

DevOps

Changing the way you deliver software

Who is this guy?

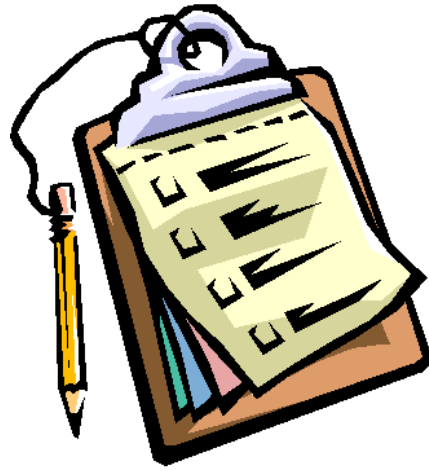
- ▶ John O'Bryant
 - ▶ Director of Application Development at HealthSystems
 - ▶ IT and Test Automation for over 17 years
 - ▶ Focus on building automated test suites and leadership roles
 - ▶ My vision at work: "I want to lead change and elevate quality within our company by inspiring, helping and preparing others"



To simplify complexities for each customer

- ▶ We manage the cost and utilization of pharmacy and ancillary medical benefits for workers' compensation insurance payers and ensure that injured worker patients receive the most appropriate care and treatment.

Agenda



- ▶ Definition of DevOps
- ▶ Delivery Pipeline
- ▶ Automation
- ▶ Metrics
- ▶ Culture
- ▶ Benefits
- ▶ Leadership
- ▶ Healthsystems DevOps
- ▶ Material



What is DevOps?



An approach, culture, philosophy, or movement aimed at improving

- ▶ Collaboration
- ▶ Communication
- ▶ Integration

Between Dev and Ops to deliver better software, faster

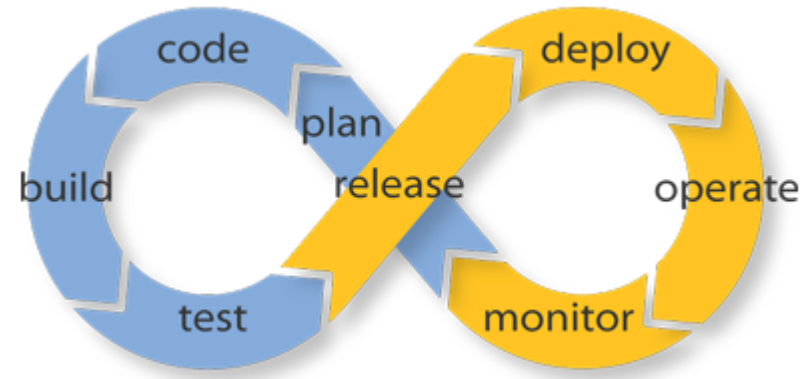
Deliver High Quality Working Software Faster

What is DevOps?



- ▶ Agile changes the way you plan, DevOps changes the way you deliver
- ▶ Apply agile principles to the planning process and use DevOps to address the basic principle of being able to economically release smaller batches of changes on a more frequent basis.
- ▶ If your development processes don't allow you to economically release small batches of new functionality, then you are not able to realize the full benefits of software. You also lose the benefits of receiving rapid feedback from your customers or market to see if the new capabilities are meeting expectations and understand outcomes to inform next steps.

CD (Continuous Delivery)



- ▶ Delivery Pipeline
- ▶ Continuous Integration
- ▶ Scripted Environments
- ▶ Scripted Deployment
- ▶ Evolutionary Database Design
 - ▶ Manage DB changes to ensure that schema changes won't break your application, tools that puts the DB under revision control and can automatically update the DB in any environment to a desired version or configuration.
- ▶ Test Automation
- ▶ Orchestrator
 - ▶ Tool that coordinates all of the automation

