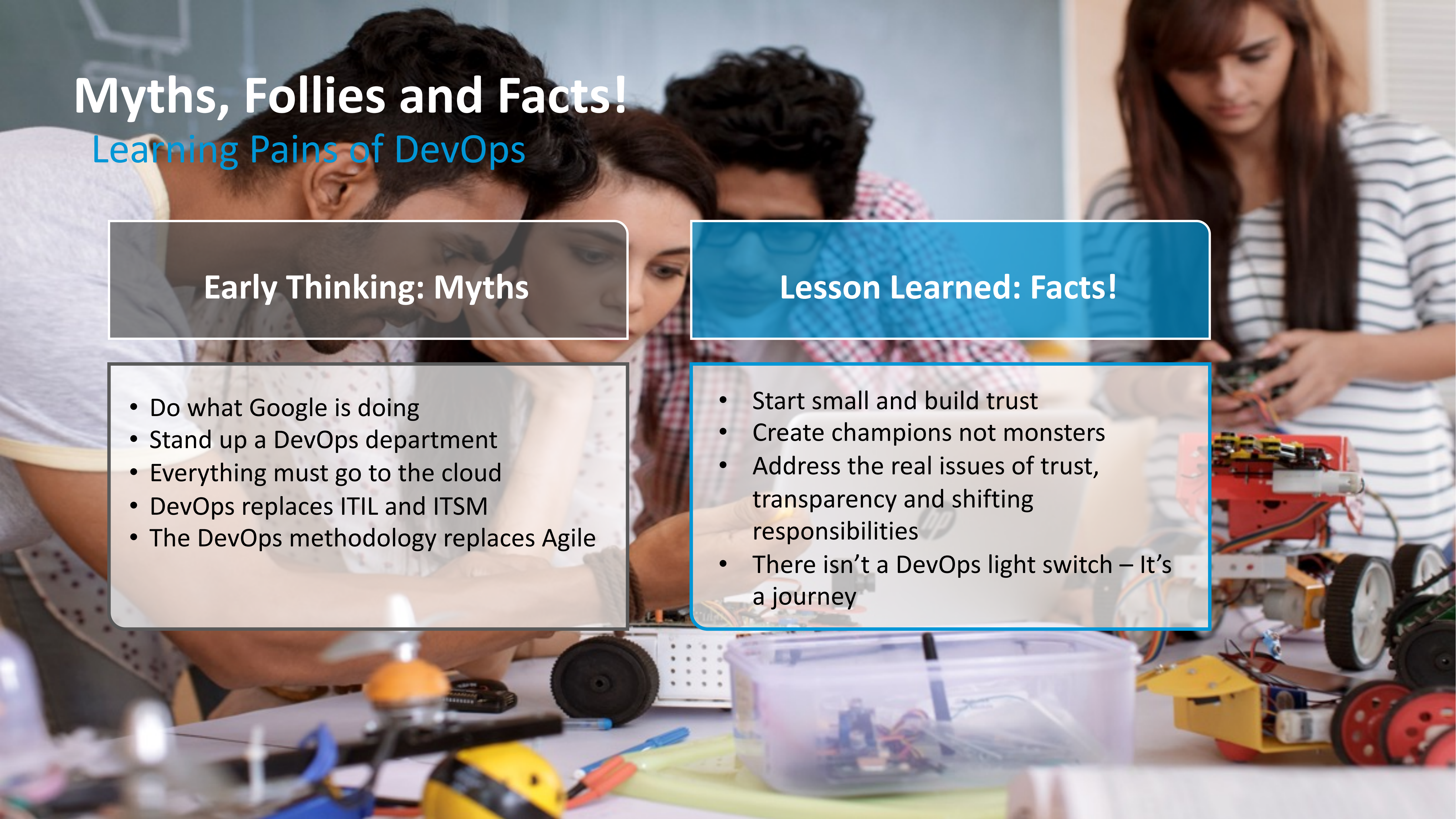
Self-Healing And Monitoring in a DevOps World

Lucas Gravley / October, 2015



Myths, Follies and Facts!

