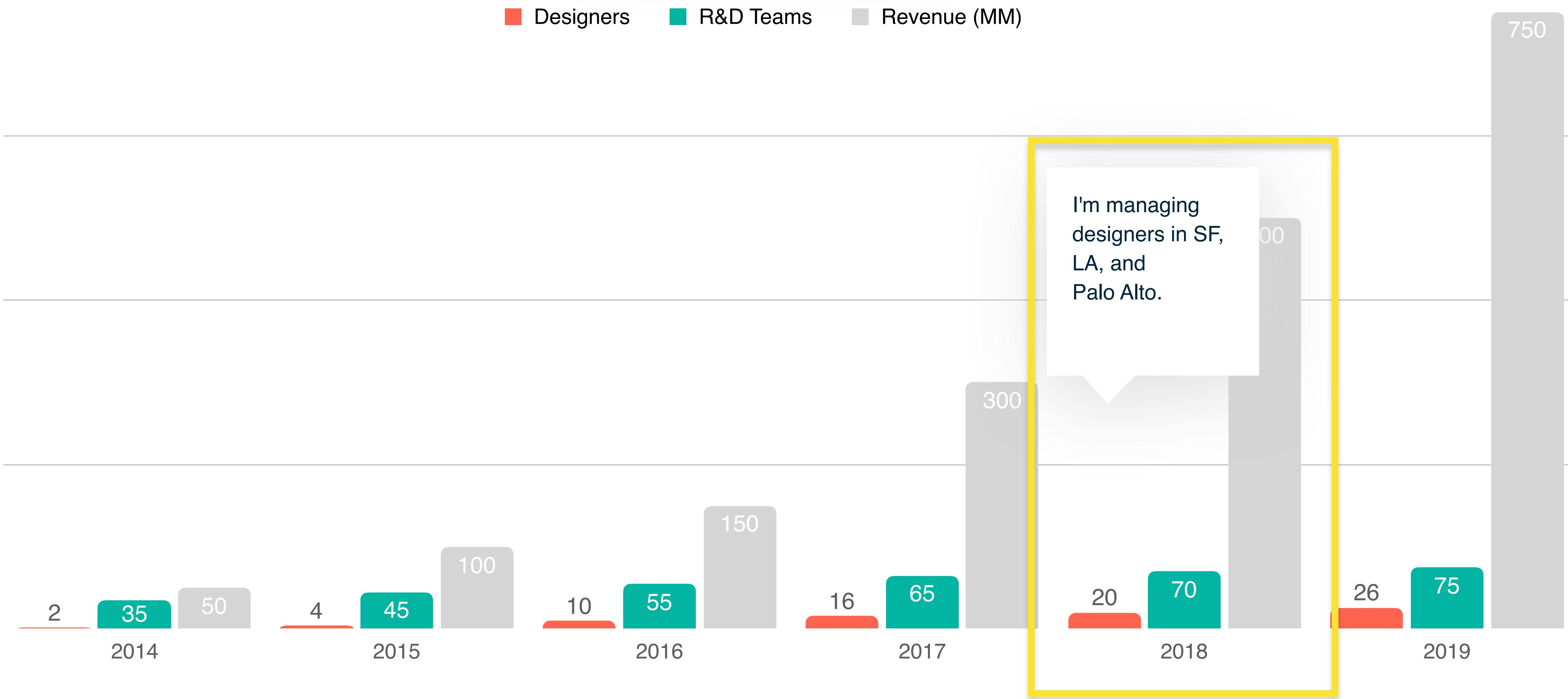
# Pivotal®



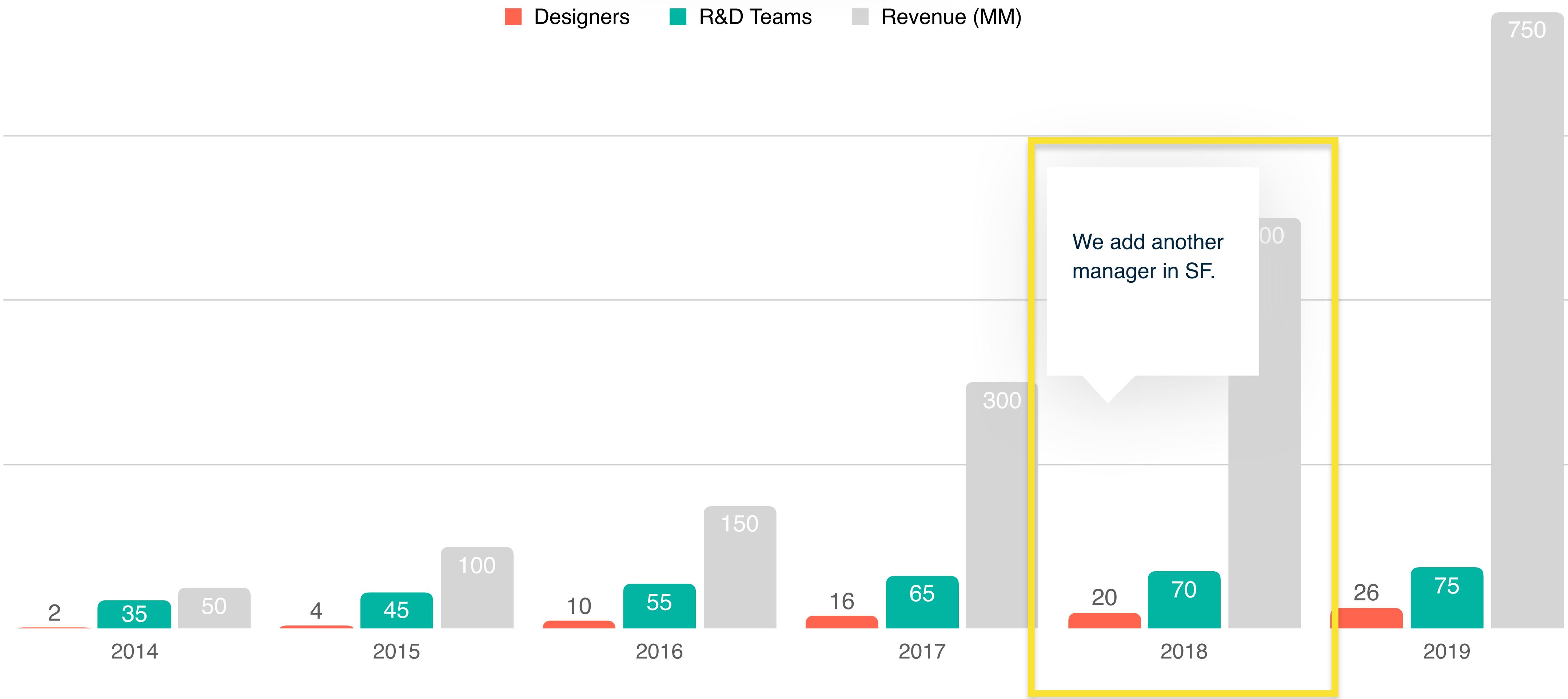
# Pivotal®



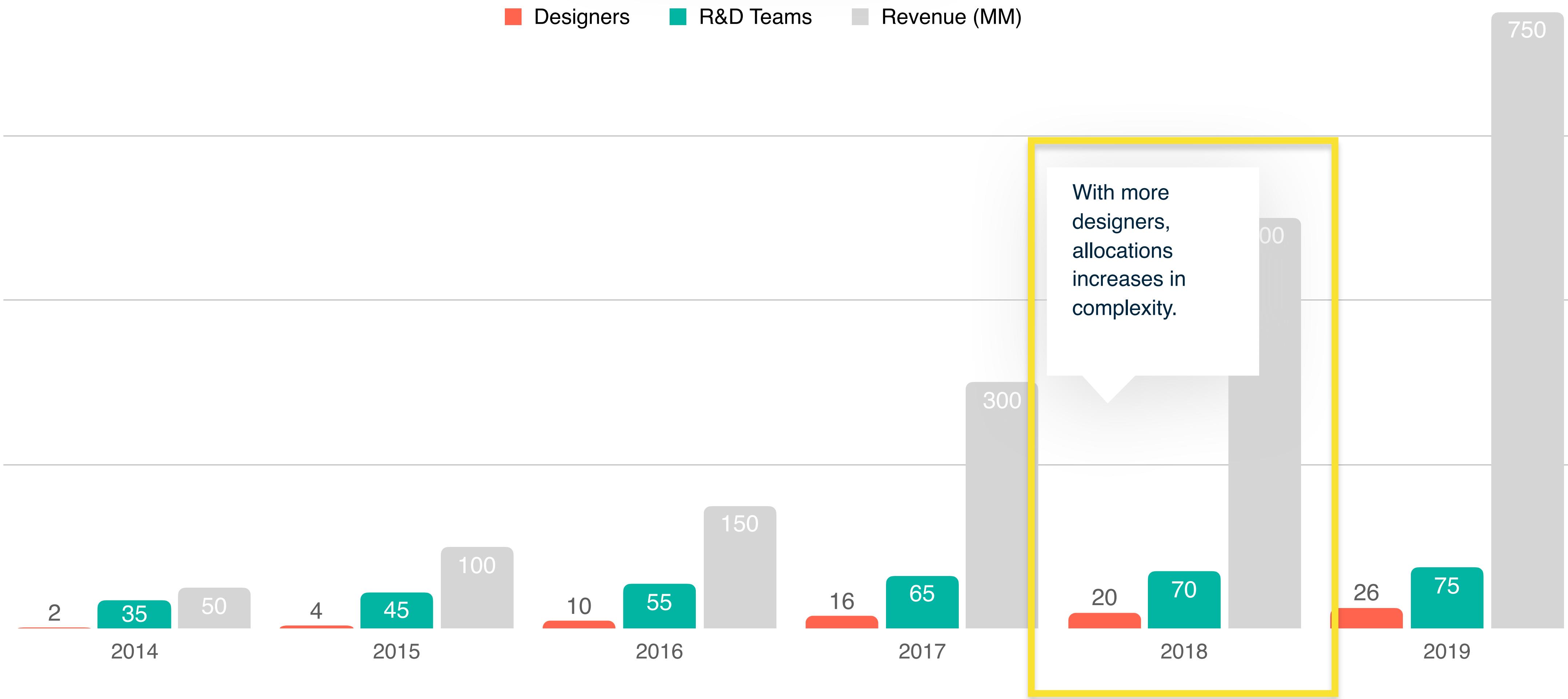
# Pivotal®



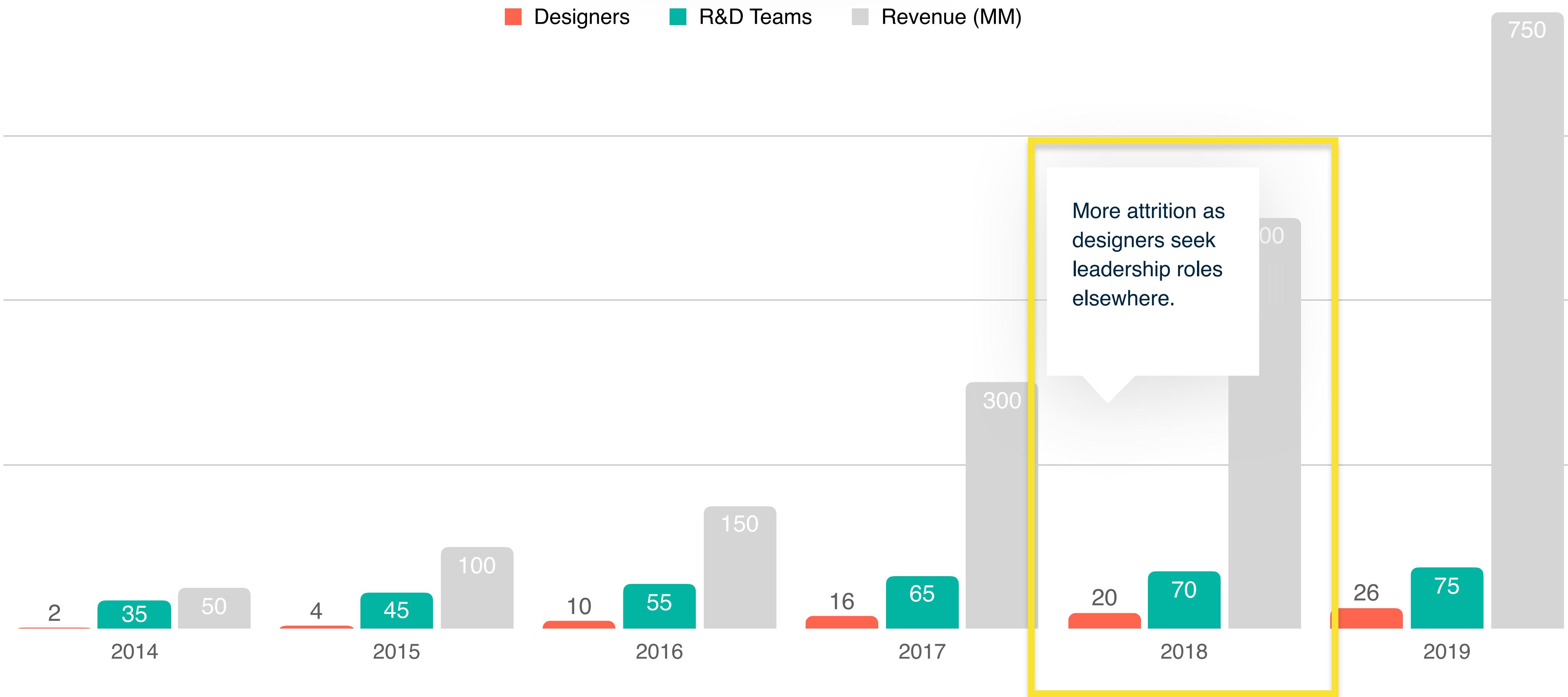
# Pivotal®



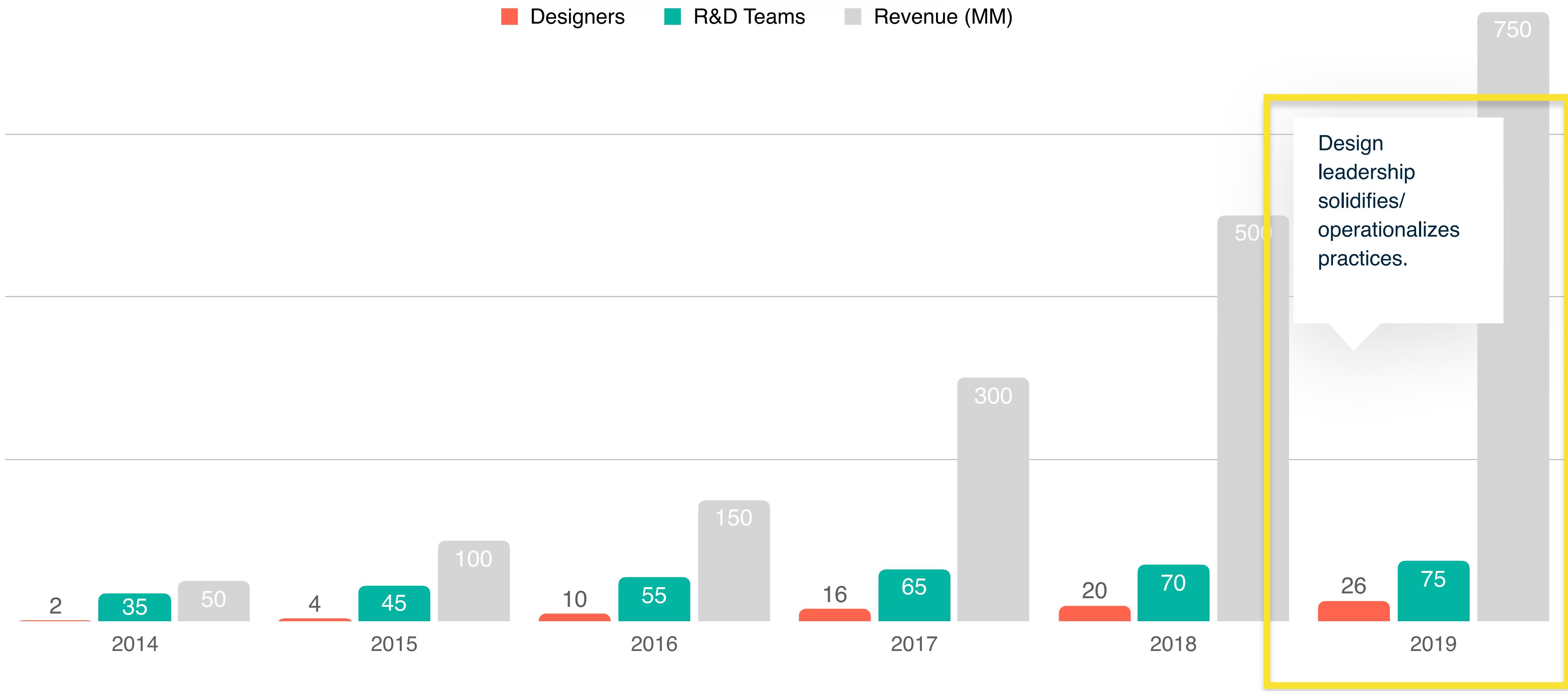
# Pivotal®



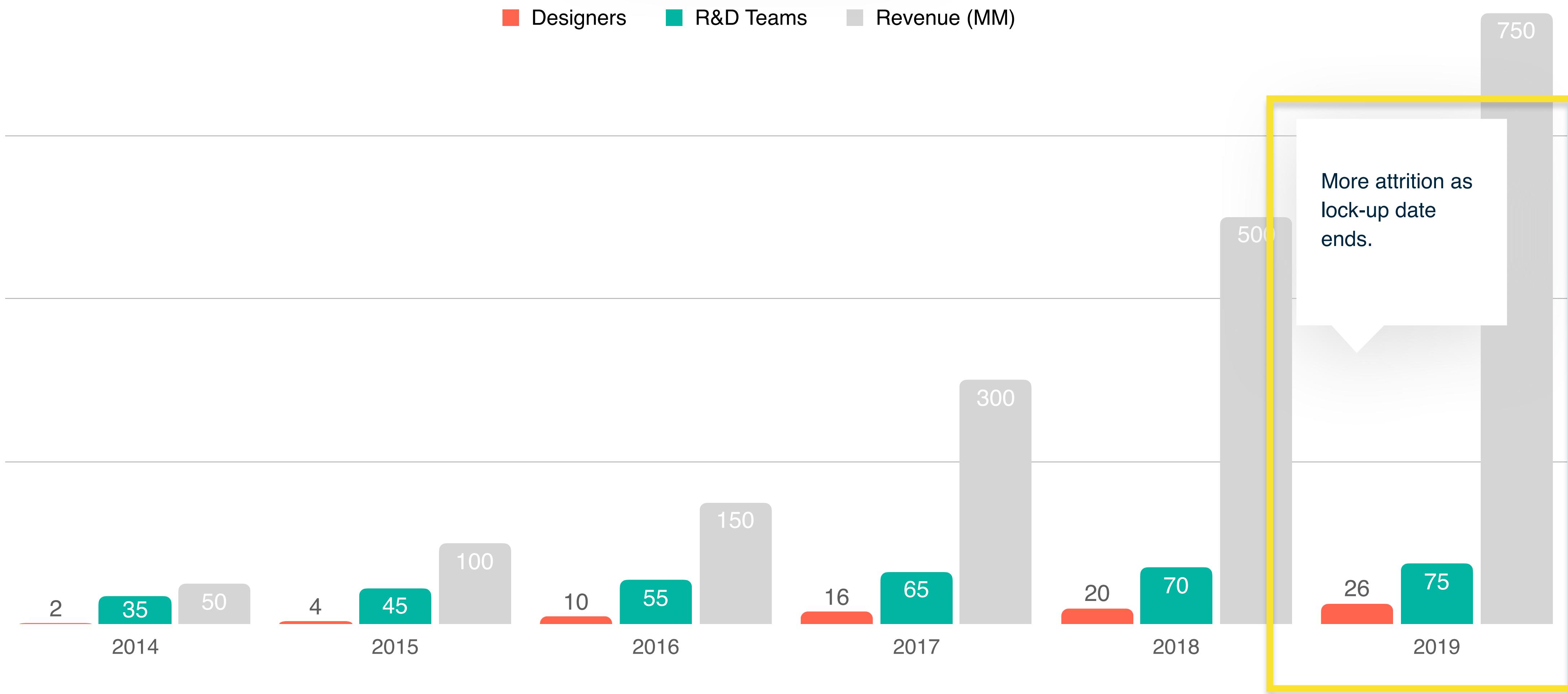
# Pivotal®



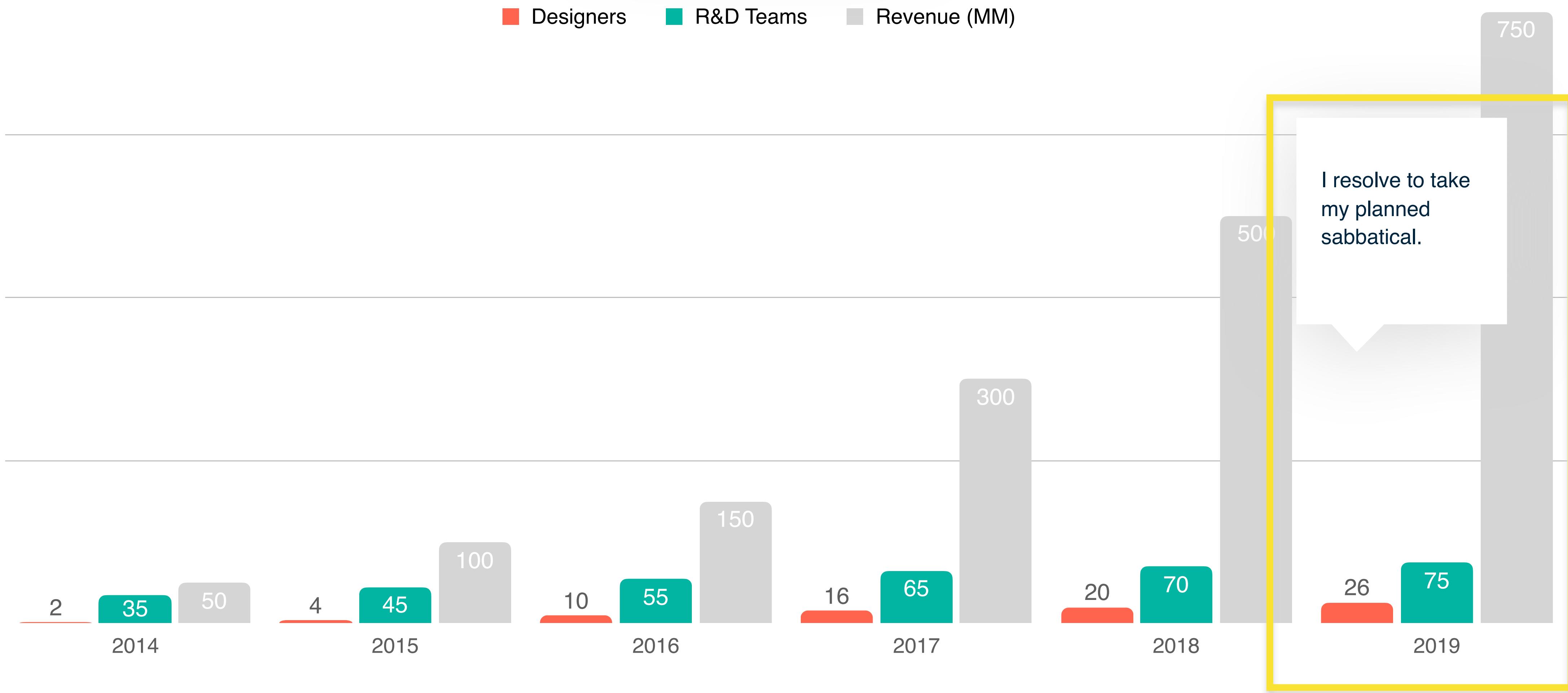
# Pivotal®



# Pivotal®



# Pivotal®



# My Values

**Questions** before **answers**

**Observations** before **evaluations**

**Change of perspective** before **point of view**

**Self-reflection** before **criticism**

# My Approach

Vision

Feedback

Growth

Reflection



KATHRYN  
2.0

## GOALS

Spend more time consciously planning my career

Refine my Lean UX process

Measure my work

Improve coaching, presenting, leadership skills

Create a system for getting things done that works for me + my team

Connect more w/coworkers

Connect with other people in the Bay Area at professional/networking events

Continue learning, e.g. Fundamentals of Human Behavior, UX writing

Move towards becoming a Sr Designer at Pivotal

Pick up side project at work - maybe colab w/ Ashley + Marion on their personal work but from bottom-up perspective

## BEHAVIORS - START/STOP/KEEP

Start / Keep - Stay organized by setting aside time each week to catch up

Stop - Getting distracted

Stop - Comparing myself to others (too much)

Keep - Stay open to new ideas

Start - pair well with others

Keep ↳ experiment + iterate w/ my process

Start - process reflection/journaling

Stop - feeling like I need to do "everything" → Set intentions for each

Start - prioritizing what needs to be done

# Vision Board



mountain



boulders



rocks

Deepen relationships with SF R&D Leadership

Improve personal network in the SF office

Increase transparency in to decision-making for how design is invested

Map out what we want to achieve in terms of the discipline and make it part of the river / the work

Come up with goals we might do to make design part of the river / the work that we can collectively own

Enter into thoughtful dialogue and consultation with members of other disciplines

Proactively schedule 1:1s with stakeholders and teammates to request feedback on what I can do to improve my interactions with them

Establish more frequent check-ins with peers and stakeholders to support reports

Co-create a decision-making framework for how design is invested w/ PLs and Dirs

Work with PLs and Dirs on a collaborative approach to W Coast allocations