Agenda

- ▶ Definition of DevOps

- ▶ **Delivery Pipeline**

- ▶ Automation

- ▶ Metrics

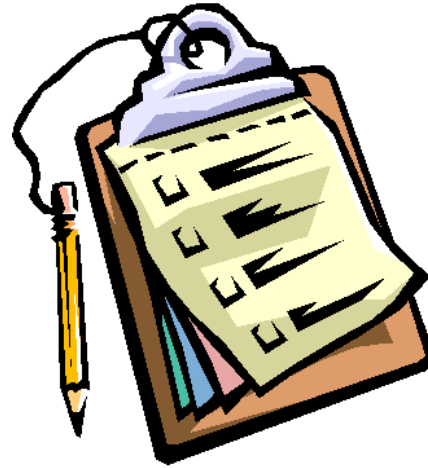
- ▶ Culture

- ▶ Benefits

- ▶ Leadership

- ▶ Healthsystems DevOps

- ▶ Material



What does your software delivery pipeline look like?

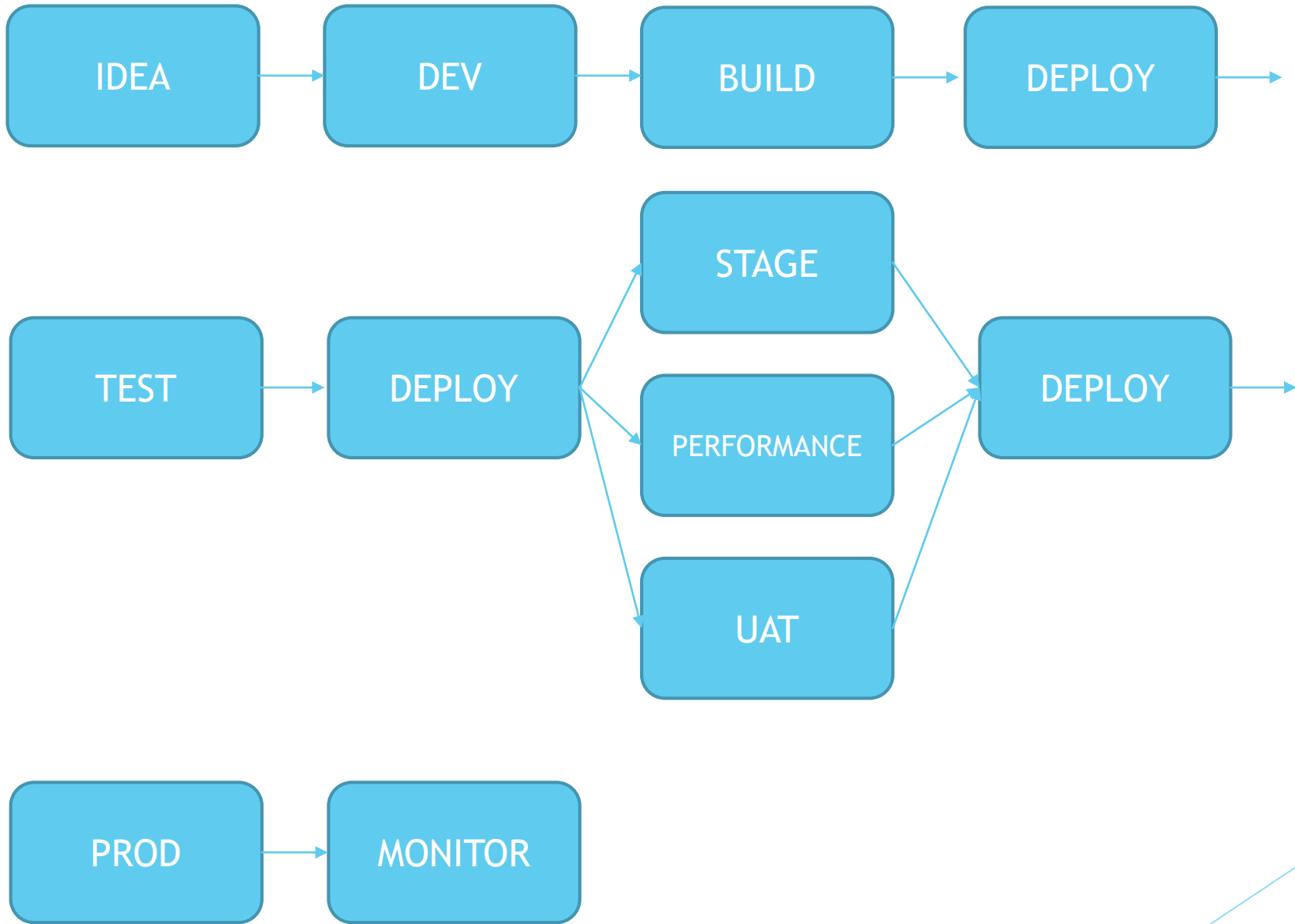
A.



