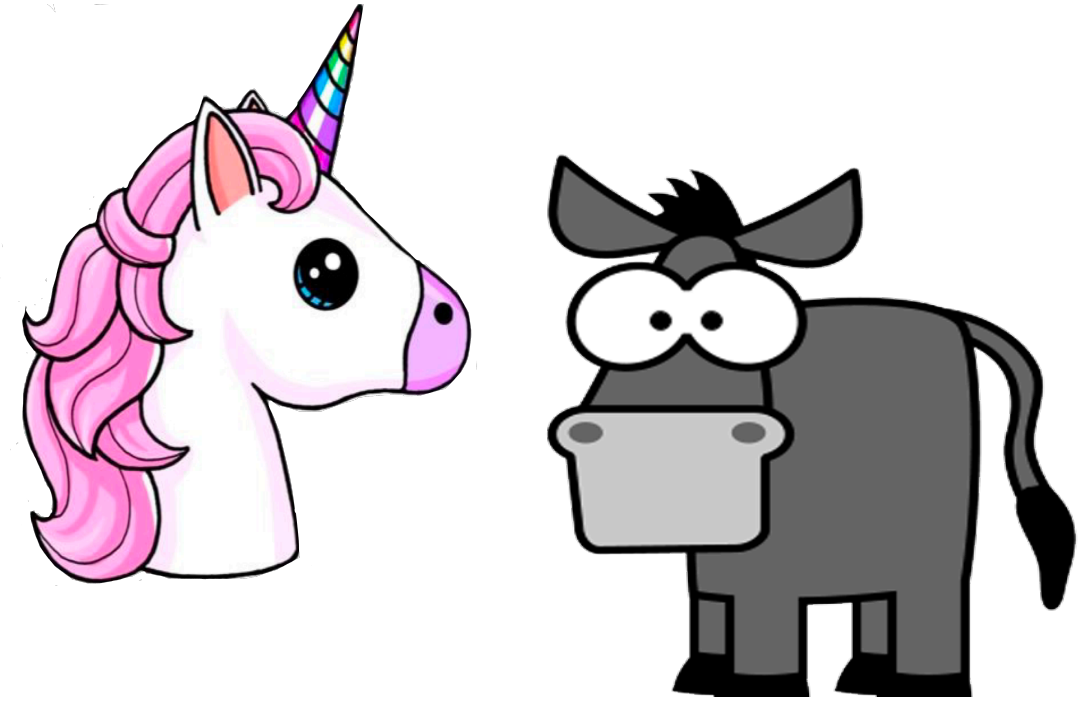
Pragmatic DevOps



Aimee D. Bechtle

Director, Product & Program Management, Cloud & Network Services
Capital One

Bio by the numbers

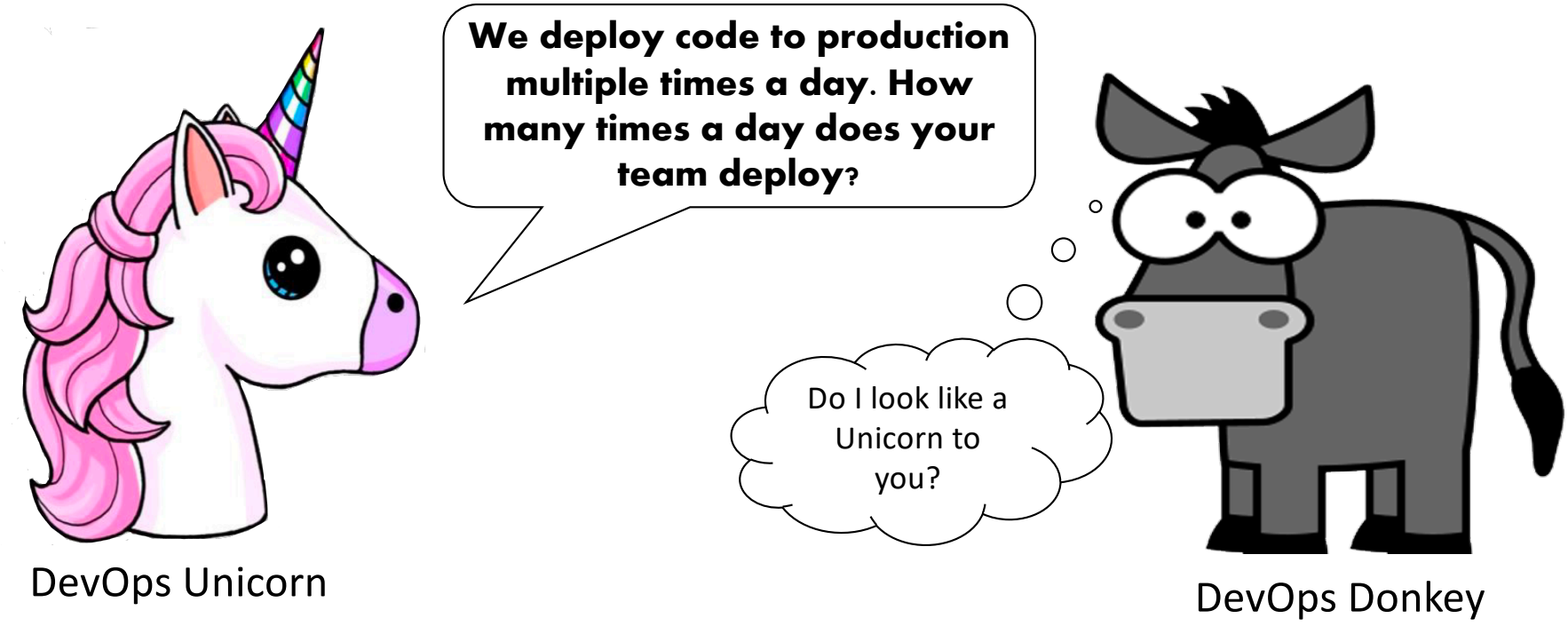