# Growth Board



**Michael**, the collaborative and supportive mentor who is working on *communication*, and *psychological safety*.

## Growth opportunities

Seeking  
and  
Receiving  
Feedback

Engage  
productively  
with Feedback  
I receive

## Behaviors to adopt

Frame the work  
as a LEARNING  
PROBLEM not as  
an execution  
problem

MODEL  
CURIOSITY; ask a  
lot of QUESTIONS;  
create a necessity  
for ppl's voices  
and ideas

ACKNOWLEDGE  
my own  
FALLIBILITY

## Needs

Access to  
leadership

Collaboration  
from teams  
where I have  
reports

Continuous  
feedback  
from reports  
and peers

## Goals

Deepen  
relationships  
with SF R&D  
Leadership

Become  
versed in  
psych safety,  
nurture D&I

# Skills Progression

	I am not very confident in this area	I have some skills, but have a lot of room to learn	I am pretty comfortable in this area	I consider this area a personal strength	I got this!
Coaching & Mentorship					
Feedback					
Advocacy					
Conflict Management					
Communication					
Leading by Example					
D&I + Psychological Safety					

SCREEN SHOT

# Management Summary

Recruiting

Hiring

Onboarding

Growing

Allocating

# Management Summary

Recruiting

Hiring

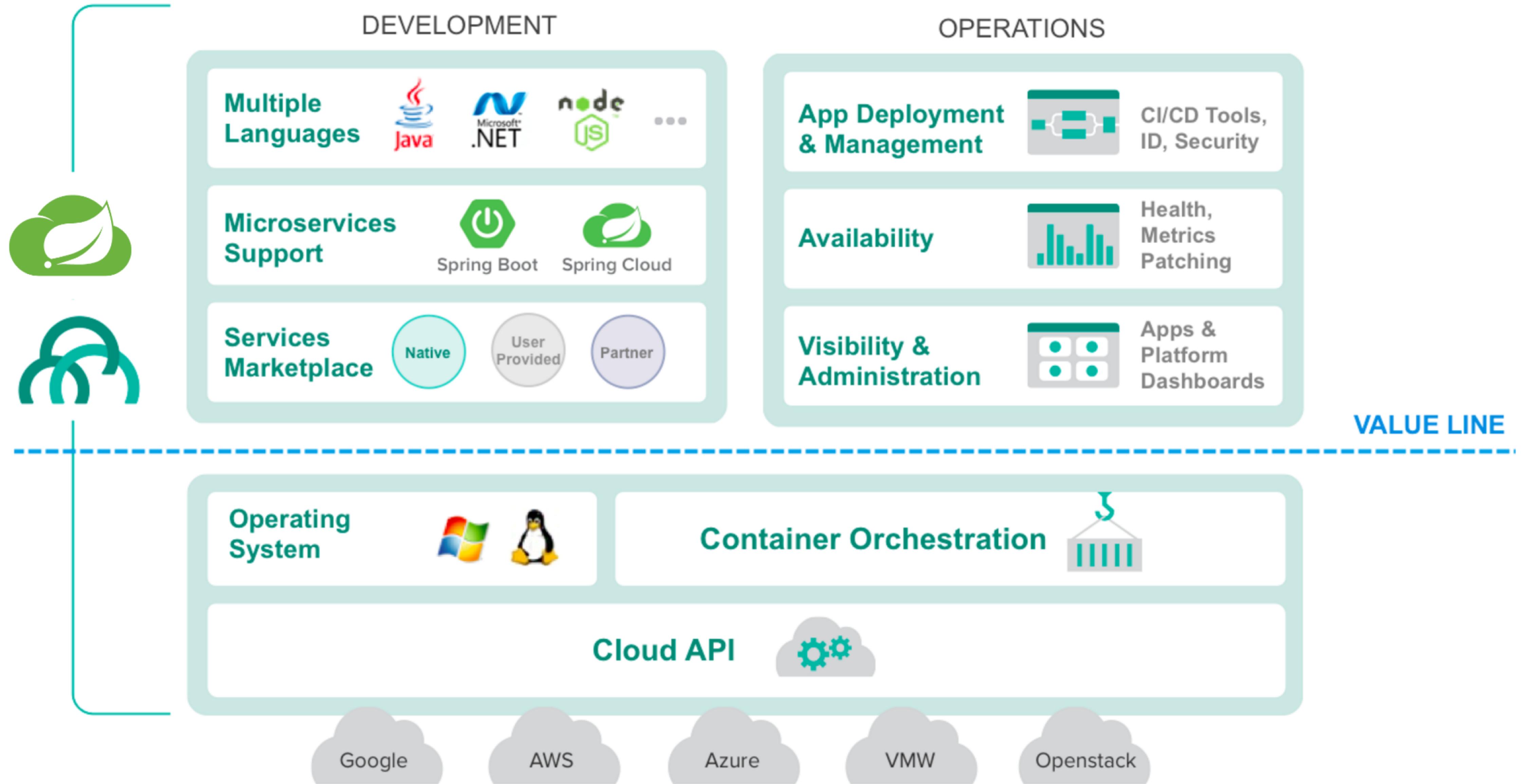
Onboarding

# Growing

Allocating

# Projects

1. User Onboarding Experience
2. Project Bootstrapping
3. Enterprise Identity Service



# Pivotal Platform

## Activities & Personas

Platform Lifecycle	Platform Security	Platform Monitoring	App Lifecycle	App Security	App Monitoring	Services Mgmt
 Platform Engineer  Platform Architect	 Platform Engineer  Security Officer	 Platform Engineer	 App Developer  App Operator	 App Developer  Security Admin	   App Developer Platform Engineer App Operator	 Service Broker Author  Platform Engineer

75 Pivotal R&D Teams + Open Source Contributors from 12 companies

# User Onboarding Experience

# User Onboarding Experience

A screenshot of a Twitter post. The profile picture of the user Francis Irving (@frabcus) is visible. His name, **Francis Irving**, is displayed in bold black text, followed by his handle @frabcus. To the right of the profile picture are two buttons: a gear icon and a "Follow" button with a person icon. The tweet itself contains two mentions in blue: "@jambay" and "@matthewcford". The main text of the tweet reads: "e.g. I just googled \"pcf cloud foundry\" and \"pcf evaluation\" and got nothing about a product I can use." The background of the image is a light gray gradient.