B.



(most people live here)



Delivery Pipeline



- ▶ How many manual steps are there along the way?
- ▶ How often does code flow through your pipeline?
- ▶ How often do you deploy to production?
- ▶ How much could be automated and made repeatable to speed that pipeline up?
- ▶ How much does it cost to move through the pipeline?
- ▶ What tools are being used throughout your pipeline?
 - ▶ Testing tools, deployment tools, development tools, etc.

Where to start?

- ▶ Start your focus on improving the build and integration process because this is where you will get the best improvement for productivity.



Release Automation Top Enterprise DevOps Enabler

Which of the following technologies have been most important to your organization's adoption of DevOps?



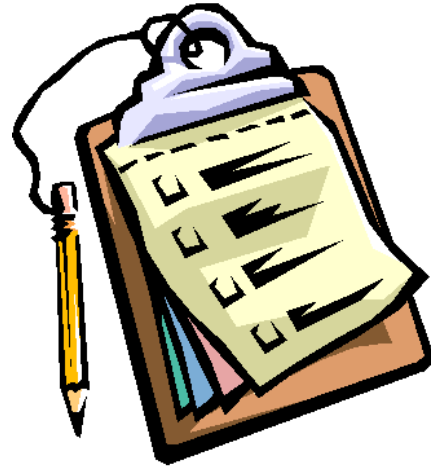
#DOES16



2015 Gartner DevOps Survey
338 IT Business Leaders



Agenda



- ▶ Definition of DevOps
- ▶ Delivery Pipeline

▶ Automation

- ▶ Metrics
- ▶ Culture
- ▶ Benefits
- ▶ Leadership
- ▶ Healthsystems DevOps
- ▶ Material

Infrastructure as Code



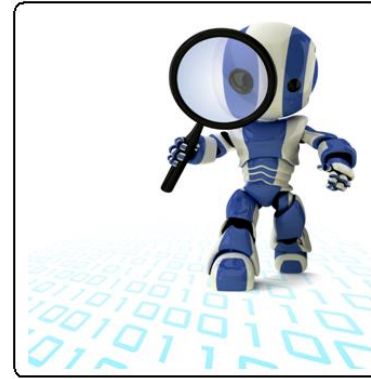
- ▶ Treat all aspects of the software development process with the same rigor as application code.
 - ▶ Process for creating environments
 - ▶ Deploying Code
 - ▶ Managing Databases
- ▶ Automate this process, document and track it in a Source Code Management (SCM) tool just like application code
- ▶ Creates a common definition of an environment across your enterprise (DEV, QA, Stage, Production)
- ▶ Increases communication with development and operation, they need to collaborate on a common script for making changes to the infrastructure

Infrastructure as Code



- ▶ SCM is designed to make it easy to track any and all changes automatically
 - ▶ You can look at a server and see exactly what was changed by who and when
- ▶ Combined with automated testing that can tell you when a system started failing, you can quickly get to the change that caused the problem
- ▶ This gets easier when the cycle time between releases gets smaller and the amount of changes is limited between each release