~7
Years in DevOps

2
Very
Different Companies

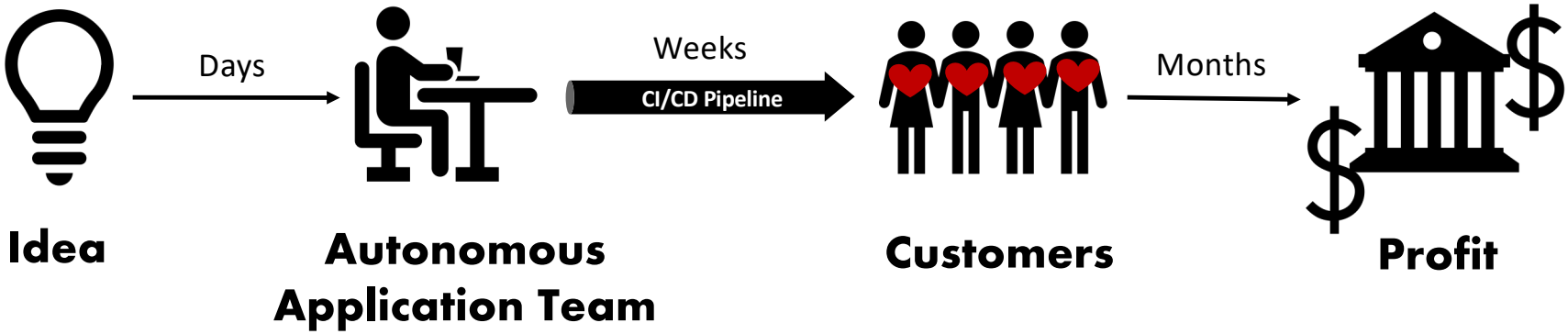
8
Different Tech
Organizations

20+
Teams

A Unicorn and a Donkey meet at a DevOps conference...