**Francis Irving**  
@frabcus

@jambay @matthewcford e.g. I just googled  
"pcf cloud foundry" and "pcf evaluation" and  
got nothing about a product I can use.

run.pivotal.io

Pivotal + VMware: Transforming how more of the world builds software →

Pivotal Web Services FEATURES PRICING MARKETPLACE BLOG SUPPORT DOCS LOG IN

# A Cloud Native Platform run by the experts at Pivotal

Pivotal Web Services is a modern runtime for Spring Boot, .NET\*, and Node apps

**SIGN UP FOR FREE**

\$87 of free trial credit. No credit card required.

The screenshot shows the Pivotal Web Services dashboard for the 'Example Org'. The top navigation bar includes links for Pivotal Web Services, FEATURES, PRICING, MARKETPLACE, BLOG, SUPPORT, DOCS, and LOG IN. Below the navigation is a large heading 'A Cloud Native Platform run by the experts at Pivotal' and a subtext 'Pivotal Web Services is a modern runtime for Spring Boot, .NET\*, and Node apps'. A prominent green button says 'SIGN UP FOR FREE' with the note '\$87 of free trial credit. No credit card required.' A modal window titled 'Example Org' displays organization statistics: QUOTA (22%, 2.2 GB of 10GB Limit), BILLING (Current Balance \$745.44, Next Statement Date 03/01/14), and DETAILS (\$745.44, 03/01/14). The main dashboard shows 4 Spaces, 6 Domains, and 26 Members. It lists four spaces: development, dev-hr, staging, and testing, each with app and service counts. The development space has 12 apps and 8 services, 10% of org quota. The dev-hr space has 6 apps and 14 services, 6% of org quota. The staging space has 2 apps and 2 services, 2% of org quota. The testing space is listed but not fully visible. On the left sidebar, there are links for Example Org, SPACES (development, dev-hr, staging, testing), Docs, Support, Tools, Blog, and Status.

# Persona



Jane

Enterprise Developer

# Persona



**Jane**  
Enterprise Developer

Needs to:

- Contribute to a build or buy decision in her company
- Learn what Pivotal Platform could do for her company
- Share what she learned with her colleagues

# Persona



Jane

Enterprise Developer

As Jane, I want to  
**deploy an app**  
in order to **evaluate Pivotal Platform**.

# Product Goals

- A web search for **Try Pivotal Platform** will result in finding a canonical guide.
- Success can be achieved in **15 mins or less**.

**Success = deploy** an app, make a **change**, **redeploy**.

- The getting started guide will **reveal a path** to learn **deeper concepts**.

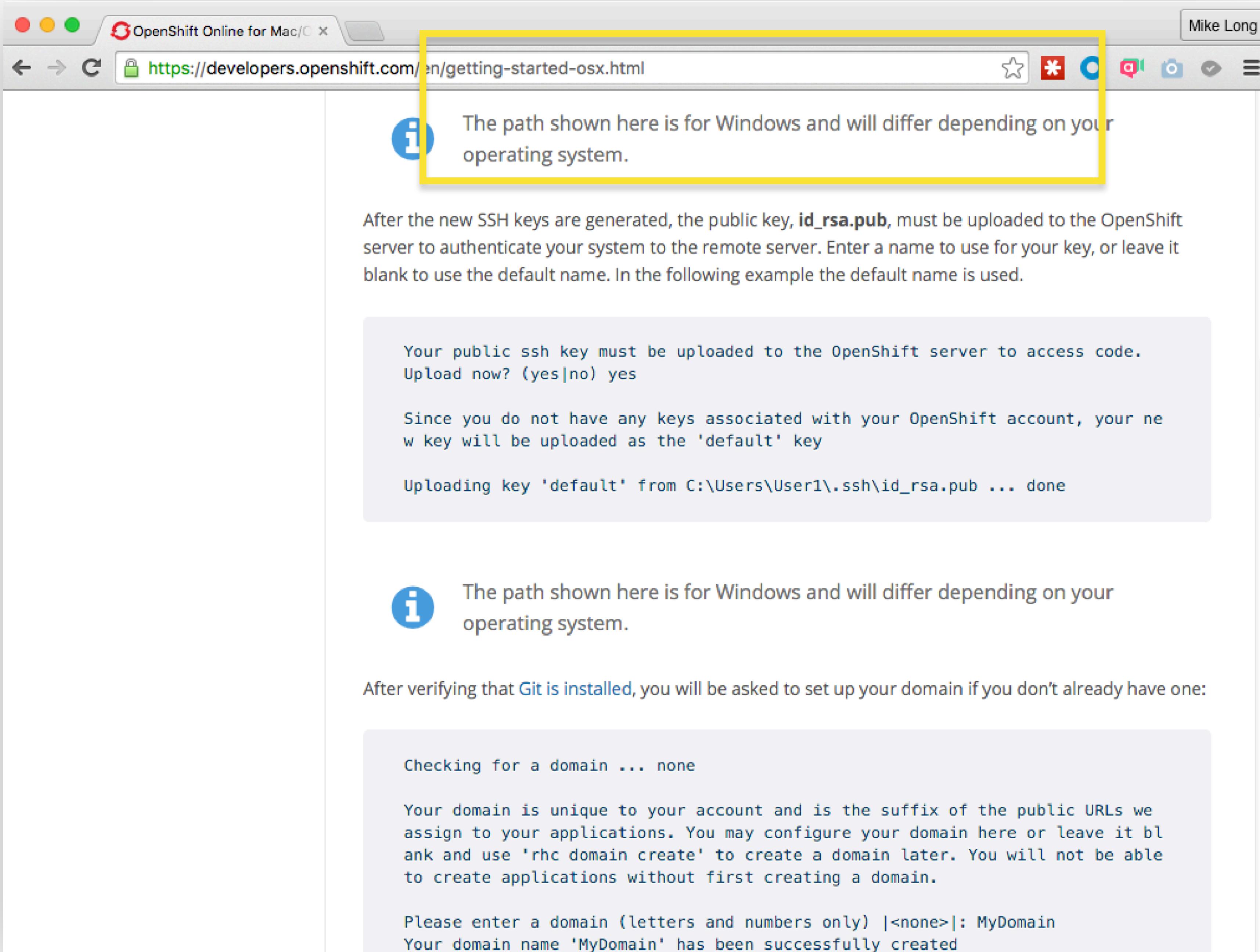
# Competitor Research

# Criteria

- A web search for **Try Red Hat OpenShift** will result in finding a canonical guide.
- Success can be achieved in **15 mins or less**.

**Success = deploy** an app, make a **change, redeploy**.

- The getting started guide will **reveal a path** to learn **deeper concepts**.

A screenshot of a Mac OS X desktop showing a web browser window titled "OpenShift Online for Mac". The URL in the address bar is "https://developers.openshift.com/en/getting-started-osx.html". The page content is about setting up SSH keys for OpenShift. A yellow box highlights a note: "The path shown here is for Windows and will differ depending on your operating system." Below this, a code block shows the terminal command "Upload now? (yes|no) yes" followed by "Since you do not have any keys associated with your OpenShift account, your new key will be uploaded as the 'default' key" and "Uploading key 'default' from C:\Users\User1\.ssh\id\_rsa.pub ... done". Another note below says "The path shown here is for Windows and will differ depending on your operating system." A final note at the bottom says "After verifying that Git is installed, you will be asked to set up your domain if you don't already have one:" followed by a code block for domain creation.

The path shown here is for Windows and will differ depending on your operating system.

After the new SSH keys are generated, the public key, **id\_rsa.pub**, must be uploaded to the OpenShift server to authenticate your system to the remote server. Enter a name to use for your key, or leave it blank to use the default name. In the following example the default name is used.

```
Your public ssh key must be uploaded to the OpenShift server to access code.  
Upload now? (yes|no) yes  
  
Since you do not have any keys associated with your OpenShift account, your new key will be uploaded as the 'default' key  
  
Uploading key 'default' from C:\Users\User1\.ssh\id_rsa.pub ... done
```

The path shown here is for Windows and will differ depending on your operating system.

After verifying that Git is installed, you will be asked to set up your domain if you don't already have one:

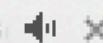
```
Checking for a domain ... none  
  
Your domain is unique to your account and is the suffix of the public URLs we assign to your applications. You may configure your domain here or leave it blank and use 'rhc domain create' to create a domain later. You will not be able to create applications without first creating a domain.  
  
Please enter a domain (letters and numbers only) |<none>|: MyDomain  
Your domain name 'MyDomain' has been successfully created
```

# Criteria

- A web search for **Try IBM Bluemix** will result in finding a canonical guide.
- Success can be achieved in **15 mins or less**.

**Success = deploy** an app, make a **change, redeploy**.

- The getting started guide will **reveal a path** to learn **deeper concepts**.

IBM developerWorks :  Mike Long

www.ibm.com/developerworks/cloud/bluemix/quick-start-bluemix.html

developerWorks Premium An all-access pass to building your next great app! [Sign up](#)

IBM Sign in | Register dW

IBM developerWorks® Technical topics Evaluation software Community Events Search developerWorks

developerWorks Premium IBM Bluemix DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY REGION: US South

IBM Bluemix - Next-Generation Cloud App Development Platform IBM developerWorks ; IBM's resource for developers and IT professionals

# IBM Bluemix

The Digital Innovation Platform

## Quick Start Guide

### 1. Explore the Platform

David Barnes, developer runtime expert

BUILD EXTEND SCALE INTEGRATE

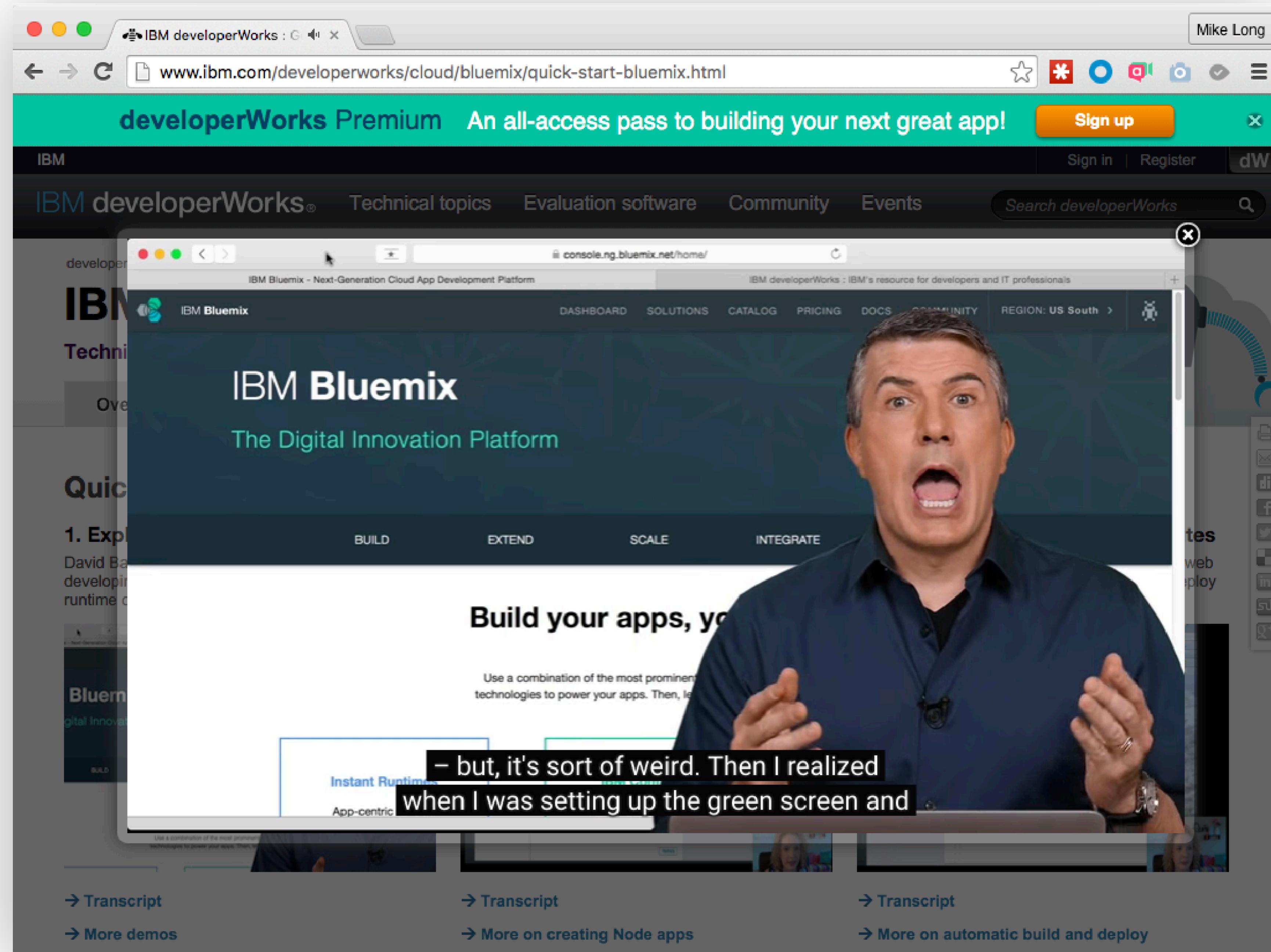
### Build your apps, your way

Use a combination of the most prominent technologies to power your apps. Then, let them scale across the cloud.

Instant Runtime App-centric

- but, it's sort of weird. Then I realized when I was setting up the green screen and

→ Transcript → Transcript → Transcript  
→ More demos → More on creating Node apps → More on automatic build and deploy



# Criteria

- A web search for **Try Heroku** will result in finding a canonical guide.
- Success can be achieved in **15 mins or less**.

**Success = deploy** an app, make a **change, redeploy**.

- The getting started guide will **reveal a path** to learn **deeper concepts**.

A screenshot of a web browser window titled "Getting Started" with a yellow box highlighting the address bar which contains "localhost:5000". The browser has three tabs open: "Getting Started", "Personal apps | Heroku", and "Getting Started". The main content area features a red coffee cup icon and the title "Getting Started with Gradle on Heroku". Below the title is a descriptive paragraph about the app being a sample Gradle application. At the bottom, there are two buttons: "Getting Started with Gradle" and "Source on GitHub". A small note at the bottom of the page encourages users to learn more about the Heroku platform.

localhost:5000

Mike Long

Home How Heroku Works Getting Started Guides Heroku Dev Center

# Getting Started with Gradle on Heroku

This is a sample Gradle application deployed to Heroku. It's a reasonably simple app - but a good foundation for understanding how to get the most out of the Heroku platform.

⚡ Getting Started with Gradle ⚡ Source on GitHub

To deploy your own copy, and learn the fundamentals of the Heroku platform, head over to the [Getting Started with Gradle on Heroku](#) tutorial.

# Criteria

- A web search for **Try Pivotal Platform** will result in finding a canonical guide.
- Success can be achieved in **15 mins or less**.

**Success = deploy** an app, make a **change, redeploy**.

- The getting started guide will **reveal a path** to learn **deeper concepts**.

Getting Started with Pivotal

https://docs.pivotal.io/pivotalcf/getstarted/

Pivotal Documentation v1.6.0.0

Back to Product Page | Support |

# Getting Started with Pivotal Cloud Foundry®

Welcome to [Pivotal Cloud Foundry®](#)!

The following IaaS-specific guides are intended to walk you through the process of getting your Pivotal Cloud Foundry® (PCF) deployment up and running.