Learning Pains of DevOps

Early Thinking: Myths

- Do what Google is doing
- Stand up a DevOps department
- Everything must go to the cloud
- DevOps replaces ITIL and ITSM
- The DevOps methodology replaces Agile

Lesson Learned: Facts!

- Start small and build trust
- Create champions not monsters
- Address the real issues of trust, transparency and shifting responsibilities
- There isn't a DevOps light switch – It's a journey

Our DevOps Vision

Increase Visibility and Alignment with Stakeholders

Improve user experience with continuous feedback and analysis loops

- Prioritize business requirements
- Monitor and analyze everything
- Provide visibility across teams

Moving at the Speed of the Business

Enabled via automation of the integrated build, test and deployment process

- Automate test infrastructure
- Maximize test coverage
- Improve application quality, performance and security
- Break work into smaller increments

Decrease the Cost of Building Applications

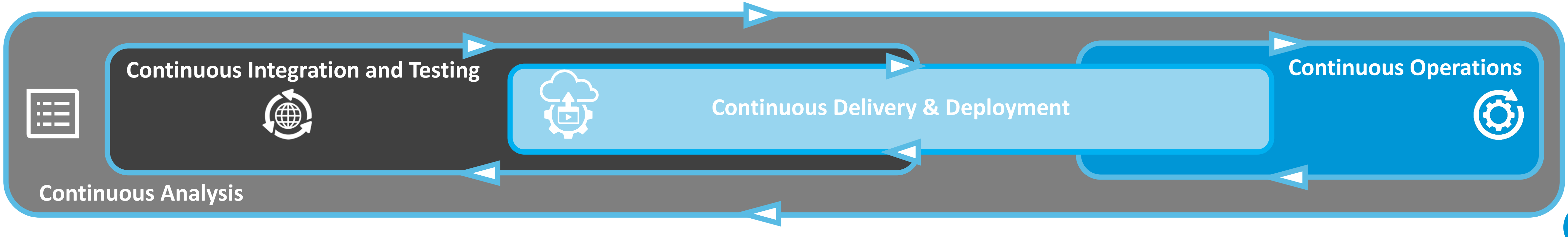
Increase developer and operational velocity by managing your infrastructure as code

- On demand environments
- Create and manage via IaaS
- Minimize downtime
- Increase reliability of releases

