# *Faster, Cheaper, Safer*

## *Secure Microservice Architectures using Docker*

**Adrian Cockcroft @adrianco**
**Technology Fellow - Battery Ventures**
June 2015

BV Battery Ventures

*Key Goals of the CIO?*
*Align IT with the business*
*Develop products faster*
*Try not to get breached*

# Security Blanket Failure

*Insecure applications hidden behind firewalls make you feel safe until the breach happens…*

http://peanuts.wikia.com/wiki/Linus'_security_blanket

# *What needs to change?*

*Developer responsibilities: Faster, cheaper, safer*

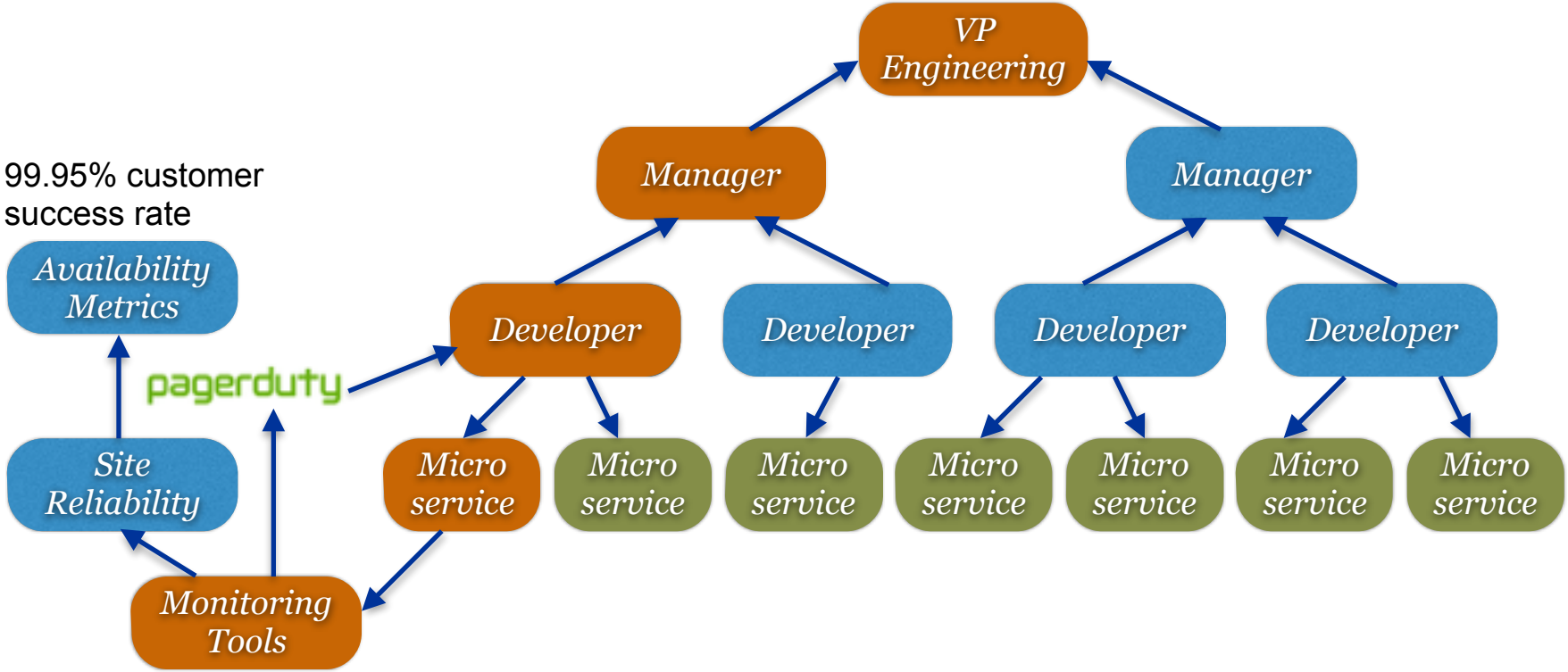
*Faster - Agile*

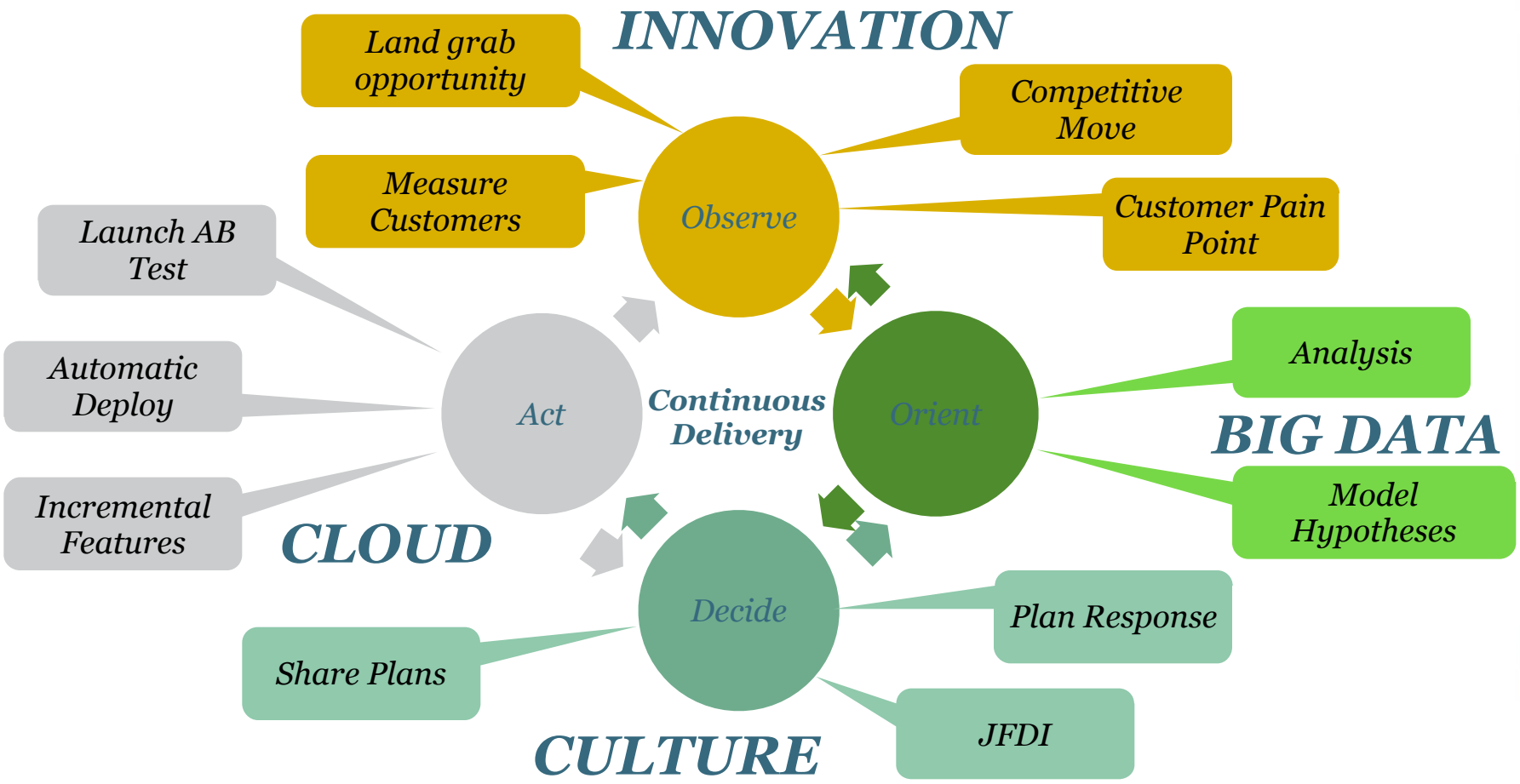
*“You build it, you run it.”*

*Werner Vogels 2006*

*DevOps*
*Continuous Delivery*
*No meetings, no tickets*
*Self service tools and APIs*

# Run What You Wrote




INNOVATION
Land grab opportunity
Competitive Move
Measure Customers
Observe
Customer Pain Point
Launch AB Test
Automatic Deploy
Act
Continuous Delivery
Orient
Analysis
BIG DATA
Incremental Features
Model Hypotheses
CLOUD
Decide
Plan Response
Share Plans
JFDI
CULTURE