If you experience a problem while following the steps below, check the [Known Issues](#), or refer to the [PCF Troubleshooting Guide](#).

Once you have completed the steps in this guide, explore the documentation on [docs.pivotal.io](#) to learn more about [Pivotal Cloud Foundry®](#) and the Pivotal product suite.

- [Installing Pivotal Cloud Foundry® on AWS](#)
- [Installing Pivotal Cloud Foundry® on OpenStack](#)
- [Installing Pivotal Cloud Foundry® on vSphere and vCloud Air](#)

Pivotal Documentation © 2016 Pivotal Software, Inc. All Rights Reserved.

Need help? Visit Support

# Prototype: v1

P Getting Started https://pivotal.io/getting-started-with-pcf

PIVOTAL NETWORK

CONTACT

EVENTS

CONTACT

EVENTS

Line Interface (CLI)

the CF CLI works:

apps deployed to Pivotal Cloud Foundry

connect with us:   

Does it work? If so, you are now ready to start building!

First, sign in to PWS with the CF CLI:

```
$ cf login -a api.run.pivotal
```

The image consists of three vertically aligned screenshots of the Pivotal website. The left screenshot shows the homepage with a large 'GET STARTED' button. The middle screenshot shows the 'Getting Started with Pivotal Cloud Foundry' guide, which includes an 'Introduction' section and a 'CF CLI' setup section. The right screenshot shows the 'Events' page. All screenshots feature the Pivotal logo and navigation bar at the top.

**Getting Started**

<https://pivotal.io/getting-started-with-pcf>

Pivotal

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

# Getting Started with Pivotal Cloud Foundry

Introduction

Install the CF CLI

Deploy the sample app

Scale the app

View logs

Connect a database

Next steps

with Pivotal Cloud Foundry

## Introduction

This guide will help you get an application up and running in a matter of minutes.

This guide requires that [Java 8 or higher](#) is installed and a [Pivotal Web Services](#) (PWS) account. PWS is a public cloud service maintained by Pivotal.

[Get your free Pivotal Web Services account](#)

I'M GOOD TO GO >

## Install the CF CLI

In this step you will install the Cloud Foundry Command Line Interface (CLI).

[DOWNLOAD CF CLI FOR MAC OS X](#)

Is it installed? Try the following command to test that the CF CLI works:

```
$ cf help
```

You can use the CF CLI to perform all commands on apps deployed to Pivotal Cloud Foundry .

LET'S KEEP GOING >

connect with us:

...  
The server is running at <http://localhost:8080/spring-music>

Does it work? If so, you are now ready to sign in to PWS so you can "push" your app.

First, sign in to PWS with the CF CLI:

```
$ cf login -a api.run.pivotal.io
```

Events

Contact

Events

BLOG RESOURCES PIVOTAL NETWORK

BLOG RESOURCES PIVOTAL NETWORK

BLOG RESOURCES

**Getting Started with Pivotal Cloud Foundry**

**Install the CF CLI**

In this step you will install the Cloud Foundry Command Line Interface (CLI).

**DOWNLOAD CF CLI FOR MAC OS X**

Is it installed? Try the following command to test the installation:

```
$ cf help
```

You can use the CF CLI to perform all commands on your application.

**LET'S KEEP GOING >**

**Getting Started with Pivotal Cloud Foundry**

**Deploy the sample app**

Now that you have the CF CLI installed and a PWS ("pdubs") account, you are sooo close to deploying an app!

Download the sample app with Git:

```
$ git clone https://github.com/cloudfoundry-samples/spring-music
```

If you don't have Git installed, you can download a zip file of the sample app here: [github.com/cloudfoundry-samples/spring-music/archive/master.zip](https://github.com/cloudfoundry-samples/spring-music/archive/master.zip)

Navigate into the directory:

```
$ cd spring-music
```

Start up the application locally: (Windows folks should use the `gradlew.bat` file)

```
$ ./gradlew tomcatRun  
...  
The server is running at http://localhost:8080/spring-music
```

Does it work? If so, you are now ready to sign in to PWS so you can "push" your app.

First, sign in to PWS with the CF CLI:

```
$ cf login -a api.run.pivotal.io
```

The CF CLI will have guided you through signing in with your credentials as well as targeting an Org and a Space to deploy your app into.

Getting Started with Pivotal Cloud Foundry

## Scale the app

Increasing the available disk space or memory can improve overall app performance. Similarly, running additional instances of an app can allow the application to handle increases in user load and concurrent requests. These adjustments are called **scaling**.

Scaling your app **horizontally** creates or destroys app instances. Adding more instances allows your application to handle increased traffic and demand.

You can increase app instances with the CF CLI:

```
$ cf scale spring-music -i 2
```

Check the status of your app and verify there are now two running instances:

```
$ cf app spring-music
```

Scaling your app **vertically** changes the disk space limit or memory limit for all app instances.

You can increase app instance memory limit with the CF CLI:

```
$ cf scale spring-music -m 1G
```

You can also increase app instance disk limit with the CF CLI:

```
$ cf scale spring-music -k 512M
```

LET'S KEEP GOING >

**Getting Started**

<https://pivotal.io/getting-started-with-pcf>

Pivotal

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

# Getting Started with Pivotal Cloud Foundry

## View logs

Introduction

Install the CF CLI

Deploy the sample app

Scale the app

**View logs**

Connect a database

Next steps

Pivotal Cloud Foundry provides a stream of log output from your application and from system components that interact with your app during updates and execution.

Every log line contains four fields: timestamp, Log type, channel, and message.

You can stream logs through the CF CLI:

```
$ cf logs spring-music
```

Press Control + C to stop log streaming.

You can also view a dump of the most recent logs with the CF CLI:

```
$ cf logs spring-music --recent
```

**LET'S KEEP GOING >**

connect with us: [YouTube](#) [Twitter](#) [Email](#)

Restage the app so the new service is detected:

```
$ cf restage spring-music
```

Verify the new service is in fact bound to your app:

```
$ cf bind-service spring-music-db
```

with Pivotal Cloud Foundry

## Scale the app

Increasing the available disk space or memory can improve your application's performance. Similarly, running additional instances of an app can accommodate increases in user load and concurrent requests. These are called scaling operations.

Scaling your app **horizontally** creates or destroys additional instances of your application. This allows your application to handle increased traffic.

You can increase app instances with the CF CLI:

```
$ cf scale spring-music -i 2
```

Check the status of your app and verify there are now two instances:

```
$ cf app spring-music
```

Scaling your app **vertically** changes the disk space allocated to each instance.

You can increase app instance memory limit with the CF CLI:

```
$ cf scale spring-music -m 1G
```

You can also increase app instance disk limit with the CF CLI:

```
$ cf scale spring-music -k 512M
```

**Getting Started with Pivotal Cloud Foundry**

**View logs**

Pivotal Cloud Foundry provides a stream of log output from system components that interact with your app during runtime.

Every log line contains four fields: timestamp, Log type, Application name, and Log message.

You can stream logs through the CF CLI:

```
$ cf logs spring-music
```

Press Control + C to stop log streaming.

You can also view a dump of the most recent logs with:

```
$ cf logs spring-music --recent
```

**Connect a database**

Introduction

Install the CF CLI

Deploy the sample app

Scale the app

View logs

View the available services in the Marketplace:

```
$ cf marketplace
```

View the available plans a service provides:

```
$ cf marketplace -s elephantsql
```

Create a service instance that you can bind to your app:

```
$ cf create-service elephantsql turtle spring-music-db
```

Bind the newly created service to your app:

```
$ cf bind-service spring-music spring-music-db
```

Restage the app so the new service is detected:

```
$ cf restage spring-music
```

Verify the new service is in fact bound to your app:

```
$ cf services
```

**LET'S KEEP GOING >**

**CONNECT WITH US:**

Getting Started

https://pivotal.io/getting-started-with-pcf

Pivotal

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

# Getting Started with Pivotal Cloud Foundry

## Next steps

Introduction

Thank you for familiarizing yourself with the basic concepts of Pivotal Cloud Foundry.

Install the CF CLI

Deploy the sample app

Scale the app

View logs

Connect a database

How PCF Works <http://docs.pivotal.io/pivotalcf/concepts/>

PCF Workstation <https://network.pivotal.io/products/pcf-dev>

Spring Guides <https://spring.io/guides>

### Next steps

\$ cf marketplace

\$ cf marketplace -s elephantsql

\$ cf create-service elephantsql turtle spring-music-db

\$ cf bind-service spring-music spring-music-db

\$ cf restage spring-music

Restage the app so the new service is detected:

Verify the new service is in fact bound to your app:

connect with us:

Prototype: v2

P Getting Started https://pivotal.io/getting-started-with-pcf

PIVOTAL NETWORK

CONTACT

EVENTS

CONTACT

EVENTS

Cloud Foundry

# Getting Started with Pivotal Cloud Foundry

15-minute tutorial for learning app deployment concepts

up and running on PCF. To , an instance of PCF hosted by

Line Interface (cf CL

works:

station

on apps deployed to PC

Connect with us!

GET STARTED

connect with us:   

connect with us:   

connect with u

```
$ cf login -a api.run.p...  
...  
API endpoint: https://e...  
User: email@e...
```

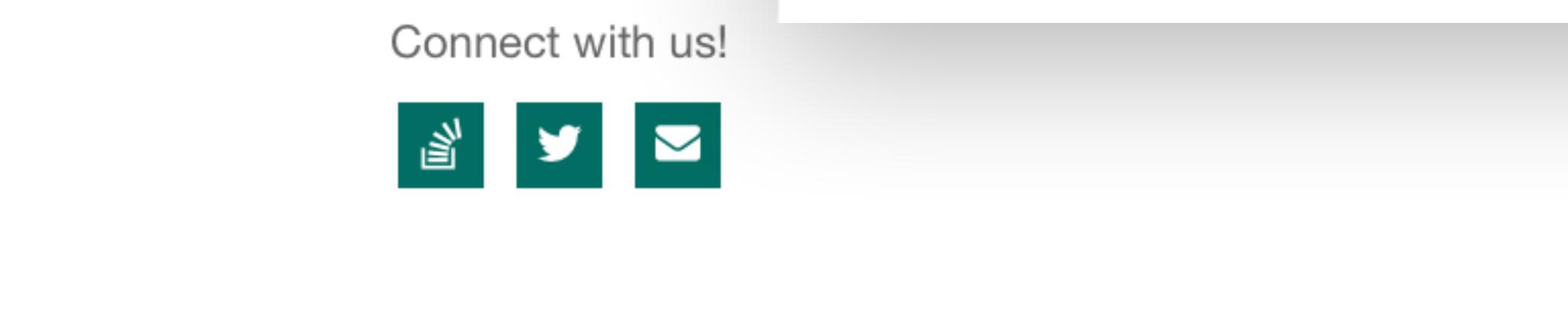
The screenshot shows a web browser window with three tabs open. The active tab is titled 'Getting Started' and has the URL <https://pivotal.io/getting-started-with-pcf>. The browser interface includes standard controls like back, forward, and search.

The main content area displays the 'Getting Started with Pivotal Cloud Foundry' page. It features a sidebar on the left with links to 'Introduction', 'Install the CF CLI', 'Deploy the sample app', 'View the logs', 'Connect a database', 'Scale the app', and 'Next steps'. A large 'GET STARTED' button is prominently displayed. The main body contains an 'Introduction' section with text about using Pivotal Web Services and a list of requirements:

- a free [Pivotal Web Services](#) account
- [Java 7 or higher](#) is installed on your workstation

A central call-to-action button says 'I'M READY TO CONTINUE'.

At the bottom, there's a 'connect with us:' section featuring social media icons for GitHub, Twitter, and Email.



```
$ cf login -a api.run.pivotal.io  
...  
API endpoint: https://api.run.pivotal.io  
User: email@example.com
```

Getting Started with Pivotal Cloud Foundry

Introduction

Install the CF CLI

Deploy the sample app

View the logs

Connect a database

Scale the app

Next steps

Install the CF CLI

Download and install the Cloud Foundry Command Line Interface (cf CLI):

DOWNLOAD FOR MAC OS X 64 BIT

Try the following command to test that the cf CLI works:

```
$ cf help
```

You can use the cf CLI to perform all commands on apps deployed to PCF.

LET'S KEEP GOING

I'M READY TO CONTINUE

connect with us: [link](#) [link](#) [link](#)

Sign in to PWS:

```
$ cf login -a api.run.pivotal.io
...
API endpoint: https://api.run.pivotal.io
User: email@example.com
```

**Getting Started**

<https://pivotal.io/getting-started-with-pcf>

**Pivotal**

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

## Getting Started with Pivotal Cloud Foundry

Introduction Deploy the sample app

Install the CF CLI Now that you have the cf CLI installed and a Pivotal Web Services (PWS) account, you are really close to deploying the sample Java app.

Deploy the sample app This sample app is built with Spring Framework and helps to demonstrate the use of database services on PCF.

View the logs

Connect a database

Scale the app

Next steps

Try the following command to test that your installation is working correctly:

```
$ cf help
```

You can use the cf CLI to perform all commands required to manage your application on PCF.

**LET'S KEEP GOING**

**Deploy the sample app**

Now that you have the cf CLI installed and a Pivotal Web Services (PWS) account, you are really close to deploying the sample Java app.

This sample app is built with Spring Framework and helps to demonstrate the use of database services on PCF.

**Download the app with git:**

```
$ git clone https://github.com/cloudfoundry-samples/spring-music
```

If you don't have Git installed, you can download a zip file of the app at [github.com/cloudfoundry-samples/spring-music/archive/master.zip](https://github.com/cloudfoundry-samples/spring-music/archive/master.zip).

**Navigate to the app directory:**

```
$ cd spring-music
```

**Sign in to PWS:**

```
$ cf login -a api.run.pivotal.io
...
API endpoint: https://api.run.pivotal.io
User: email@example.com
Org: your-org
Space: development
```

**Info icon**

Profiles: cloud,in-memory

Database type

Are You Experienced?

Getting Started

https://pivotal.io/getting-started-with-pcf

Pivotal

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

# Getting Started with Pivotal Cloud Foundry

Introduction

Install the CF CLI

Deploy the sample app

**View the logs**

Connect a database

Scale the app

Next steps

with Pivotal Cloud Foundry

Deploy the sample app

Now that you have the cf CLI installed and signed in to your account, you are really close to deploying your first application.

This sample app is built with Spring Framework and uses MongoDB as its primary in-memory database.

Download the app with git:

```
$ git clone https://github.com/cloudfoundry/spring-music
```

If you don't have Git installed, you can download it from [github.com/cloudfoundry-samples/spring-music](https://github.com/cloudfoundry-samples/spring-music).

Navigate to the app directory:

```
$ cd spring-music
```

Sign in to PWS:

```
$ cf login -a api.run.pivotal.io
...
API endpoint: https://api.run.pivotal.io
User: email@example.com
```

View the logs

PCF provides access to an aggregated view of logs related to your application. This includes HTTP access logs, as well as output from app operations such as scaling, restarting, and restaging.

Every log line contains four fields:

- Timestamp
- Log type
- Channel
- Message

Timestamp	Log type	Channel	Message
2016-03-01T16:41:01.18-0800	[RTR/4]	OUT	cf-spring-RANDOM-WORD.cfapps.io
3:50:02.244 +0000]	"GET /pui-3.0.0/font-awesome-webfont.woff2?v=4.40		HTTP/1.1" 200 0 64464 "http://cf-spring.cfapps.io/pui-3.0.0/pivotal-ui.min.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36" 10.10.66.236:36629 x_forwarded_for:"209.234.137.222" x_forwarded_proto:"http" vcap_request_id:3d4a38c-e183-484b-6564-4f6dbbf73751 response_time:0.004753977 app_id:f5a143fd-b3ea-72a71e75e2e5

View a snapshot of recent logs:

```
$ cf logs spring-music --recent
```

Stream live logs:

```
$ cf logs spring-music
```

Info icon

Profiles: cloud,in-memory

Database type

You Experienced?

base to a Postgres

an improve overall app  
ces of an app can allow an app  
requests. These adjustments  
top instances. Adding more  
ased traffic and demand.