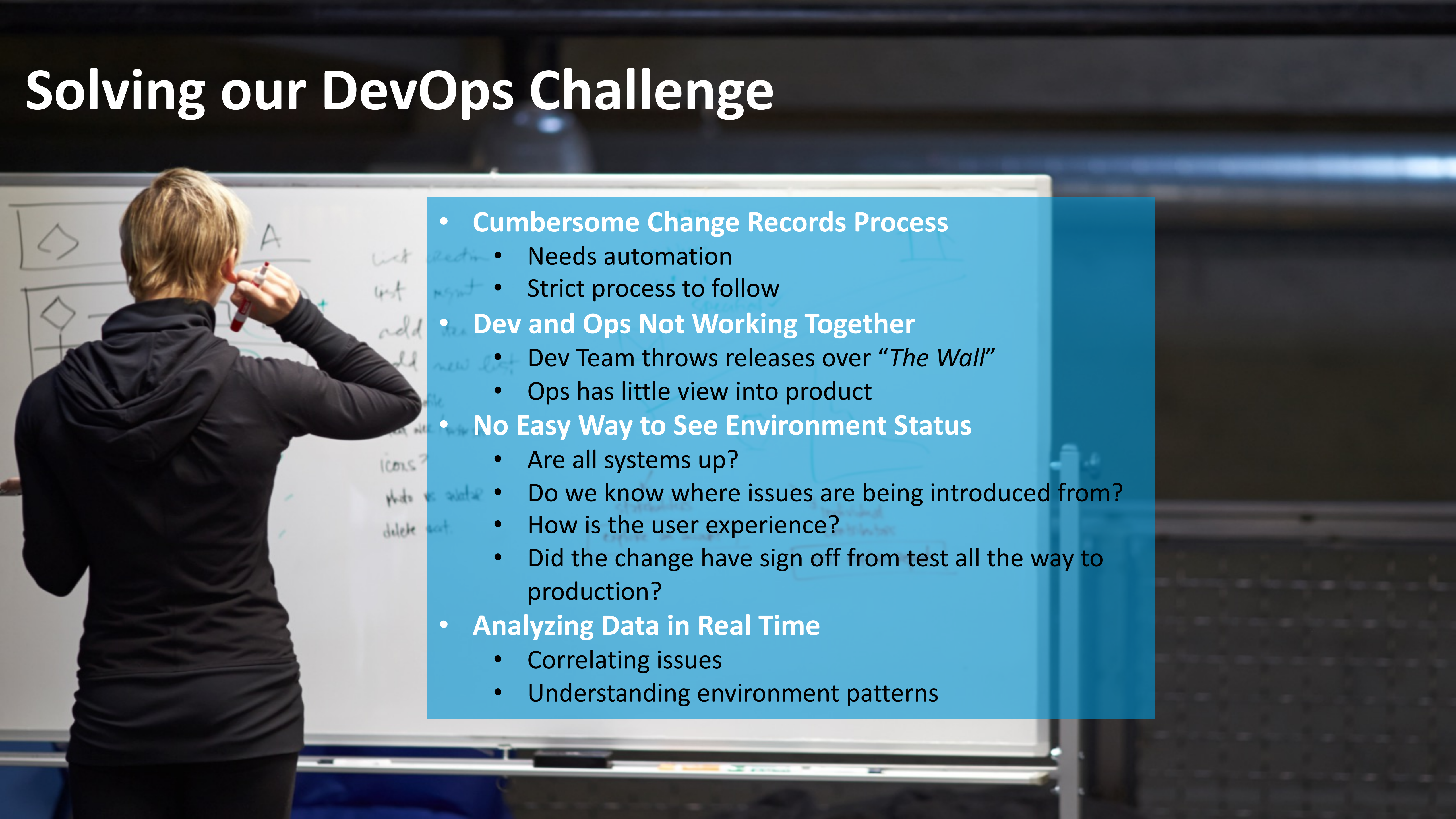
Improve Quality and Performance

Manage software and hardware changes in a non-disruptive way to the end users

- Extract, correlate and exploit 100% of metrics
- Real Time and historical analytics
- Improved monitoring and self healing


Continuous Integration and Testing
Continuous Delivery & Deployment
Continuous Operations
Continuous Analysis

Solving our DevOps Challenge



- **Cumbersome Change Records Process**
  - Needs automation
  - Strict process to follow
- **Dev and Ops Not Working Together**
  - Dev Team throws releases over “*The Wall*”
  - Ops has little view into product
- **No Easy Way to See Environment Status**
  - Are all systems up?
  - Do we know where issues are being introduced from?
  - How is the user experience?
  - Did the change have sign off from test all the way to production?
- **Analyzing Data in Real Time**
  - Correlating issues
  - Understanding environment patterns

What to Monitor?

Anything and Everything

Designed for Total Coverage

Monitoring

- **Out of the Box and Custom**… Turn it on and watch it run
- **Key Components**… Network, Services, Applications, User experience
- **Extensible**…Create custom monitors for your backlog issue

Data Mining

- **Insight**…Be able to diagnose monitoring data
- **Understanding** …Actively use the data

Automation

- **Speed**… Self deployment of monitoring
- **Elegance**… Automate install of agents
- **Encompass**… Includes DEV, TEST, and PRO

How does Self-Healing work?