I bought into the DevOps Unicorn fantasy…



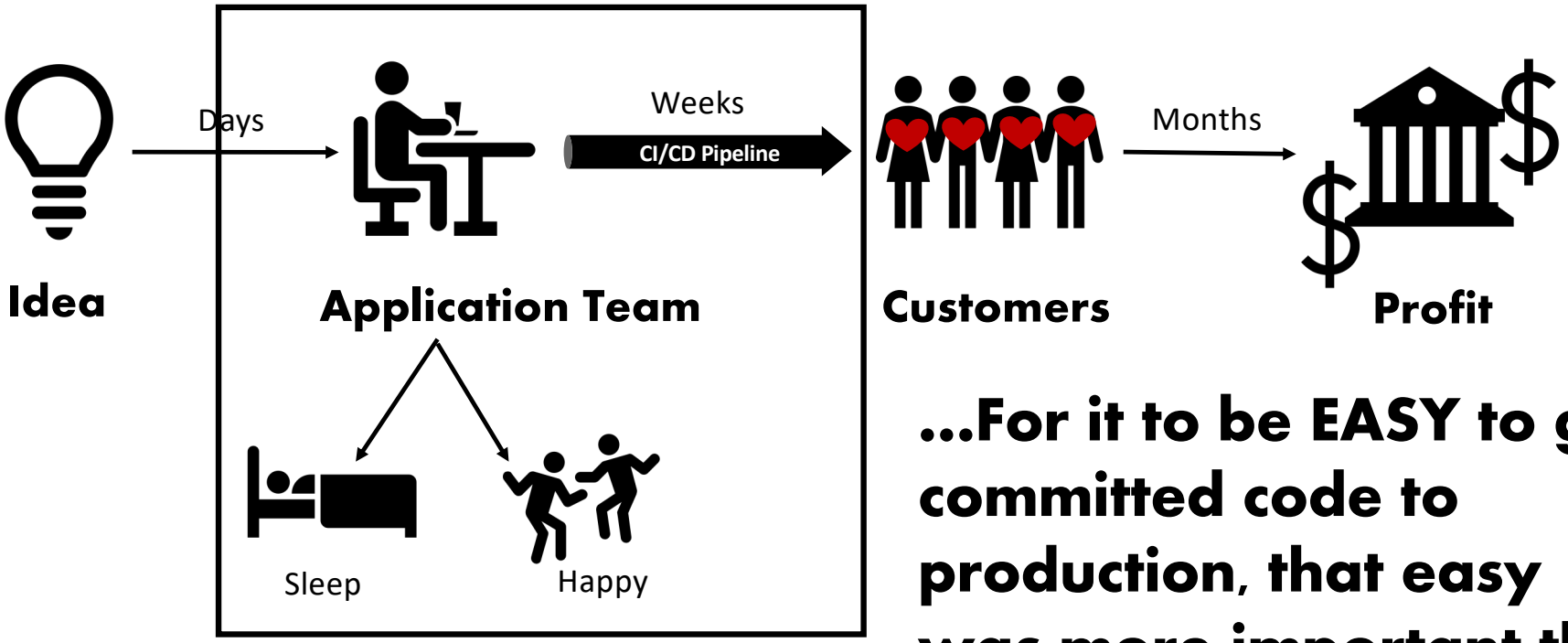
… that by adopting DevOps application teams will be able to deploy multiples times a day and rapidly deliver innovation

I have learned there are the multiple and complex factors that influence a DevOps adoption and make it challenging

- Lack of leadership buy-in at all levels
- Tech stack complexity
- Atypical value streams
- Heavy regulatory requirements
- Poor implementation of a full-stack “DevOps” teams
- Team churn, stability

So be pragmatic by understanding that not every organization can be Unicorns and embrace the Donkey

Every organization I have worked with in the last seven years has really wanted one fundamental thing…



…For it to be EASY to get committed code to production, that easy was more important than fast

The following case studies will illustrate how different teams wanted different things and faced different challenges…

1. Handoff Hell
2. Testing Turmoil
3. DevOps is No Science
4. Infrastructure Impediments

…And had to be pragmatic about their DevOps adoption

Case #1: "Handoff Hell"

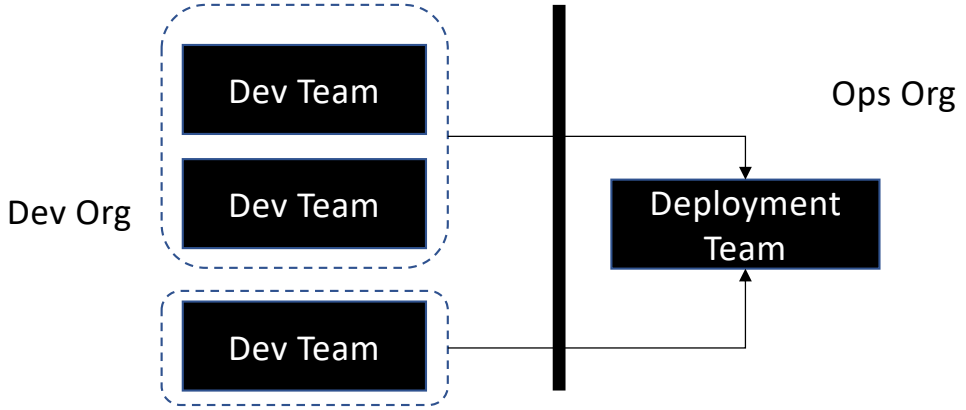
Corporate IT department whose development teams adopted agile and were frustrated with the manual and resource intensive deployment process that had to be executed off hours and often failed from mismatched environments.