6 8

Getting Started

https://pivotal.io/getting-started-with-pcf

Pivotal

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

## Getting Started with Pivotal Cloud Foundry

Introduction

Install the CF CLI

Deploy the sample app

View the logs

Connect a database

Scale the app

Next steps

with Pivotal Cloud Foundry

**View the logs**

PCF provides access to an aggregated log viewer. This includes HTTP access logs, as well as logs from scaling, restarting, and restaging.

Every log line contains four fields:

- Timestamp
- Log type
- Channel
- Message

Timestamp	Log type	Channel	Message
2016-03-18T16:50:01.91-07:00 [RTR/3]			
3:50:01.909 +0000] "GET /images/cloud-			
f-spring.cfapps.io/" "Mozilla/5.0 (Mac			
t/537.36 (KHTML, like Gecko) Chrome/50			
739 x_forwarded_for:"209.234.137.222"			
2d4a38c-e183-484b-6564-4f6dbbf73751 re			
eb-481e-9ea-72a71e75e2e5			
<b>2016-03-01T16:41:01.18-08:00 [RTR/4]</b>			
3:50:02.244 +0000] "GET /pui-3.0.0/fon			
0 HTTP/1.1" 200 0 64464 "http://cf-spr			
" "Mozilla/5.0 (Macintosh; Intel Mac O			
ke Gecko) Chrome/50.0.2661.102 Safari/			
"U200-234-137-222" x_forwarded_for:"			

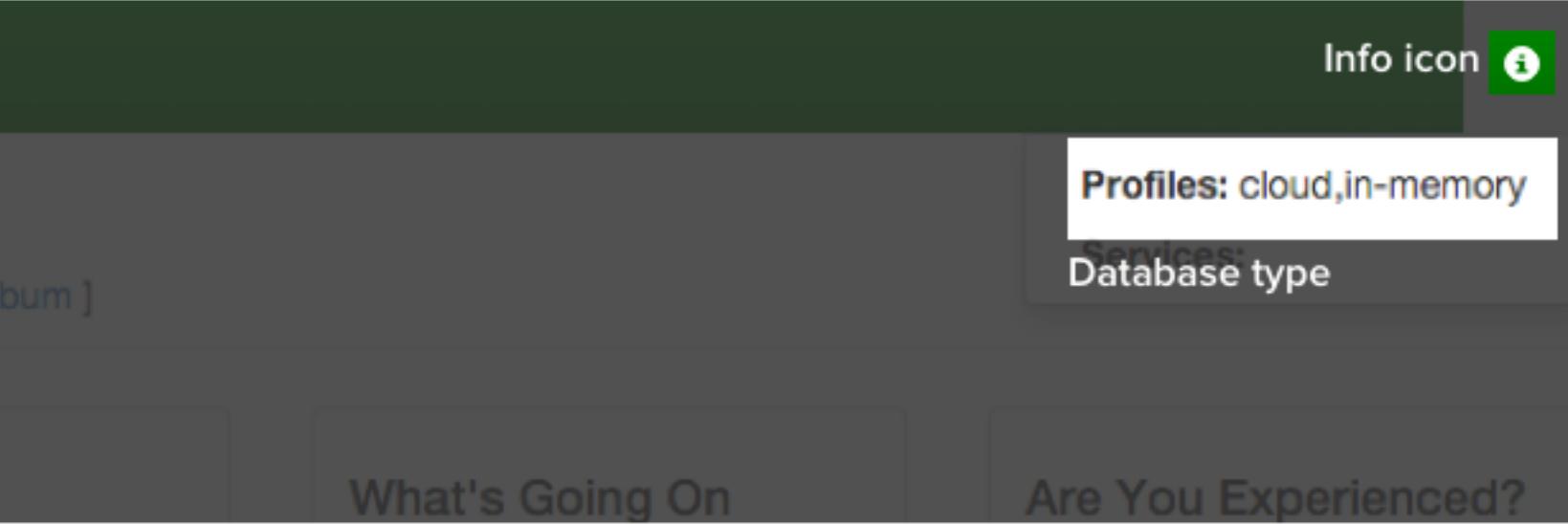
**View a snapshot of recent logs:**

```
$ cf logs spring-music --recent
```

**Connect a database**

If a database isn't available, the sample app uses a temporary in-memory database. This app supports MySQL, Postgres, Redis, and MongoDB.

You can see what type of database this app is using by clicking the info icon in the upper right corner of your app.



**What's Going On**

**Are You Experienced?**

**Profiles: cloud,in-memory**

**Database type**

Let's switch this app away from using the in-memory database to a Postgres database.

List what plans are available:

```
$ cf marketplace -s elephantsql
```

Create a database service instance for the app with the free plan:

```
$ cf create-service elephantsql turtle my-spring-db
```

ENTS

BLOG RESOURCES PIVOTAL NETWORK

CONTACT

EVENTS

Events

overall app

an app can allow an app

tests. These adjustments

stances. Adding more

traffic and demand.

PCF)

natcher\_rails

natcher\_sails

8

Getting Started

https://pivotal.io/getting-started-with-pcf

PIVOTAL NETWORK

Pivotal

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

# Getting Started with Pivotal Cloud Foundry

Introduction

Install the CF CLI

Deploy the sample app

View the logs

Connect a database

Scale the app

Next steps

with Pivotal Cloud Foundry

Connect a database

If a database isn't available, the sample database. This app supports MySQL, P

You can see what type of database thi the upper right corner of your app.

album ]

What's Going On

Let's switch this app away from using t database.

List what plans are available:

```
$ cf marketplace -s elephantsql
```

Scale the app

Increasing the available disk space or memory can improve overall app performance. Similarly, running additional instances of an app can allow an app to handle increases in user load and concurrent requests. These adjustments are called scaling.

Scaling your app *horizontally* adds or removes app instances. Adding more instances allows your application to handle increased traffic and demand.

Horizontal scaling increases the number of app instances to handle increased demand.

PCF)

natcher\_rails

natcher\_sails

Getting Started

https://pivotal.io/getting-started-with-pcf

Pivotal

PLATFORM DATA LABS CUSTOMERS ABOUT NEWS EVENTS CONTACT

# Getting Started with Pivotal Cloud Foundry

Introduction

Install the CF CLI

Deploy the sample app

View the logs

Connect a database

Scale the app

Next steps

We hope you have enjoyed learning the basic concepts of deploying an app to Pivotal Cloud Foundry.

**Topics to explore:**

How PCF Works  
<https://docs.pivotal.io/pivotalcf/concepts/>

Deploying PCF (IaaS-specific guides for installing PCF)  
<https://docs.pivotal.io/pivotalcf/deploying/>

Spring Guides  
<https://spring.io/guides>

**More sample apps:**

Ruby on Rails  
[https://github.com/cloudfoundry-samples/pong\\_matcher\\_rails](https://github.com/cloudfoundry-samples/pong_matcher_rails)

Node.js  
[https://github.com/cloudfoundry-samples/pong\\_matcher\\_sails](https://github.com/cloudfoundry-samples/pong_matcher_sails)

Python  
[https://github.com/cloudfoundry-samples/pong\\_matcher\\_django](https://github.com/cloudfoundry-samples/pong_matcher_django)

Go



Instances	1	4
1	1	4

Introduction  
Install the CF CLI  
Deploy the sample app  
View the logs  
Connect a database  
Scale the app

## Next steps

### Next steps

We hope you have enjoyed learning the basic concepts of deploying an app to Pivotal Cloud Foundry.

### Topics to explore:

How PCF Works

<https://docs.pivotal.io/pivotalcf/concepts/>

Deploying PCF (IaaS-specific guides for installing PCF)

<https://docs.pivotal.io/pivotalcf/deploying/>

Spring Guides

<https://spring.io/guides>

### More sample apps:

Ruby on Rails

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_rails](https://github.com/cloudfoundry-samples/pong_matcher_rails)

Node.js

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_sails](https://github.com/cloudfoundry-samples/pong_matcher_sails)

Python

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_django](https://github.com/cloudfoundry-samples/pong_matcher_django)

Go

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_go](https://github.com/cloudfoundry-samples/pong_matcher_go)

Introduction  
Install the CF CLI  
Deploy the sample app  
View the logs  
Connect a database  
Scale the app

## Next steps

### Next steps

Nice work! You have deployed and scaled your first app in PCF.

### Topics to explore:

How PCF Works

<https://docs.pivotal.io/pivotalcf/concepts/>

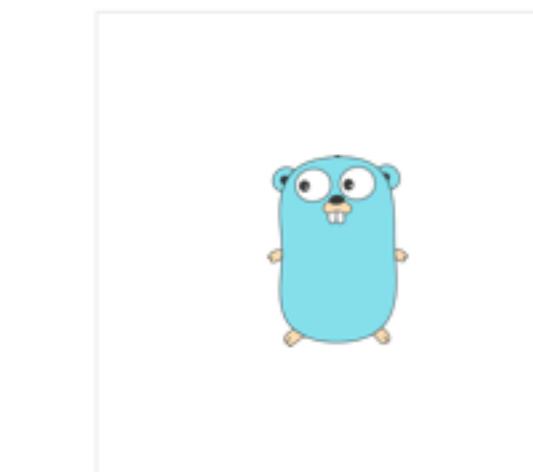
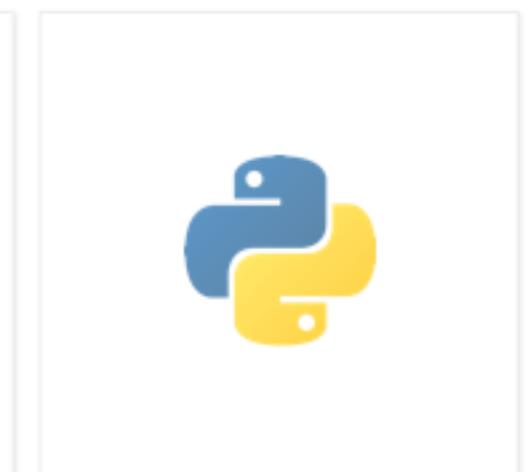
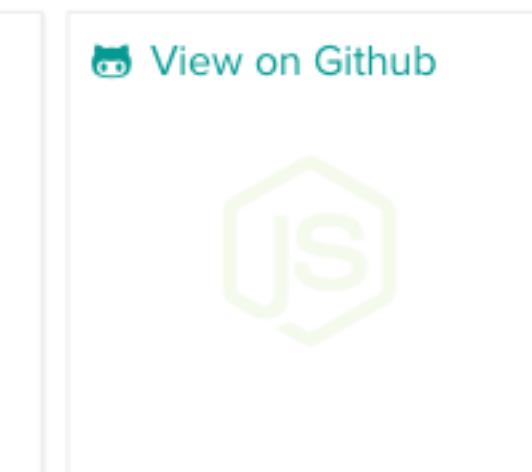
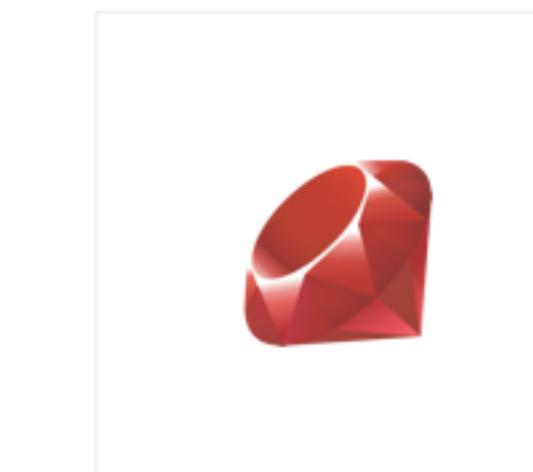
Deploying PCF (IaaS-specific guides for installing PCF)

<https://docs.pivotal.io/pivotalcf/deploying/>

Spring Guides

<https://spring.io/guides>

### Additional language-specific sample apps:



Connect with us!



# Getting Started with Pivotal Cloud Foundry

Introduction
Install the CF CLI
Deploy the sample app
View the logs
Connect a database
Scale the app
<b>Next steps</b>

## Next steps

We hope you have enjoyed learning the basic concepts of deploying an app to Pivotal Cloud Foundry.

### Topics to explore:

How PCF Works

<https://docs.pivotal.io/pivotalcf/concepts/>

Deploying PCF (IaaS-specific guides for installing PCF)

<https://docs.pivotal.io/pivotalcf/deploying/>

Spring Guides

<https://spring.io/guides>

### More sample apps:

Ruby on Rails

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_rails](https://github.com/cloudfoundry-samples/pong_matcher_rails)

Node.js

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_sails](https://github.com/cloudfoundry-samples/pong_matcher_sails)

Python

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_django](https://github.com/cloudfoundry-samples/pong_matcher_django)

Go

[https://github.com/cloudfoundry-samples/pong\\_matcher\\_go](https://github.com/cloudfoundry-samples/pong_matcher_go)

# Getting Started with Pivotal Cloud Foundry

Introduction
Install the CF CLI
Deploy the sample app
View the logs
Connect a database
Scale the app
<b>Next steps</b>

## Next steps

Nice work! You have deployed and scaled your first app in PCF.

### Topics to explore:

How PCF Works

<https://docs.pivotal.io/pivotalcf/concepts/>

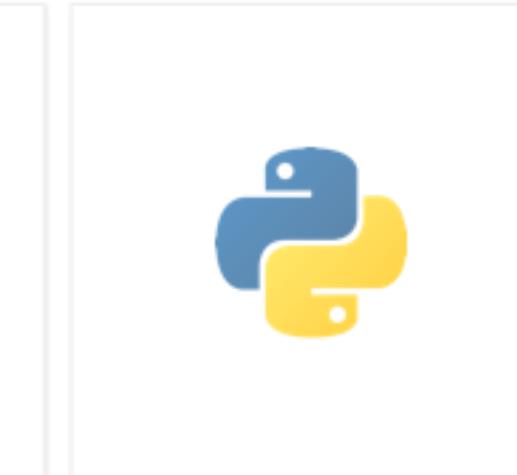
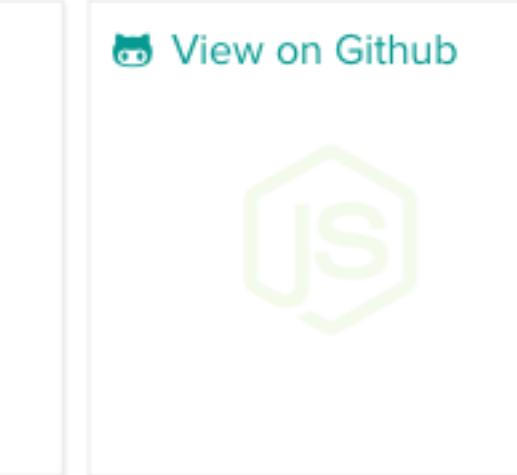
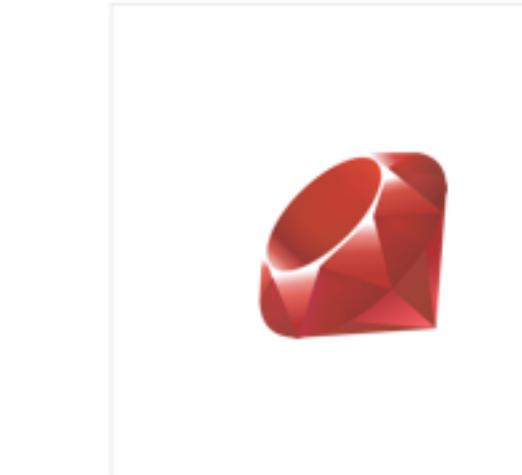
Deploying PCF (IaaS-specific guides for installing PCF)

<https://docs.pivotal.io/pivotalcf/deploying/>

Spring Guides

<https://spring.io/guides>

### Additional language-specific sample apps:



# Deploy to Production

# Criteria

- A web search for **Try Pivotal Platform** will result in finding a canonical guide.
- Success can be achieved in **15 mins or less**.