# Low Cost of Change Using Docker

**Developers**
- Compile/Build
- Seconds

docker

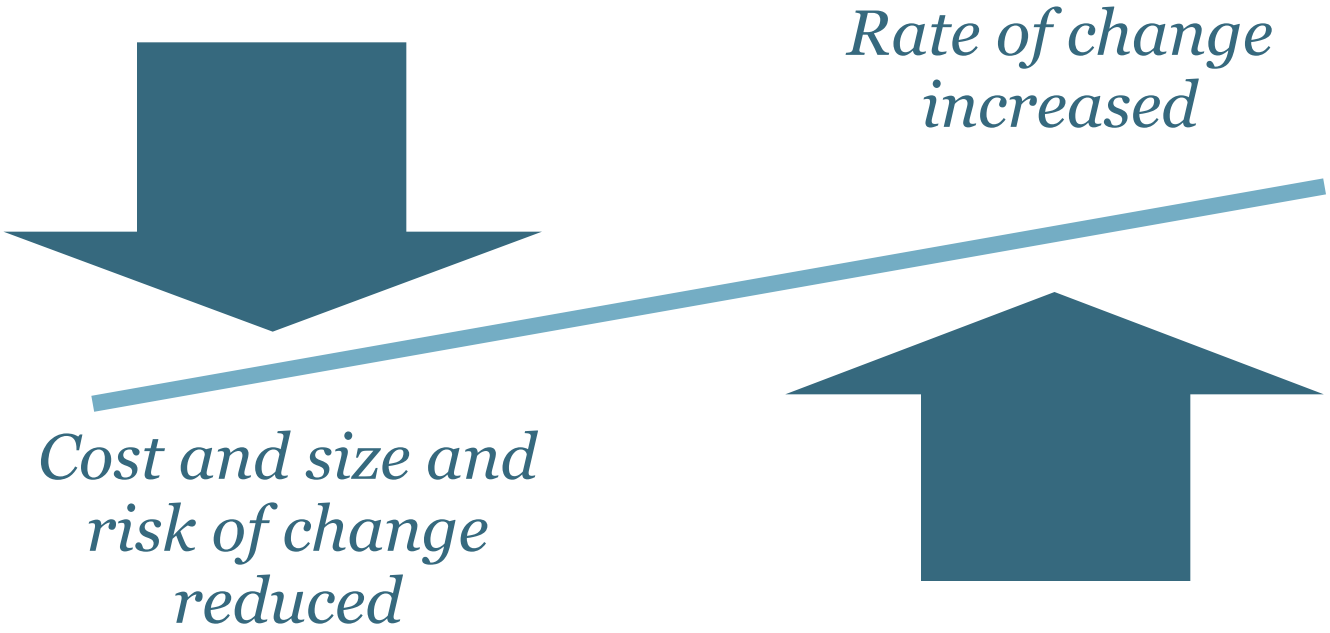
**Extend container**
- Package dependencies
- Seconds

**PaaS deploy Container**
- Docker startup
- Seconds

*Fast tooling supports continuous delivery of many tiny changes*

*Change One Thing at a Time!*

# What Happened?



# *Cheaper - Lean*

*“Freedom and responsibility”*

*Reed Hastings 2009*

*Fail early and often*
*Instrument everything*
*Hypothesis driven development*
*Efficient and autoscaled*

*Efficiency Gains:*
*Virtualization consolidates CPUs*
*Docker consolidates CPU and RAM*

*With Docker a test environment should only exist for the few seconds it takes to run a test*

*Autoscale production to consume just the resources you need, by the second*

# *Safer - Rugged*

# *“Developer Defined Infrastructure”*

*Jerry Chen 2015*

*What can developers do about the threats?*

# *External Threats*

*Build using penetration test tools*
*Manage image supply chain*
*Hardened immutable services*
*Service roles and security groups*

# *Internal Threats*

*Assume employees are compromised*
*User roles, minimum privilege*
*Audit logs for everything*
*Encrypt data at rest*

# *Patterns and practices*

# In Production





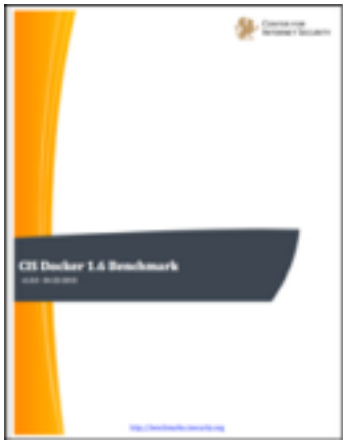


https://www.docker.com/resources/usecases/

*and many more....*

# *Patterns and practices*

# Best Practices





https://blog.docker.com/2015/05/understanding-docker-security-and-best-practices/

*Immutable deployments*

*Automated penetration testing*

*Role based identity and access*

*Trusted container supply chain*

*Continuous audit*

# *Workloads*

# Need for Speed

*CPU and IO Intensive workloads*
*Hadoop, streaming, datastores*
*Bare metal for efficiency*
*Well isolated for security*

# Cutting the Cost

*Many similar containers per VM*
*Saving on RAM, oversubscribe CPU*
*Deploy with Swarm, Mesos, ECS, GKE*
*VM based single tenant security*

# Playing it Safe

*One critical container per VM*
*Extra security for exposed services*
*Deploy as immutable VM image*
*Docker adds to VM security*

# Tooling for Docker


Google Cloud Platform

tutum


RANCHER

APCERA

MESOSPHERE

*and many more....*

# *Docker in Production*

*2014 - DIY frameworks*

*2015 - Hardening and best practices*

*2016 - Mature production tooling*

Disclosure: some of the companies mentioned may be Battery Ventures Portfolio Companies
See www.battery.com for a list of portfolio investments

# *Thanks !*

*Continue the discussion on Twitter @adrianco*

**Adrian Cockcroft**
**Technology Fellow - Battery Ventures**
June 2015

BV Battery Ventures