Alerts and Updates

Scripts

- **Simple Scripting…** Bash/Perl/PowerShell/Coffee
- **Complex Scripting…** Call to Automation Environment

Reproducible API

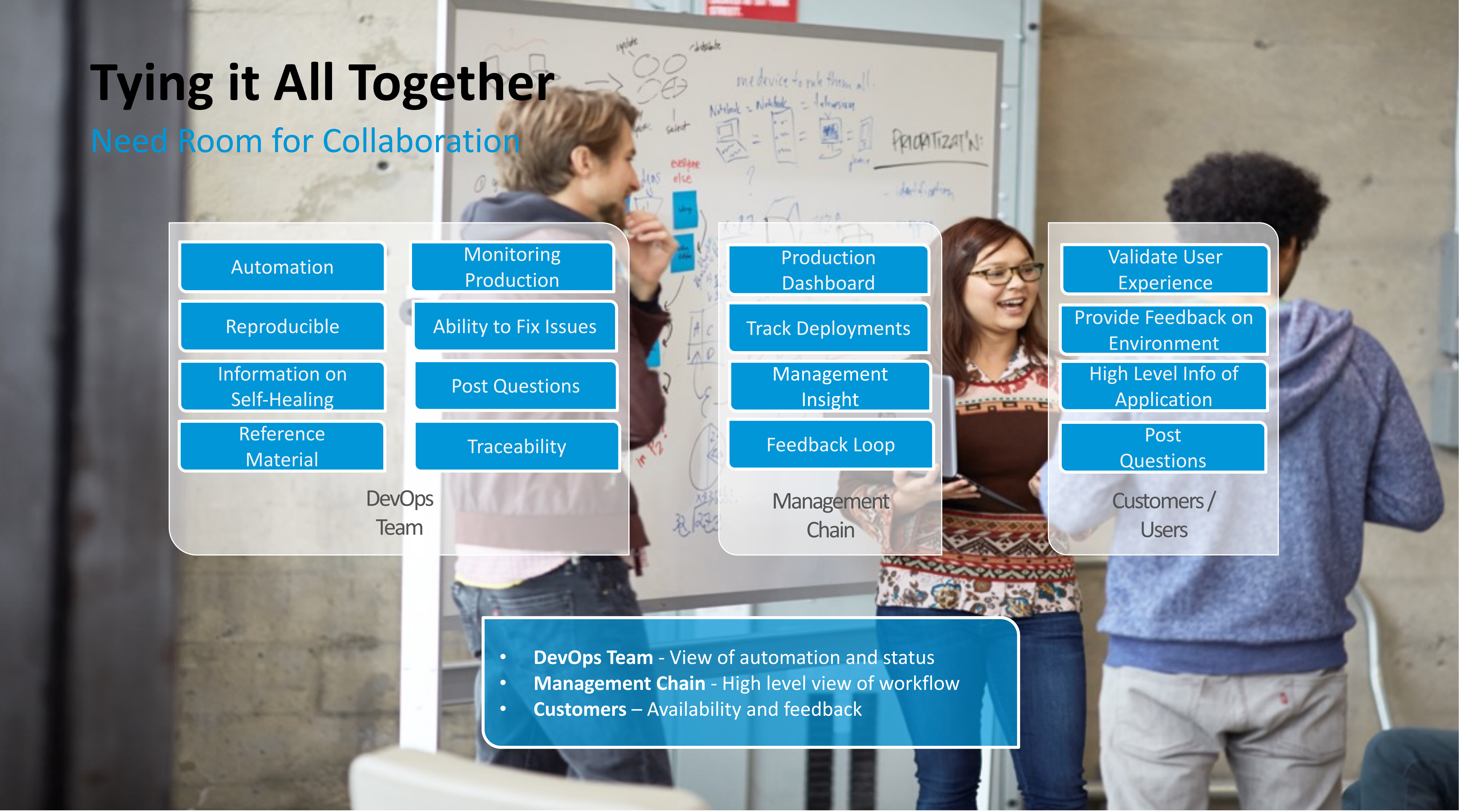
- **Ability to Leverage**… No one needs to reinvent the wheel
- **Alerts**… Notification to the team on all types of mediums

Integrated Pipeline

- **Unified**… Documents the Process for team
- **Continuous**… Picks up monitoring issues & tries to resolve
- **Vigilant**… Proactive self-healing
- **Notification**… Informs change process pipeline