**Success = deploy** an app, make a **change**, **redeploy**

- The getting started guide will **reveal a path** to learn **deeper concepts**.

A screenshot of a Google search results page on a Mac OS X system. The search query "try pivotal platform" is entered in the search bar. The results are displayed in a clean, modern interface with a light gray background. The first result is a link to an introduction tutorial on Pivotal's website, followed by several other links related to getting started with Pivotal Platform.

try pivotal platform

All News Shopping Images Videos More Settings Tools

About 13,000,000 results (0.47 seconds)

**Introduction | Try Pivotal Platform on the Public Cloud | Pivotal**  
https://pivotal.io › pcf-tutorials › getting-started-with-pivotal-cloud-foundry ▾  
Introduction. Use this tutorial to get a simple Spring microservice up and running on **Pivotal Platform**. To save time, we will be using **Pivotal Web Services**, ...

**Introduction | Try Pivotal Platform on your Local Workstation ...**  
https://pivotal.io › platform › getting-started-with-pivotal-cloud-foundry-dev ▾  
Try **Pivotal Platform** on your Local Workstation with Pivotal Platform Dev. A tutorial for learning Pivotal Platform app development concepts with Pivotal Platform ...

**Getting Started with Pivotal Platform | Pivotal**  
https://pivotal.io › platform › pcf-tutorials › getting-started-with-pivotal-cl... ▾  
Try **Pivotal Platform** by deploying a sample app within a matter of minutes. ... running on Pivotal Web Services, an instance of Pivotal Platform hosted by Pivotal.

**Install | Try Pivotal Platform on the Public Cloud | Pivotal**  
https://pivotal.io › pcf-tutorials › getting-started-with-pivotal-cloud-foundry ▾  
Try the following command to test that the cf CLI works: cf help. You can use the cf CLI to perform all commands on apps deployed to **Pivotal Platform**. Let's Keep ...

**Next steps | Try Pivotal Platform on the Public Cloud | Pivotal**  
https://pivotal.io › pcf-tutorials › getting-started-with-pivotal-cloud-foundry ▾  
Try **Pivotal Platform** on the Public Cloud. 15-minute tutorial for learning Pivotal Platform app deployment concepts. Introduction ... Pivotal Platform Documentation

**Deploy | Try Pivotal Platform on the Public Cloud | Pivotal**  
https://pivotal.io › platform › pcf-tutorials › deploy-the-sample-app ▾

gle

try pivotal platform

All News Shopping Images

About 13,000,000 results (0.47 seconds)

### Introduction | Try Pivotal Platform

<https://pivotal.io> › pcf-tutorials › getting-started

Introduction. Use this tutorial to get a simple Spring microservice up and running on Pivotal Platform. To save time, we will be using Pivotal Web Services.

### Introduction | Try Pivotal Platform

<https://pivotal.io> › platform › getting-started

Try Pivotal Platform on your Local Workstation. Learn Pivotal Platform app development concepts without leaving your workstation.

### Getting Started with Pivotal Platform

<https://pivotal.io> › platform › pcf-tutorials

Try Pivotal Platform by deploying a sample application to Pivotal Platform. Learn Pivotal Platform app deployment concepts with Pivotal Web Services, an instance of Pivotal Platform.

### Install | Try Pivotal Platform on the Public Cloud

<https://pivotal.io> › pcf-tutorials › getting-started

Try the following command to test that the cf command is available. You can also perform all commands on apps deployed to Pivotal Platform.

### Next steps | Try Pivotal Platform

<https://pivotal.io> › pcf-tutorials › getting-started

Try Pivotal Platform on the Public Cloud. 15-minute tutorial for learning Pivotal Platform app deployment concepts. Introduction ... Pivotal Platform.

### Deploy | Try Pivotal Platform on the Public Cloud

<https://pivotal.io> › platform › pcf-tutorials

Pivotal + VMware: Transforming how more of the world builds software →



Why Pivotal Customers Products & Services Resources Company Support Sign In



Pivotal Platform

# Try Pivotal Platform on the Public Cloud

15-minute tutorial for learning Pivotal Platform app deployment concepts

## Introduction

Use this tutorial to get a simple Spring microservice up and running on Pivotal Platform. To save time, we will be using Pivotal Web Services, an instance of Pivotal Platform hosted by Pivotal.

Ensure you have:

- a free **Pivotal Web Services** account
- familiarity with command line interfaces

### Introduction

Install the CF CLI

Deploy the Sample App

View the Logs

Connect a Database

Scale the App

Next Steps

Contact us

# 45 days of data — 8000 visitors — 25% completed



## Getting Started with Pivotal Cloud Foundry

15-minute tutorial for learning Pivotal Cloud Foundry app deployment concepts

INTRODUCTION  
INSTALL THE PCF  
DEPLOY THE SAY\_HELLO\_APP  
VIEW THE LOGS  
CONNECT A DATABASE  
SCALE THE APP  
NEXT STEPS

### Introduction

Use this tutorial to get a sample Java application up and running on Pivotal Cloud Foundry (PCF). To save time, we will be using Pivotal Web Services, an instance of PCF hosted by Pivotal.

Ensure you have:

- a free Pivotal Web Services account
- Java 7 or higher installed on your workstation

[I'M READY TO CONTINUE >](#)



## Getting Started with Pivotal Cloud Foundry

15-minute tutorial for learning Pivotal Cloud Foundry app deployment concepts

INTRODUCTION  
INSTALL THE PCF  
DEPLOY THE SAY\_HELLO\_APP  
VIEW THE LOGS  
CONNECT A DATABASE  
SCALE THE APP  
NEXT STEPS

### Next Steps

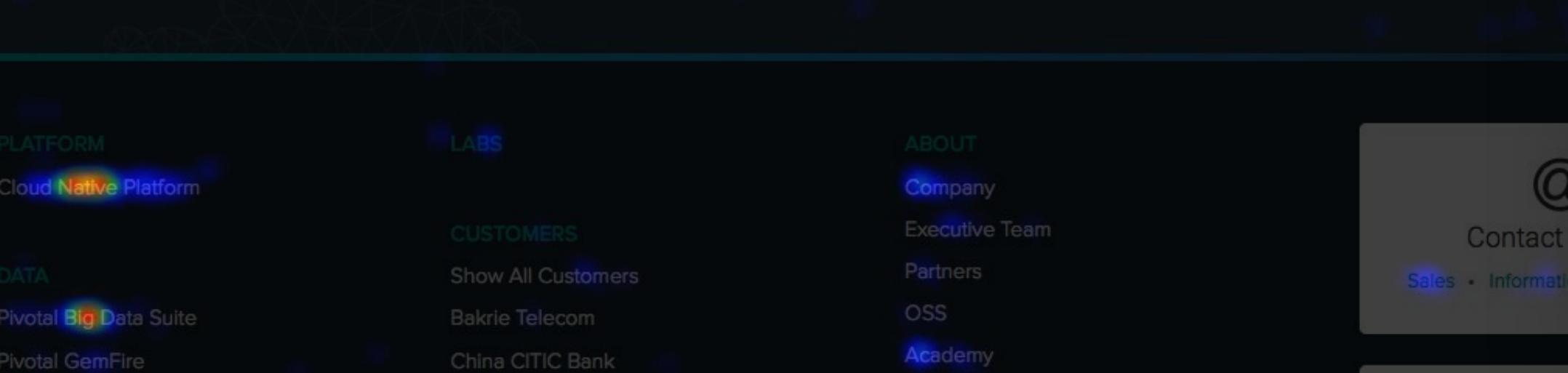
Nice work! You have just deployed and scaled an app with PCF!

**Topics to explore:**

- How PCF Works  
<https://docs.pivotal.io/pivotalcf/concepts>
- PCF Documentation  
<https://docs.pivotal.io/pivotalcf/installing/pcf-docs.html>
- Installing PCF (IaaS-specific guides for installing PCF)  
<https://docs.pivotal.io/pivotalcf/installing/>

Learn more about the Spring framework  
<https://spring.io/guides>

**Additional language-specific sample apps:**



# 45 days of data — 8000 visitors — 25% completed

The image displays two versions of a "Getting Started with Pivotal Cloud Foundry" tutorial page, overlaid with a heatmap showing visitor interaction patterns. The heatmap uses a color gradient where darker shades indicate higher visitor activity.

**Left Page (Version 1):**

- Header:** Pivotal Cloud Foundry logo.
- Title:** Getting Started with Pivotal Cloud Foundry
- Text:** 15-minute tutorial for learning Pivotal Cloud Foundry app deployment concepts
- Content:** Introduction, Deploy the Sample App, View the Logs, Connect a Database, Scale the App, and Next Steps.
- Section:** Introduction (with sub-sections: Use this tutorial to get a sample Java application up and running on Pivotal Cloud Foundry (PCF). To save time, we will be using Pivotal Web Services, an instance of PCF hosted by Pivotal. Ensure you have: a free Pivotal Web Services account, Java 7 or higher installed on your workstation.)
- Call-to-Action:** I'M READY TO CONTINUE >

**Right Page (Version 2):**

- Header:** Pivotal Cloud Foundry logo.
- Title:** Getting Started with Pivotal Cloud Foundry
- Text:** 15-minute tutorial for learning Pivotal Cloud Foundry app deployment concepts
- Content:** Introduction, Install PCF, Deploy the Sample App, View the Logs, Connect to a Database, Scale the App, and Next Steps.
- Section:** Next Steps (with sub-sections: Nice work! You have just deployed and scaled an app with PCF! Topics to explore: <https://docs.pivotal.io/pivotalcf/concepts>, PCF Documentation: <https://docs.pivotal.io/pivotalcf/installing/pcf-docs.html>, Installing PCF (IaaS-specific guides for installing PCF): <https://docs.pivotal.io/pivotalcf/installing/>, Learn more about the Spring framework: <https://spring.io/guides>, Additional language-specific sample apps: Node.js sample app icon)

**Bottom Navigation:**

- Platform: Cloud Native Platform
- Labs
- About Company: Executive Team, Partners, OSS, Academy
- Customers: Show All Customers, Bakrie Telecom, China CITIC Bank
- Data: Pivotal Big Data Suite, Pivotal GemFire
- Contact Pivot: Sales, Information, C

```
$ cd spring-music
```

Sign in to PWS:

```
$ cf login -a api.run.pivotal.io
...
API endpoint: https://api.run.pivotal.io
User: email@example.com
Org: your-org
Space: development
```

Use Gradle to assemble the app locally:

```
$ ./gradlew assemble
```

Push the app to PWS:

```
$ cf push
```

View the app running on PWS:

```
requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: spring-music-RANDOM-WORD.cfapps.io
last uploaded: Thu Feb 25 15:45:49 UTC 2016
stack: unknown
...
state      since
details          cpu    memory    disk
```

# Criteria

- A web search for **Try Pivotal Platform** will result in finding a canonical guide.
- Success can be achieved in **15 mins or less**.

**Success = deploy** an app, make a **change**, **redeploy**

- The getting started guide will **reveal a path** to learn **deeper concepts**.

Not Secure — node-js-sample-app-chatty-quokka.dev.cfdev.sh

The screenshot shows the Cloud Foundry interface for a sample application named "node-js-sample-app". The top navigation bar includes standard browser controls (back, forward, search) and a refresh icon. The title bar displays the URL "Not Secure — node-js-sample-app-chatty-quokka.dev.cfdev.sh". The main content area features a logo of a blue rabbit with gears and a green JS icon. Below this, a table provides detailed information about the application:

	BUILDPACK	Buildpacks		
	Node.js	<a href="#">Buildpacks ↗</a>		
	APP NAME	APP URIS	Routes & Domains	
	node-js-sample-app	node-js-sample-app-chatty-quokka.dev.cfdev.sh	<a href="#">Routes &amp; Domains ↗</a>	
	INSTANCE INDEX	MEMORY LIMIT	DISK LIMIT	Scaling
	0	512M	1024M	<a href="#">Scaling ↗</a>
	SPACE NAME	Orgs & Spaces		
	cfdev-space	<a href="#">Orgs &amp; Spaces ↗</a>		
	Manage Services			
	There aren't any services bound to this app.			<a href="#">Manage Services ↗</a>

At the bottom, a small icon of a blue gear with a circle is followed by the text "This is a Cloud Foundry sample application."

Not Secure — node-js-sample-app-chatty-quokka.dev.cfdev.sh

The screenshot shows the Cloud Foundry application dashboard for the app 'node-js-sample-app'. The dashboard includes a summary table with details like buildpack, app name, instance index, space name, and memory limits. To the right, there's a sidebar with links for managing buildpacks, routes, scaling, orgs, and services. A yellow box highlights the sidebar area.

	BUILDPACK		
	Node.js		
	APP NAME	APP URIS	
	node-js-sample-app	node-js-sample-app-chatty-quokka.dev.cfdev.sh	
	INSTANCE INDEX	MEMORY LIMIT	DISK LIMIT
	0	512M	1024M
	SPACE NAME		
	cfdev-space		
	There aren't any services bound to this app.		

This is a Cloud Foundry sample application.

[Buildpacks](#)

[Routes & Domains](#)

[Scaling](#)

[Orgs & Spaces](#)

[Manage Services](#)

Not Secure — node-js-sample-app-chatty-quokka.dev.cfdev.sh

The screenshot shows the Cloud Foundry UI for the application "node-js-sample-app". The top navigation bar includes standard browser controls (back, forward, search) and a refresh icon. The title bar displays the URL "Not Secure — node-js-sample-app-chatty-quokka.dev.cfdev.sh". The main content area features a logo with a blue rabbit and a green JS icon. Below the logo, the application's configuration is listed:

	BUILDPACK	Buildpacks		
	Node.js	<a href="#">Buildpacks ↗</a>		
	APP NAME	APP URIS	Routes & Domains	
	node-js-sample-app	node-js-sample-app-chatty-quokka.dev.cfdev.sh	<a href="#">Routes &amp; Domains ↗</a>	
	INSTANCE INDEX	MEMORY LIMIT	DISK LIMIT	Scaling
	0	512M	1024M	<a href="#">Scaling ↗</a>
	SPACE NAME	Orgs & Spaces		
	cfdev-space	<a href="#">Orgs &amp; Spaces ↗</a>		
	There aren't any services bound to this app.			<a href="#">Manage Services ↗</a>

A note at the bottom states: "This is a Cloud Foundry sample application."

Cloud Foundry Documentation

Services Overview

Managing Service Instances

Sharing Service Instances

Delivering Service Credentials to an App

Managing Service Keys

Buildpacks

Configuring Play Framework Service Connections

Using an External File System (Volume Services)

User-Provided Service Instances

> Streaming App Logs

> Managing Apps with the cf CLI

Cloud Foundry Environment Variables

Cloud Controller API Client Libraries

Considerations for Designing and Running an App in the Cloud

App Revisions

> Buildpacks

## Creating Service Instances

You can create a service instance with the following command:

```
cf create-service SERVICE PLAN SERVICE_INSTANCE
```

Use the information in the list below to replace `SERVICE`, `PLAN`, and `SERVICE_INSTANCE` with appropriate values.

- `SERVICE` : The name of the service you want to create an instance of.
- `PLAN` : The name of a plan that meets your needs. Service providers use [plans](#) to offer varying levels of resources or features for the same service.
- `SERVICE_INSTANCE` : The name you provide for your service instance. You use this name to refer to your service instance with other commands. Service instance names can include alphanumeric characters, hyphens, and underscores, and you can rename the service instance at any time.

```
$ cf create-service rabbitmq small-plan my-rabbitmq
Creating service my-rabbitmq in org console / space development as user@example.com
OK
```

User Provided Service Instances provide a way for developers to bind apps with services that are not available in their Cloud Foundry marketplace. For more information, see the [User Provided Service Instances](#) topic.