- ✓ Leadership buy-in at all levels
- ✓ Tech stack simplicity
- ✓ Typical Value Stream process
- ✓ Lightweight regulatory requirements

  Application of a full-stack team
- ✓ Team stability

Easier Requirements:

- Release every sprint
- Automated build
- Automated deploy the same way to every environment
- Automated rollback on failed deploy



Case #1: "Handoff Hell"

Results:

- 9 months to adopt for 1 single app
- After 18 months over 75+ applications being built and deployed
- Acceptance testing is still manual but performed in smaller batches
- Deployments at 6am every day

Sleep

Happy

Pragmatic Point: A corporate IT department doesn't really need to deploy multiple times a day and have continuous deployments

Case #2: "Testing Turmoil"

Customer-facing application team developing a mobile and web application working with real-time streaming data and decisions. Seeking to federate the platform but it took three weeks to get a single line-of-code out the door due to a manual functional and performance testing process

- ✓ Leadership buy-in at all levels AND the business

  Tech stack simplicity

- ✓ Typical Value Stream process

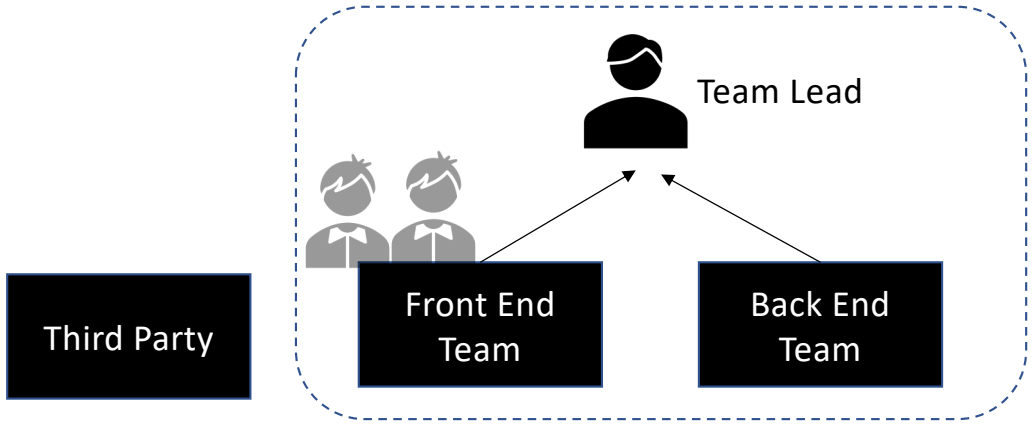
  Lightweight regulatory requirements

- ✓ Application of a full-stack team

  Team stability, availability

Easier Requirements:

- Continuous Integration and Delivery of application code
- Release several times a day
- Zero-downtime, stateless deployment



Case #2: "Testing Turmoil"

Results:

- 3 months to adopt CI/CD through lower environments for app code changes
- Adopted Test Data Management and Service Virtualization for streaming API that reduced the need to coordinate testing with third party
- From 3 weeks to 3 hours or less to get code tested
- Another 2 -3 months to perform zero-downtime, stateless anytime deployments

Sleep

Happy

Pragmatic Point: Having to coordinate with dependent systems to test and operate your application puts a damper on things

Case #3: "DevOps is No Science"

Machine Learning organization developing machine learning models that took months and many people to get tested, deployed and promoted to higher environments only to have to redo the model when new data or behaviors changed how the model worked.

- ✓ Leadership buy-in at all levels AND the business

  Tech stack simplicity

  Typical Value Stream process

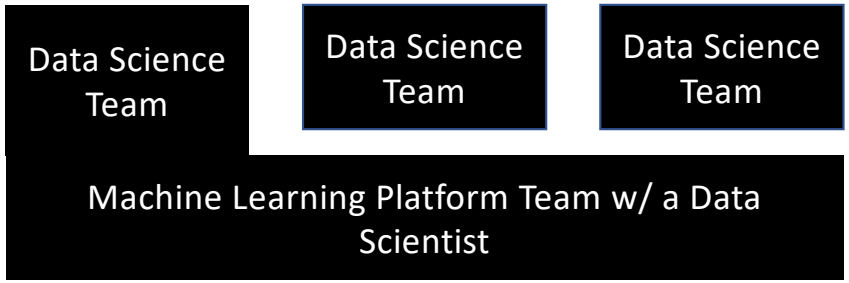
  Lightweight regulatory requirements

- ✓ Application of a full-stack team

  Team stability, availability

Easier Requirements:

- CI/CD of ML Models through multiple non-PROD environments
- Containerization and Cluster Management
- Realistic Training environment using PROD data
- Host multiple types of models and versions of same model



Centralized Platform Team

Case #3: "DevOps is No Science"

Results:

- Federated, Managed ML Platform
- ~12 weeks to get CI/CD through two lower environments deploying with Docker Containers into a Kubernetes cluster
- ~1 year for a fully managed, resilient platform

Sleep

Happy

Pragmatic Point: Building a new and scalable platform for a newer technology with newer technologies for a new team is really challenging

Case #4: ”Infrastructure Impediment”

Web application analyzing real-time streaming data for duplicate transactions analysis. App came from ”the lab” and was developed rapidly and was brittle. Manual machine image rehydration process and maintenance of infrastructure disrupted the development flow and created downtime for maintenance

- ✓ Leadership buy-in at all levels

  Tech stack simplicity

- ✓ Typical Value Stream process

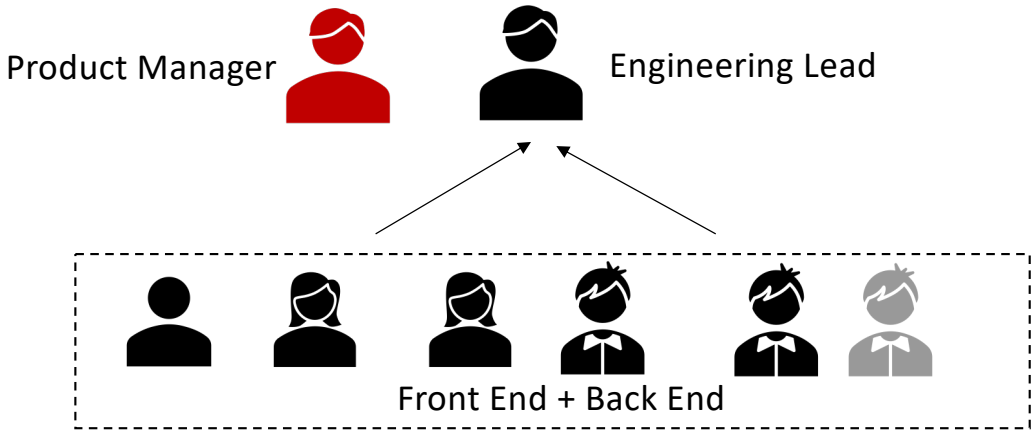
  Lightweight regulatory requirements

- ✓ Application of a full-stack team

  Team stability, availability

Easier Requirements:

- Automate Machine Image rehydration process
- Automated provisioning and configuring
- Zero-downtime deploys



Case #4: "Infrastructure Impediment"

Results:

- Abandoned the effort two weeks in
- Priorities changed and the focus shifted
- There was a non-associate labor cut and the contractor was released

Pragmatic Point: Nothing like a good priority shift, budget cut and some team churn to undermine your efforts

What will slow you down?

Attribute	What makes things harder
Leadership Buy-In	Lack of leadership buy-in at all levels especially. From middle management
Tech Stack Simplicity	The more complex the tech stack and configuration the harder it is. E.g. Third party, newer technologies, high availability requirements
Typical Value Stream	The more environments and hand-offs in the process to go from code to commit
Light Regulatory	Personal and financial/account information protection and constantly changing regulatory requirements
Full-Stack Team	Not having dedicated people to DevOps where they have to choose between feature development and platform development
Team Stability	Team churn from development. Program rotations, contractors and regular churn

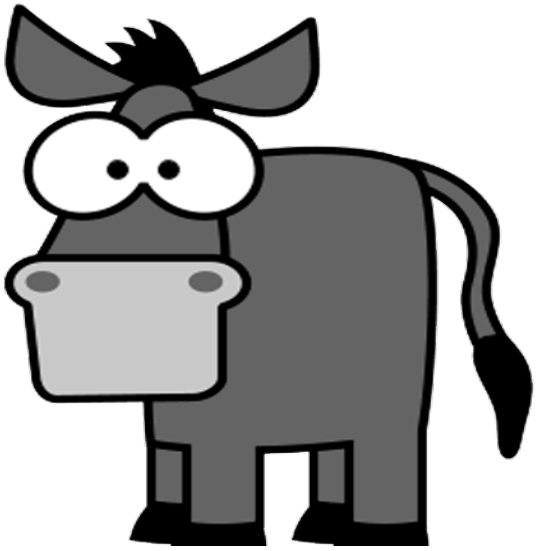
The organization with the most pragmatism, patience, perseverance and perspiration sleeps and is happy

A Unicorn and a Donkey meet at a DevOps conference...

DevOps Unicorn

We deploy code to production multiple times a day. How many times a day does your team deploy?

We reduced our cycle time by 75% and eliminated off hours deployments.
I sleep and I am happy.



DevOps Donkey