Tying it All Together

Need Room for Collaboration



DevOps Team

- Automation
- Reproducible
- Information on Self-Healing
- Reference Material
- Monitoring Production
- Ability to Fix Issues
- Post Questions
- Traceability

Management Chain

- Production Dashboard
- Track Deployments
- Management Insight
- Feedback Loop

Customers / Users

- Validate User Experience
- Provide Feedback on Environment
- High Level Info of Application
- Post Questions

- **DevOps Team** - View of automation and status
- **Management Chain** - High level view of workflow
- **Customers** – Availability and feedback

What Is ChatOps?

Putting Tools Into the Conversation

#ChatOps

ChatOps is a term coined by Github to describe their growing culture of "Putting tools in the middle of the conversation"

The Idea

- **Assist Team**… Bring teams together
- **Ease of Use**… Simplicity of tool
- **Bots**… As many as you need

ChatOps

Persistent Chat Working for You

Available Anywhere

- **Desktop**… Real time notification
- **Mobile**… Any device and carrier
- **Browser**… Any industry standard browser

Visibility for all Parties

- **Collaboration**… All teams sharing data and information
- **Closing the Loop**… End users feedback loop
- **Auditability…** being able o go back

Bot Interaction

- **Powerful Teammate**… Able to train, supply info, and solve issues, validate delegation of powers
- **Graphing**… Overview of environment in seconds
- **Running Pipelines**… Transparency through the service lifecycle

How Much is This Going Cost?

Deploying DevOps is a Function of Doing Business

Costs to …..eliminate waste, stream delivery pipelines and standard tool chains

- Standardized tooling and business process: **establish your pipelines**
- **Educate**
- Build expertise in functional teams and leverage a few **champions**
- Cost of removing waste – with full transparency take a good look at the current practices and **eliminate waste**

Show Me the Savings

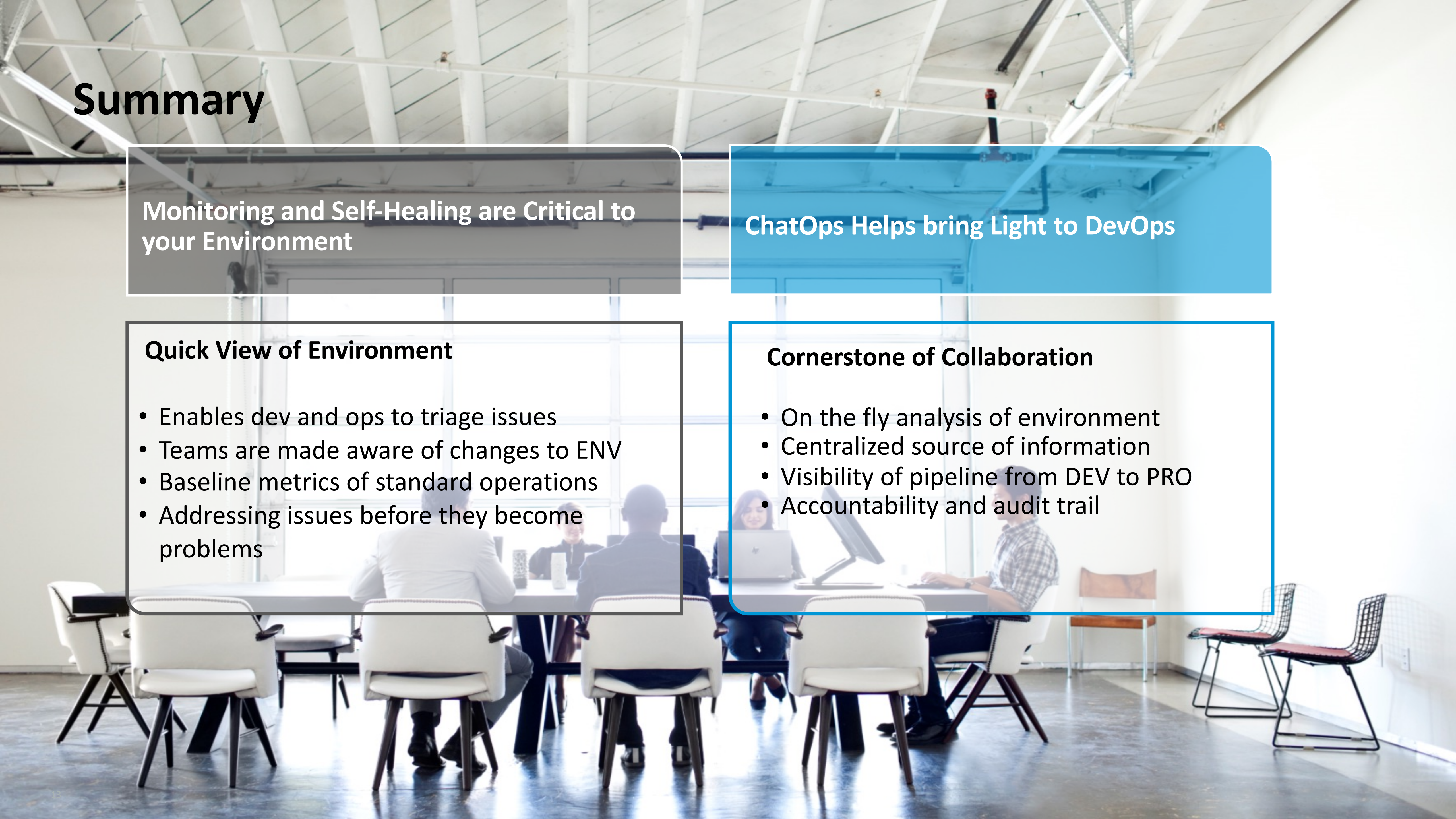
Deploying DevOps is a Function of Doing Business