**Note:** When multiple brokers provide two or more services with the same name, you must specify the broker by including the `-b BROKER` flag in the `cf create-service` command.

```
[mike@Lahar cf-sample-app-nodejs % cf create-service p-mysql 20mb mysql
Creating service instance mysql in org cfdev-org / space cfdev-space as user...
OK
y exists
-app-nodejs % cf bind-service node-js-sample-app mysql
1 to app node-js-sample-app in org cfdev-org / space cfdev-s
node-js-sample-app' to ensure your env variable changes tak
-app-nodejs % cf restart node-js-sample-app
js-sample-app in org cfdev-org / space cfdev-space as user..
tart...
ode-js-sample-app
tarted
ode-js-sample-app-chatty-quokka.dev.cfdev.sh
ri 25 Oct 10:45:00 PDT 2019
flinuxfs3
odejs
M
start
cpu      memory      disk      details
10-25T19:11:41Z  0.3%  53.1M of 512M  88M of 1G
-app-nodejs % cf services
org cfdev-org / space cfdev-space as user...
apps      last operation      broker      upgra
mb      node-js-sample-app      create succeeded      p-mysql
-app-nodejs %
```

Cloud Foundry Documentation

Services Overview

Managing Service Instances

Sharing Service Instances

Delivering Service Credentials to an App

Managing Service Keys

Configuring Play Framework Service Connections

Using an External File System (Volume Services)

User-Provided Service Instances

> Streaming App Logs

> Managing Apps with the cf CLI

Cloud Foundry Environment Variables

Cloud Controller API Client Libraries

Considerations for Designing and Running an App in the Cloud

App Revisions

> Buildpacks

Creating Service Instances

You can create a service instance with the following command:

```
cf create-service SERVICE PLAN SERVICE_INSTANCE
```

Use the information in the list below to replace `SERVICE`, `PLAN`, and `SERVICE_INSTANCE` with appropriate values.

- `SERVICE`: The name of the service you want to create an instance of.
- `PLAN`: The name of a plan that meets your needs. Service providers use plans to offer different levels of resources or features for the same service.
- `SERVICE_INSTANCE`: The name you provide for your service instance. You use this name to refer to your service instance with other commands. Service instance names can include lowercase numeric characters, hyphens, and underscores, and you can rename the service instance at any time.

```
$ cf create-service rabbitmq small-plan my-rabbitmq
Creating service my-rabbitmq in org console / space development as user@myorg
OK
```

User Provided Service Instances provide a way for developers to bind apps with services that are not available in their Cloud Foundry marketplace. For more information, see the [User Provided Service Instances](#) topic.

**Note:** When multiple brokers provide two or more services with the same name, specify the broker by including the `-b BROKER` flag in the `cf create-service` command.

```
mike@Lahar cf-sample-app-nodejs % cf create-service p-mysql 20mb mysql
Creating service instance mysql in org cfdev-org / space cfdev-space as user...
OK

Service mysql already exists
mike@Lahar cf-sample-app-nodejs % cf bind-service node-js-sample-app mysql
Binding service mysql to app node-js-sample-app in org cfdev-org / space cfdev-space as user...
OK

TIP: Use 'cf restage node-js-sample-app' to ensure your env variable changes take effect
mike@Lahar cf-sample-app-nodejs % cf restart node-js-sample-app
Restarting app node-js-sample-app in org cfdev-org / space cfdev-space as user...
.

Stopping app...
Waiting for app to start...

name: node-js-sample-app
requested state: started
routes: node-js-sample-app-chatty-quokka.dev.cfdev.sh
last uploaded: Fri 25 Oct 10:45:00 PDT 2019
stack: cflinuxfs3
buildpacks: nodejs

type: web
instances: 1/1
memory usage: 512M
start command: npm start
state since cpu memory disk details
#0 running 2019-10-25T19:11:41Z 0.3% 53.1M of 512M 88M of 1G
mike@Lahar cf-sample-app-nodejs % cf services
Getting services in org cfdev-org / space cfdev-space as user...

name service plan apps last operation broker upgrade
mysql p-mysql 20mb node-js-sample-app create succeeded p-mysql
mike@Lahar cf-sample-app-nodejs %
```

Not Secure — node-js-sample-app-chatty-quokka.dev.cfdev.sh

The screenshot shows the Cloud Foundry application dashboard for the 'node-js-sample-app' application. The dashboard includes sections for Buildpack (Node.js), Routes & Domains (APP NAME: node-js-sample-app, APP URIS: node-js-sample-app-chatty-quatty-dev.cfdev.sh), Scaling (INSTANCE INDEX: 0, MEMORY LIMIT: 512M, DISK LIMIT: 1024M), Orgs & Spaces (SPACE NAME: cfdev-space), and Services. The Services section displays a service instance 'p-mysql' with details: SERVICE: p-mysql, SERVICE NAME: mysql, and SERVICE PLAN: 20mb. A 'Configuring Services' button is also present. The entire Services section is highlighted with a yellow box.

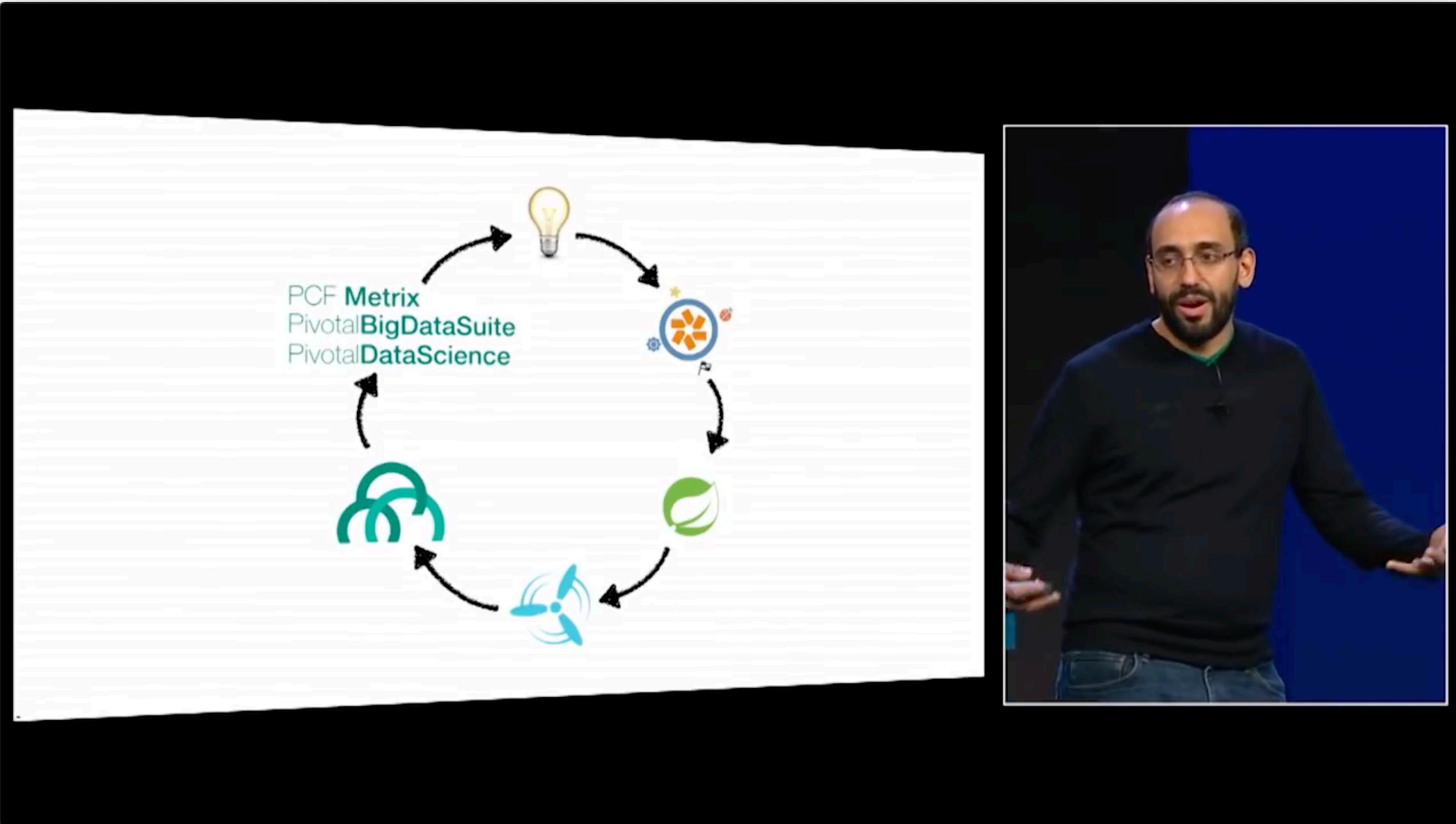
```
[mike@Lahar cf-sample-app-nodejs % cf create-service p-mysql 20mb mysql
Creating service instance mysql in org cfdev-org / space cfdev-space as user...
OK
y exists
-app-nodejs % cf bind-service node-js-sample-app mysql
1 to app node-js-sample-app in org cfdev-org / space cfdev-s
node-js-sample-app' to ensure your env variable changes tak
-app-nodejs % cf restart node-js-sample-app
ode-js-sample-app
tart...
ode-js-sample-app-chatty-quatty-dev.cfdev.sh
ri 25 Oct 10:45:00 PDT 2019
flinuxfs3
odejs
M
start
cpu      memory      disk      details
10-25T19:11:41Z  0.3%  53.1M of 512M  88M of 1G
-app-nodejs % cf services
org cfdev-org / space cfdev-space as user...
mb      node-js-sample-app      create succeeded      p-mysql
-app-nodejs %
```

# User Onboarding Experience

## Outcomes:

- Top acquisition channel for our public cloud
- Recognized as a well conceived "getting started" guide
- Emulated by other teams

# Project Bootstrapping







## Customer Journey Map - Application Development

Clinton - Enterprise Application Developer - The Home Depot



\*This activity or process occurs when deemed necessary

ect to do some level of discovery  
cts any D&F activities preceeding  
l be helpful\*

- Expect to be straightforward, easy
- Expect less time than if provisioned infrastructure and platform themselves

- As easy as Jenkins
- Good documentation
- Language I understand
- Not involve duplicative work

- Quick review of stories
- Expect small story size
- Expect sizing that gets easier relatively soon into project

- Expect set up to be complete
- Expect to be able to focus on TDD and story at hand
- Expect to pair

PCF isn't always the right tool to use for app development i.e. If you need to do a lot of background processing, it could be done asynchronously... maybe it's large amounts of data, then it isn't the right tool. Instead, some messaging system. Would use RabbitMQ, Kafka, Google pub. AWS also has services like SQS."

"I Love PCF. Does a lot of the heavy lifting from a developer perspective"  
"Takes away all the boilerplate duties a developer would have to set up an environment."

[Setting up PCF roles] "We always assumed that if it could be capabilities-based, and [we] define our own roles OR even if it came out of the box with the Pivotal opinion, 'this is a good structure' - that would be great but then we could go in and tweak it."

"They have a very complicated build process...we were trying to use Concourse but the job config got very hairy so we abandoned it for Jenkins 2. Our job spec in Concourse was about 1,500 lines of yaml where Jenkins is like 21."

"...there really wasn't much documentation..."

[Would like a UI to auto build yaml file]  
"I want to accomplish this task: here are my secrets/keys/repos... and then presto, CI/CD set up."

[What's working] "IPM meetings. I really like the Gherkin format because initially it was really difficult for me to understand every time I would sit in the IPM meeting, but now once we start coding into it, we know what's the use case we are writing.  
Development becomes really easy actually."

"As we've gone along, we've made the stories more and more granular."

"Another benefit I've seen out of pairing is the bringing to the table a different set of ideas, whereas before it was a ... "Well, I know how to do it this way," and it's like, "Oh, well there's actually this more efficient way to do it. Awesome. Cool. Let's do it, let's look at this better way to do it."



Developers expect a certain amount of technical discovery prior to a project to do deeper thinking about how to build their MVP: which language to use, whether or not PCF will be a good fit,

App developers at The Home Depot expressed wanting the ability to add multiple users at once when setting up a project in Pivotal Tracker. They can't currently do this using LDAP IDs.

The Home Depot built a Pipeline Generator Tool that takes a list of app names from which individual pipelines are created. Apps follow a common template. Changes made to template get cascaded down to all apps. This makes maintenance manageable.

Some good news: Automated pipelines are coming but ONLY for approved teams/apps. To be approved, apps need to be "out of scope" (meaning, not impacting financials).

Overall, the devs we spoke to were far along on their transformation journey so they mentioned the D&F as part of their normal process even when building internal apps that fill gaps in Pivotal products.

The key challenge we heard was the art of identifying the MVP.

One team we spoke with aspires to write stories that can take about a half a day.

These devs had a positive view on pairing. They mentioned knowledge sharing, efficiency and communication improvements within their working culture.

They are measuring their success by delivering features, not points.

set up to be complete  
to be able to focus on TDD and  
hand  
to pair

- Expect this to be technically simple
- May involve add'l process (if other than PM needs to accept)
- May help demo story, if no UI

Another benefit I've seen out of pairing is the bringing to the table a different set of ideas, whereas before I'd say ... "Well, I know how to do it this way," and it's like, "Oh, well there's already this more efficient way to do it somehow. Cool. Let's do it, let's do it this better way to do it."

"It'd be useful to have finer-grained permissions in Tracker. For example, for my project, we'd like the PO to only accept/reject stories."



Devs had a positive view on pairing. They mentioned knowledge sharing, collaboration and communication are key components within their working culture. They measured their success by shipping features, not points.



Releasing to non-prod for acceptance is relatively easy, and accepting/rejecting is also often an easy process. We did hear a desire to put safeguards on who could accept/reject a story (and also doing other things in Tracker) but overall, this is neither happy or sad-- just neutral.

- Be able to take next story in queue
- Have this next story be an efficient use of time

"I'd like an easier way to link stories and manage dependencies."

"Dependencies can be a challenge -- like, when one pair picks up a story, then another pair picks up a story that really needs to happen before the first... this happens a lot."

"Being stuck with four story types in Tracker can be a little limiting."



More than one team brought up how difficult it is to manage dependencies in Tracker, which suggest this could present challenges more often than not. They talked also about how stories that get blocked can become dependencies on multiple other stories and that creates a logjam at the top of their backlog.



- Expect wide number of add-ons
- Expect add-ons be available in non-prod
- Expect selected add-ons to be made available easily in prod

"I'd like to be able to access more add-ons to do quick POCs in non-prod."

"I wanted to see how easy it'd be to build Search into our app, and ended up doing sql-like queries that are pre-configured. I'd rather have used something like Solr, but it wasn't offered as an add-on and it was taking too long to explore integration."

"Heroku has tons of add-ons!"



The Home Depot seems to have a limited number of add-on options which can compromise their ability to come up with the fastest/most efficient solutions. Because this only sometimes becomes an issue, we noted the sentiment as neutral.

- Expect paperwork
- Expect to have to go back for security
- Expectation this will last 2-3 weeks
- Expect to be given timebound opportunity

"IRB is a speed to market blocker for us.  
"In scope apps (those needing SOX compliance) require a lot of work."

"Pushing to production involves [Tony] checking all the boxes, someone on the team has to do it." This is a challenge for the ability to push to prod for a finite period of time.

"We're struggling w/ blue-green deployment."

"Doing a canary deployment is becoming a challenge because all our APIs are consumed by multiple teams. They could first test certain load and then roll it out."