Automate your testing



- ▶ Manual tests are time consuming, expensive and not repeatable
- ▶ Time it takes to run through a full manual test cycle:
 - ▶ Delays feedback to developers
 - ▶ Slower rework cycles
 - ▶ Reduces the flow through delivery pipeline
- ▶ Manual tests causes you to batch features into major releases, which slows the flow down and makes triage much more difficult and inefficient
- ▶ Providing fast, high quality feedback helps to minimize the waste

Testing approach

- ▶ Development is going to have to modify how they write code so that automated testing will be stable and maintainable
- ▶ Development needs to respond to test failures and keep build stability as their top priority. Testers need to increase the frequency they run tests and ensure testing framework is maintainable.
- ▶ Process of moving to automated tests gating code is going to be a big cultural shift, but is probably one of the most important steps in changing how software is developed
- ▶ Have a smoke test/business acceptance test to includes any defects that have escaped previous test cycles, constantly update this suite to have the most valuable and critical tests. Don't start other testing until these pass.



TESTING

I FIND YOUR LACK OF TESTS DISTURBING.

DIYDESIGN.COM

% of time manually performing tasks

Work Item	High Performer	Mid/Low Performer
Configuration Management	28%	47%
Testing	35%	50%
Deployment	26%	47%

Keeping Your Software Testing Abilities Relevant Today, Tomorrow, and Beyond

By Sunil Sehgal - June 15, 2017



“Quality at speed” is the new mantra in the digital age, and user experience is key.

The software and product development processes have already changed to better fulfill customer expectations by embracing agile and DevOps in order to adapt to new requirements. What can testers do to keep up with their development peers?

I have some ideas about what testers can learn, what skills we can add, and what processes we can start doing in order to continue delivering quality at speed today, tomorrow, and further into the future.

Today

Adopting test automation, if you haven't already, should be your first step, particularly if your development team is agile.

Automation allows you to accomplish regression testing and new feature testing quickly within the current agile sprint. The tools and methodologies you use should pull requirements from the development team's user stories and acceptance criteria so that you can build the automation framework and scripts in parallel to development, saving time and ensuring quality.

To keep pace with development, talking to your team about adopting agile testing and using test automation is something you can do today.

In the Next Year or Two

This will be when DevOps implementation becomes mainstream for testing. DevOps is primarily about building a technology layer for implementing continuous integration and continuous delivery, which needs test automation to happen. With a firm foundation in

test automation already, testers can start exploring continuous testing made possible with DevOps.

To prepare for the immediate future, I think it would be a good idea to start reading and taking some courses in order to become a DevOps implementation expert with a focus on testing. Think about building a technology layer for testing that integrates with development tools and enables continuous deployment and testing.

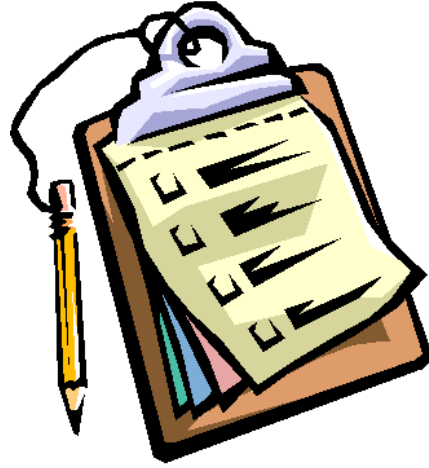
Beyond

Artificial intelligence and machine learning are on the horizon. As an extension of test automation, we can expect to see more use of testing bots, particularly for finding bugs, functional testing, and click-through pass/fail tasks.

Some people think that with the advent of AI, machine learning, and testing bots, human testers will become obsolete, but in my view, there is a huge opportunity for testers willing to take on new challenges. Start playing around with writing a bot to validate tests so you become familiar with quality engineering. You can also start delving into exploratory testing and user experience, because there won't be a substitute for human testers when it comes to determining usability.

The role of testers is changing, and quality will become everyone's responsibility. Staying current with the trends and keeping an eye on the future are important to remaining a relevant software tester today and tomorrow.