- CI/CD investments **reduce cycle time**
- Excessive operational **overhead eliminated**
- **Cost avoidance** thru real-time insight
- **Higher quality solutions**
- Standardized tool chains **eliminate technical debt and uncontrolled spend**
- Collaborative teams yielding high **productivity**
- **Leverage gained** velocity to further innovate
- Embrace hackathons, meet ups and collaboration – **get some free stuff!**

Live Demo

- Persistent Chat Interaction
- Lookup Data
- Common Commands
- Graphing Data
- Dash Boarding
- Automation
- Follow Self-Healing
- Upgrading Instance

Summary



Monitoring and Self-Healing are Critical to your Environment

Quick View of Environment

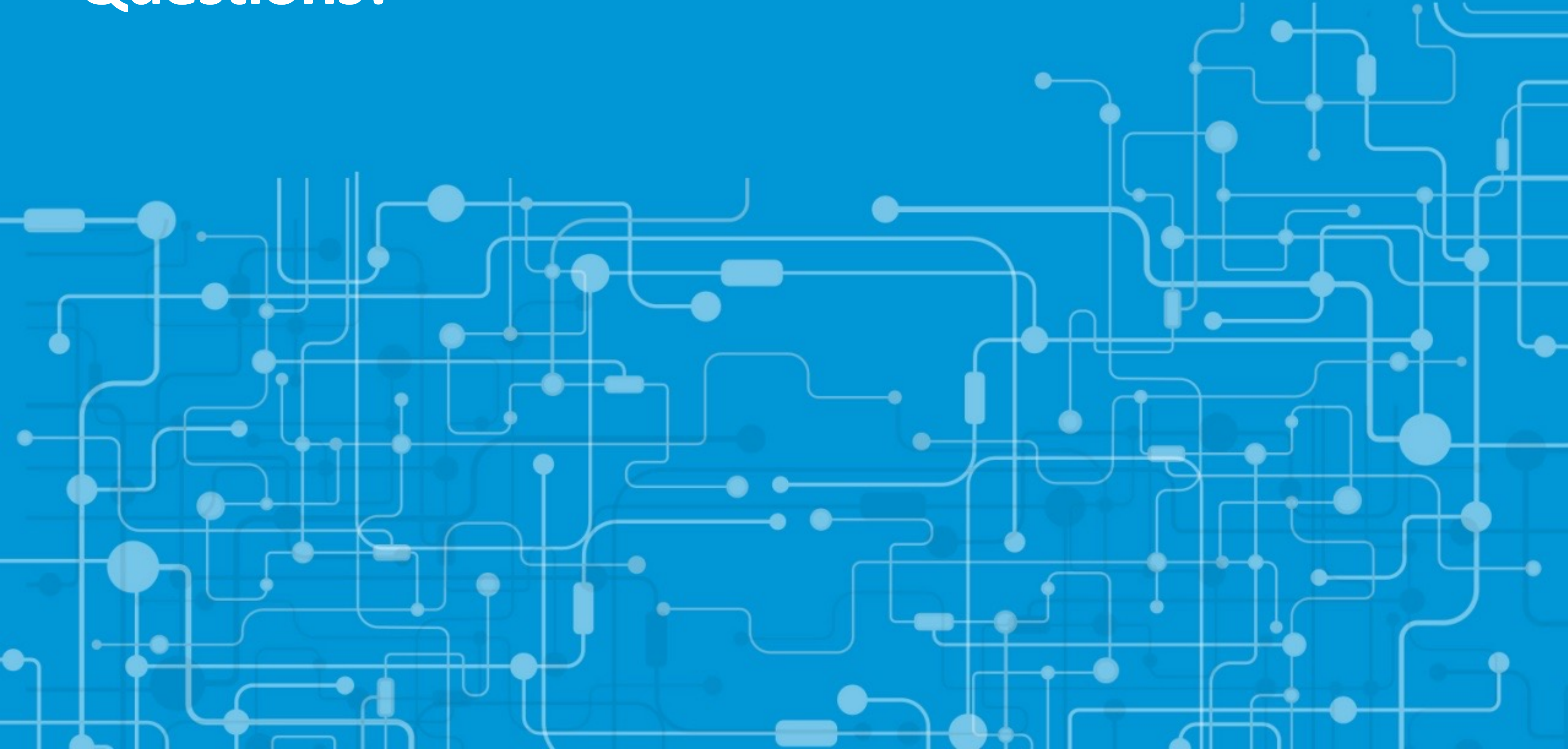
- Enables dev and ops to triage issues
- Teams are made aware of changes to ENV
- Baseline metrics of standard operations
- Addressing issues before they become problems

ChatOps Helps bring Light to DevOps

Cornerstone of Collaboration

- On the fly analysis of environment
- Centralized source of information
- Visibility of pipeline from DEV to PRO
- Accountability and audit trail

Questions?