All permissions for app devs to push to prod are controlled by the DevOps team following the process to ensure SOX compliance. This is a challenge for new process that will enable teams to get their code to prod faster if they meet very specific criteria (e.g. no sensitive data). Compliance is a priority for The Home Depot.

vide number of add-ons  
dd-ons be available in non-prod  
elected add-ons to be made  
easily in prod

- Expect paperwork
- Expect to have to go back for security review as many times as needed
- Expectation this will last 2-3 weeks
- Expect to be given timebound opportunity to CF Push after passing IRB

- Expect to easily monitor app
- Expect performance alerts
- Expect ability to easily share health of releases with stakeholders

- Expect ease in locating issues in logs
- Expect issue alerts
- Expect ability to push quick hotfixes

I'd like to be able to access more logs to do quick POCs in prod."

"I'd like to see how easy it'd be to search into our app, and ended up writing sql-like queries that are reconfigured. I'd rather have used something like Solr, but it wasn't available as an add-on and it was too long to explore this option."

"You has tons of add-ons!"



The Home Depot seems to have a limited set of add-on options which can compromise their ability to come up with best/most efficient solutions. This only sometimes becomes an issue, so I'm noting the sentiment as neutral.

"IRB is a speed to market blocker for us"

"In scope apps (those needing SOX compliance) can't get pre-approval for CI/CD to production"

"Pushing to production involves [Tony's] team getting a change request and, if the app checks all the boxes, someone on the team who didn't write the code will get the ability to push to prod for a finite period of time"

"We're struggling w/ blue-green deployments."

"Doing a canary deployment is becoming a challenge. That would be good because all our APIs are consumed by external applications so it would be good if they could first test certain load and then open it up to the entire audience."



"I'd like improved metric information in a GUI that is light enough that The Home Depot won't turn it off. They turned off PCF metrics for us because it was too heavy."

"I'd like to be able to measure the frequency of crashes."



"I'd like the ability to query and search logs, and be alerted about issues as soon as they occur."

"Having to sort through a bunch of logs that aren't sorted in any particular order is painful when I'm trying to understand what happened during a crash."



"It'd be nice to know who pushes the application to PCF last, with some additional information about that push."



All permissions for app devs to push to prod were revoked when some teams weren't following the process to ensure SOX compliance. The Home Depot is about to launch a new process that will enable teams to get pre-approval for pipelines for CF push to prod if they meet very specific criteria (e.g. not needing SOX compliance). Ensuring SOX compliance is a priority for The Home Depot, and applies to most applications.

App developers have some basic expectations about what information they'd get about their app in order to ensure it's functioning properly. It seems that THD turned off providing this information for app devs because of the weight of the analytics application.

Troubleshooting PCF apps involves combing through disorganized logs. Add to this that developers aren't able to push fixes directly to prod, and the process of troubleshooting becomes a painful one.

The screenshot shows a Google Docs spreadsheet with a single column of text. The text contains a series of Enso commands, each preceded by a bolded section header. The sections and their corresponding commands are:

- Set template repo target for app templates**  
\$ enso target-app <https://github.com/enterprise/templates/spring/app>
- Create local app, pulling template from template repo**  
\$ enso create-app spring microservice myapp
- Set my git repo target**  
\$ enso target-git <http://github.com/enterprise/ag/>
- Login to git repo**  
\$ enso login-git
- Create a new git project**  
\$ enso create-git project0
- Project tracking & git integration**  
\$ enso target-project tracker <https://www.pivotaltracker.com>
- Login to pivotal tracker**  
\$ enso login-project tracker
- Create tracker project**  
\$ enso create-project project0\_tracker
- Bind project tracker to git project**  
\$ enso bind-tracker project0\_tracker project0 master
- Set template repo target for pipeline templates**  
\$ enso target-ci <https://github.com/enterprise/templates/spring/ci>
- Download pipeline template for bg deploy to local**  
\$ enso create-ci bluegreen\_deploy pipeline.yml
- Load the pipeline into PCF concourse instance**

docs.google.com

Search the menus (Option+/)

100% Normal text Courier New 11 ...

1 2 3 4

Set template repo target for app templates  
\$ enso target-app <https://github.com/enterprise>

Create local app, pulling template from template repo  
\$ enso create-app spring microservice myapp

Set my git repo target  
\$ enso target-git <http://github.com/enterprise>

Login to git repo  
\$ enso login-git

Create a new git project  
\$ enso create-git project0

Project tracking & git integration  
\$ enso target-project tracker <https://www.pivotaltracker.com>

Login to pivotal tracker  
\$ enso login-project tracker

Create tracker project  
\$ enso create-project project0\_tracker

Bind project tracker to git project  
\$ enso bind-tracker project0\_tracker project0

Set template repo target for pipeline templates  
\$ enso target-ci <https://github.com/enterprise>

Download pipeline template for bg deploy to local instance  
\$ enso create-ci bluegreen\_deploy pipeline.yml

Load the pipeline into PCF concourse instance  
\$ enso target-pipeline ci

docs.google.com

Search the menus (Option+/)

100% Normal text Courier New 11 B I U A ...

1 2 3 4 5 6 7

'<https://github.com/mycompany/my-app>'?

Git repo URL> <https://github.com/mycompany/my-app>

Now attempting to access the git repo... It looks like this is a private repo. Please login.

User> myusername  
Password> mypassword

Creating repo... Success! You now have a Git repo for your project.

Create a Pivotal Tracker project? Y/n> Y

Please specify Pivotal Tracker URL.

Tracker URL press Enter for default>  
Tracker URL: <https://www.pivotaltracker.com>

Now attempting to access Pivotal Tracker. Please login.

User> myusername  
Password> mypassword

Creating project... Success!

Connect Git repo to Tracker? Y/n> Y

Creating webhook service between Git and Tracker... Success!

Do you have a CI/CD template that you'd like to use? If you choose No, we'll provide a list of templates.

Y/n> Y

Can you provide me with your Git repo URL for your CI templates (e.g. '<https://github.com/mycompany/ci-templates/bluegreen.yml>')?

Git repo URL> <https://github.com/mycompany/ci-templates/bluegreen.yml>

MICHAEL LONG Aug 16, 2016 Resolve ...

ex. have access to private keys.  
From imported document

Michael Long Aug 16, 2016 Resolve ...

likes git integration  
From imported document

Unknown Aug 16, 2016 Resolve ...

What happens in the N case? Would it be possible to integrate directly with an existing Tracker project?  
From imported document

Reply...  
Viewers of this file can see comments and suggestions.

Michael Long Aug 16, 2016 Resolve ...

Designing for greenfield apps. Could have difficulty with larger legacy apps.  
From imported document

Unknown Aug 16, 2016 Resolve ...

In the Tracker section, would like to have a section where I can add the

nd Tracker... Success!

d like to use? If you choose

URL for your CI templates (e.g.  
[ates/bluegreen.yml](#))?

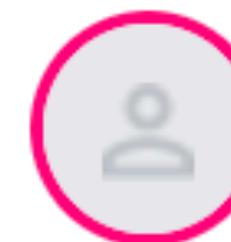
[pany/ci-templates/bluegreen.yml](#)

↳ `envc create-ci bluegreen_deploy pipeline.yml`

Load the pipeline into PCF concourse instance

Reply...

Viewers of this file can see comments and suggestions.



**Michael Long**  
Aug 16, 2016

[Resolve](#)

⋮

Designing for greenfield apps. Could have difficulty with larger legacy apps.

*From imported document*



**Unknown**  
Aug 16, 2016

[Resolve](#)

⋮

In the Tracker section, would like to have a section where I can add the



# Enterprise Identity Service

# Security Admin Experience – Identity Service

The image displays three separate browser windows, each showing a different view of the Pivotal CF Identity Services interface. All three windows are titled "localhost" and have a header bar with tabs for "IDaaS Proto - v1", "IDaaS Proto - v8", and "IDaaS Proto - v19". The top window shows the "User Stores" tab selected, displaying a list of applications and their details. The middle window shows the "Settings" tab selected. The bottom window shows the "User Stores" tab selected.

**Pivotal CF Identity Services**

SPACE East Coast ORG Acme Anvil Administrator

Acme Authentication

User Stores Settings

**Makin' The Bacon**

APP TYPE Native Mobile App

USER STORES Internal User Store and 1 more

updated 22 days ago

**Eventbrite for Golfers**

APP TYPE API Resource

USER STORE Internal User Store

updated 22 days ago

**Dropbox for Toddlers**

APP TYPE Single-Page JavaScript App

USER STORE LDAP

updated 22 days ago

**Fitbit for Pets**

APP TYPE Service-to-Service App

USER STORE Internal User Store

updated 22 days ago

**The App That Jack Built**

APP TYPE Web Browser App

USER STORE Internal User Store

updated 22 days ago

**Eventbrite for Golfers**

APP TYPE API Resource

USER STORE Internal User Store

updated 22 days ago

**Dropbox for Toddlers**

APP TYPE Single-Page JavaScript App

USER STORE LDAP

updated 22 days ago

# App Developer Experience – Identity Service

The screenshot displays the Pivotal CF Identity Services interface, specifically the Single Sign-On configuration page for an application named "The App That Jack Built".

**Header:** Shows three tabs: "IDaaS Proto – v1", "localhost", and "IDaaS Proto – v19". Below the tabs, it shows "SPACE East Coast" and "ORG Acme Anvil". A user "Administrator" is logged in.

**Breadcrumbs:** "Acme Authentication > The App That Jack Built"

**Application Details:** "The App That Jack Built" (Web Browser App) with an "Edit Config" button.

**Configuration Section:** "Set the following variables in your application's environment".

- App ID:** Unique Identifier for the application. Value: 4031479cc3479b1719e7f03a89d15923619211446edfaea2... (Copied!)
- App Secret:** Authenticates the application. Value: ..... (Regenerate App Secret)
- ID Service URL:** Need a description here. Value: https://the-app-that-jack-built.system-domain.com
- Authorize URI:** Need a description here. Value: https://the-app-that-jack-built.system-domain.com/auth
- Token URI:** Need a description here. Value: https://the-app-that-jack-built.system-domain.com/oauth

**Download Options:** "Download a Sample App" (button) and "Easily test these settings with a ready-to-go Java Spring sample app." (text).

**Left Sidebar:** Shows other applications: "Dropbox for Toddlers" (Single-Page App), "Acme Authentication > Dropbox for Toddlers", "App ID" (b035ca5...), "App Secret" (.....), "ID Service URL" (https://...), "Authorize URI" (https://...), and "Token URI" (https://...). It also includes a "Download a Sample App" button and a note to include Java code.

**Right Sidebar:** Shows "Pivotal Account" and "Visit app" links for the application.

# End User Experience – Identity Service

The image displays three side-by-side screenshots of a web-based identity service interface, likely Pivotal's, showing different views of user profiles and applications.

**Screenshot 1 (Left): Applications View**

- Header: "Pivotal Account" (repeated twice), "Home - Pivotal Web Services", "Pivotal Account".
- User Profile: "Michaeonius Long" (highlighted in blue), "mblongii@gmail.com", "Sign out".
- Profile Picture: "ML".
- Section: "The following applications have been granted permission to access your account".
- Application Card: "Pivotal Tracker" with options: "View your email address" (unchecked) and "View your basic profile information" (unchecked).
- Buttons: "Update Permissions" (blue button).

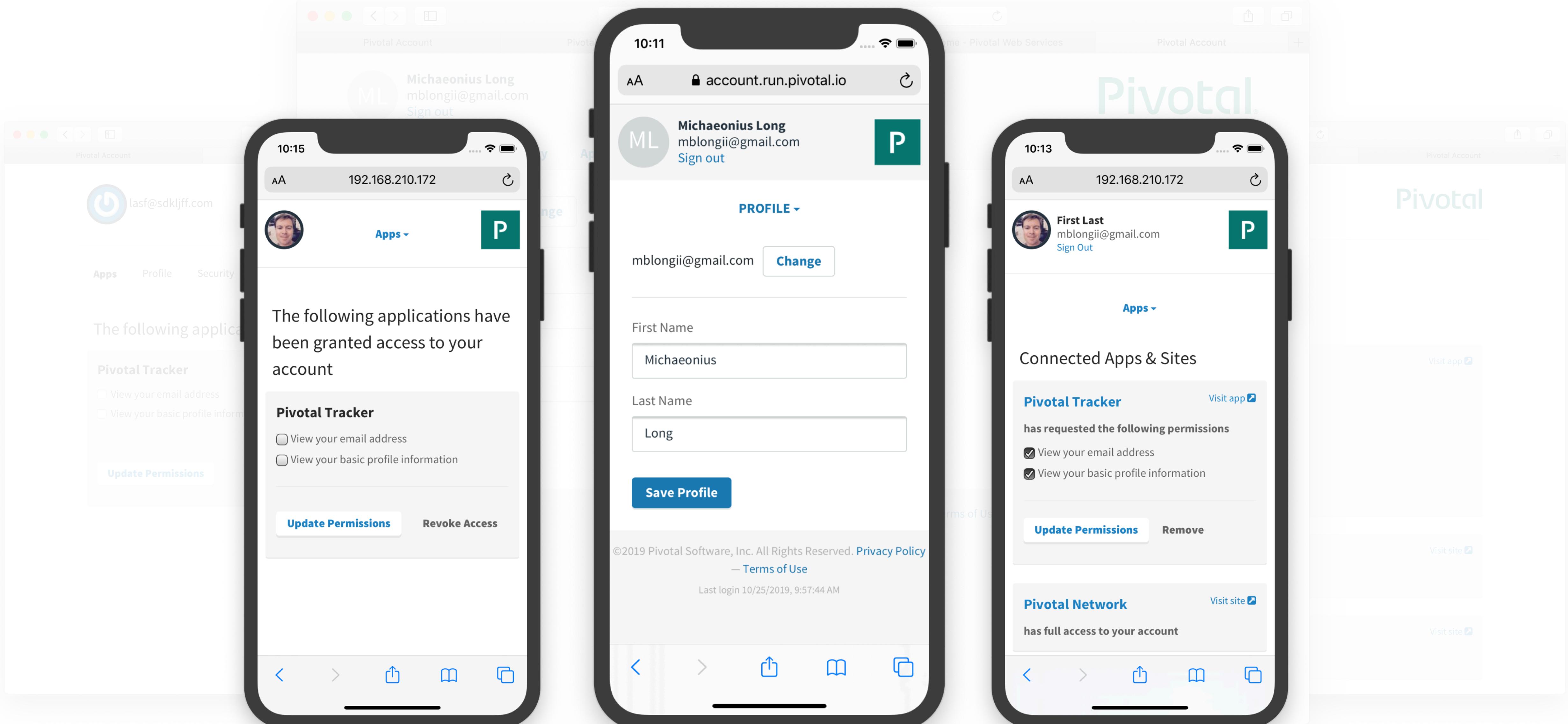
**Screenshot 2 (Middle): Profile View**

- Header: "account.run.pivotal.io" (in a browser window tab bar), "Pivotal Account" (repeated twice), "Pivotal Account - v4", "Home - Pivotal Web Services", "Pivotal Account".
- User Profile: "Michaeonius Long" (highlighted in blue), "mblongii@gmail.com", "Sign out".
- Profile Picture: "ML".
- Section: "Profile" (selected tab), "Security", "Approvals", "Notifications".
- Form Fields:
  - Email: "mblongii@gmail.com" (with "Change" button).
  - First Name: "Michaeonius".
  - Last Name: "Long".
- Buttons: "Save Profile" (blue button).
- Footer: "©2019 Pivotal Software, Inc. All Rights Reserved. [Privacy Policy](#) — [Terms of Use](#)", "Last login 10/16/2019, 12:45:17 PM".

**Screenshot 3 (Right): Home View**

- Header: "Pivotal Account" (repeated twice), "Pivotal Account".
- Pivotal Logo: "Pivotal®".
- Section: "Visit app" (with a blue link).
- Section: "Visit site" (with a blue link).
- Section: "Visit site" (with a blue link).

# End User Experience – Identity Service



## *Contents*

Background

Management Experience

Project Experience

- Enterprise Identity Service
- User Onboarding Experience
- Application Development Bootstrapping

**Michael Long**  
Product Designer + Manager