Agenda



- ▶ Definition of DevOps
- ▶ Delivery Pipeline
- ▶ Automation

▶ Metrics

- ▶ Culture
- ▶ Benefits
- ▶ Leadership
- ▶ Healthsystems DevOps
- ▶ Material

Metrics - Requirements and Planning

- ▶ What % of the organizations capacity is spent on documenting and planning
- ▶ What is the amount of requirement inventory waiting for development
- ▶ What % of requirements are reworked after originally being defined
- ▶ What % of the delivered features are being used by the customer and are achieving the expected business results



Metrics - Environment

- ▶ Time from environment to delivery
- ▶ How frequently new environments are required
- ▶ % of time environments need fixing before acceptance
- ▶ % of defects associated with code vs environment vs deployment vs database vs other at each state of the delivery pipeline



Metrics - Testing

- ▶ Time it takes to run the full set of tests
- ▶ Repeatability of the tests (false failures)
- ▶ % of defects found with unit tests, automated tests, manual tests
- ▶ Time it takes the release branch to meet production quality
- ▶ Approval times
- ▶ Batch sizes or release frequency at each stage



Metrics - Production Release

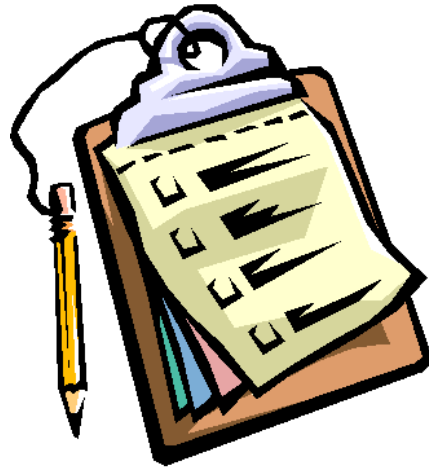
- ▶ Time and effort required to deploy and release into production
- ▶ Number of issues found during release and their source (Code, environment, deployment, test, data, etc)

Metrics - Operation and Monitoring

- ▶ Issues found in production
- ▶ Time to restore service



Agenda



- ▶ Definition of DevOps
- ▶ Delivery Pipeline
- ▶ Automation
- ▶ Metrics

▶ Culture

- ▶ Benefits
- ▶ Leadership
- ▶ Healthsystems DevOps
- ▶ Material

Can your teams say...

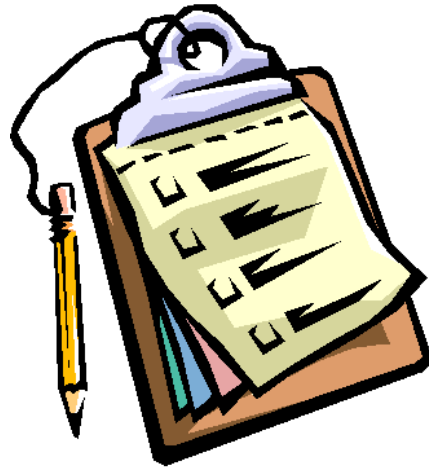
- ▶ Information is actively sought
- ▶ Failures are learning opportunities and messengers of them are not punished
- ▶ Responsibility is shared
- ▶ Cross-functional collaboration is encouraged and rewarded
- ▶ Failure causes inquiry
- ▶ New ideas are welcomed
- ▶ Empowered
- ▶ Can-do-attitude
- ▶ Continuous learning
- ▶ Self awareness and transparency

Culture

- ▶ Create high performing cross functional teams
 - ▶ Leaders
 - ▶ Cultural Anchors
 - ▶ Align with vision and you move them around
 - ▶ Thinkers
 - ▶ Handle complex issues
 - ▶ Doers
- ▶ Be intentional about building a culture, otherwise something you don't like will fill the void
- ▶ Set Vision, Adapt and Reiterate
- ▶ Team Cohesion: Passion, Trust, Honesty, Autonomy



Agenda



- ▶ Definition of DevOps
- ▶ Delivery Pipeline
- ▶ Automation
- ▶ Metrics
- ▶ Culture
- ▶ **Benefits**
- ▶ Leadership
- ▶ Healthsystems DevOps
- ▶ Material