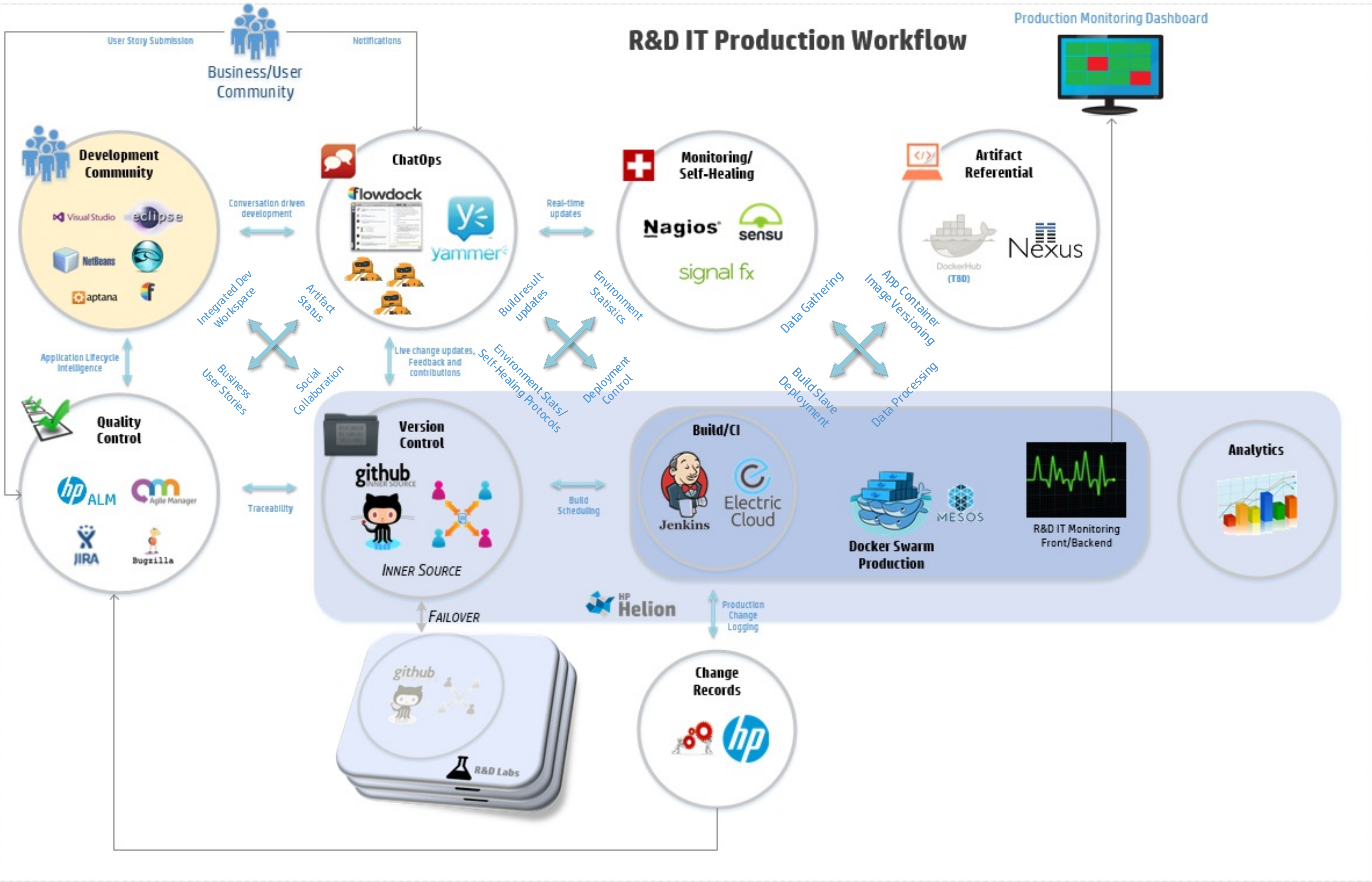
Thank You

Lucas.Gravley@hpe.com

Operations Analytics
18 10:45
19 6:00
Application Transaction Response Time Trends
Application Counts per Location
Applications in Locations

BackUp Slides

R&D IT Production Workflow


Production Monitoring Dashboard
User Story Submission
Business/User Community
Notifications
Development Community
Visual Studio
eclipse
NetBeans
aptana
ChatOps
flowdock
yammer
Monitoring/ Self-Healing
Nagios
sensu
signal fx
Artifact Referential
DockerHub (TBD)
Nexus
Conversation driven development
Real-time updates
Integrated Dev Workspace
Artifact Status
Business User Stories
Social Collaboration
Build result updates
Environment Statistics
Environment Stats/ Self-Healing Protocols
Deployment Control
Data Gathering
App Container Image Versioning
Build Slave Deployment
Data Processing
Application Lifecycle Intelligence
Live change updates, Feedback and contributions
Quality Control
ALM
Agile Manager
JIRA
Bugzilla
Traceability
Version Control
github INNER SOURCE
INNER SOURCE
Build Scheduling
Build/CI
Jenkins
Electric Cloud
Docker Swarm Production
MESOS
R&D IT Monitoring Front/Backend
Analytics
HP Helion
FAILOVER
github
R&D Labs
Production Change Logging
Change Records