DevOps Transformation Benefits

- ▶ 200x more frequent deployments
 - ▶ On demand (multiple times per day) opposed to 1 per week or 1 per month
- ▶ 24x faster recovery from failures
 - ▶ Mean time to recover is 1 hour, where other performers are 24 hours
- ▶ 3x less likely to have failure
- ▶ Downtime costs are SIGNIFICANTLY lower due to this
- ▶ 2555x shorter lead times
 - ▶ Lead time of 1 hour compared to 1 week - 1 month



DevOps Transformation Benefits

- ▶ 21% Less time spend on Unplanned Work/Rework
- ▶ 44% More time on new work
- ▶ 50% Less time on security issues



Leaders

- ▶ Amazon - Deploys code every 11.7 seconds
- ▶ Netflix - Deploys code thousands of times per day
- ▶ Etsy - went from 4 hour full site deployments 2/week to fully automated delivery pipeline that have 50 deployments per day with fewer disruptions than before
- ▶ Target - transformed their entire culture and powers their Cartwheel application and website



Great! Let's do it then....

70% of change initiatives fail

- *John Kotter*

“70% of all change efforts (in the study) were unsuccessful...”

- *Burke, 2011*

“41% to 93% failure rate among quality improvement initiatives...”

- *Cândido & Santos, 2015*

“Companies engaging in strategic OC initiatives are more likely to fail than to succeed...”

- *Decker, Durand, Mayfield, McCormack, Skinner & Purdue, 2012*

“50% failure rate in multiple studies of all types of change initiatives...”

- *Shin, Taylor, & Seo, 2012*

“Initiatives involving culture change have a 19% success rate...”

- *Smith, 2002*

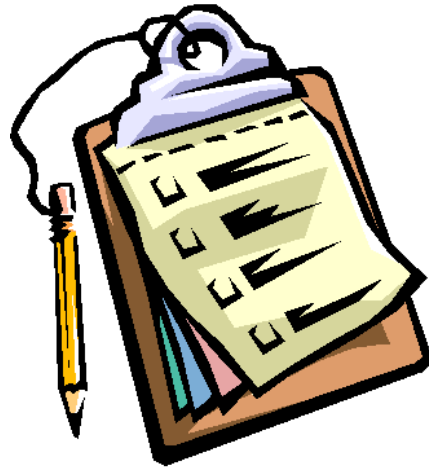
Why does organizational change fail?

- ▶ Resistance to change
- ▶ Low readiness for change
- ▶ Inadequate communications
- ▶ Poor planning
- ▶ Lack of training
- ▶ Institutional inertia (process, structure)
- ▶ Technology gaps
- ▶ Lack of organizational alignment
- ▶ History of change failures
- ▶ Lack of employee involvement
- ▶ Bureaucracy, politics, conflict
- ▶ Poor strategy / wrong change
- ▶ Unrealistic expectations
- ▶ Low transparency and trust

CHANGE IS HARD

John O'Bryant, July 2017

Agenda



- ▶ Definition of DevOps
- ▶ Delivery Pipeline
- ▶ Automation
- ▶ Metrics
- ▶ Culture
- ▶ Benefits
- ▶ **Leadership**
- ▶ Healthsystems DevOps
- ▶ Material

Leaders hold the key to successful change

People are already doing their best; the problems are with the system.

Only management can change the system.

—W. Edwards Deming



Transformational leadership

Intellectual Stimulation (IS)

- Challenge the status quo
- Encourage followers to learn, be creative, explore new ways of doing things
- Empower decision making
- Expect relentless improvement
- Encourage innovative thinking

Inspirational Motivation (IM)

- Articulate a clear vision
- Inspire passion and motivation to achieve goals
- Drive organizational alignment



Individualized Consideration (IC)

- Offer support, coaching, and encouragement to individual followers
- Keep lines of communication open
- Offer direct recognition for contributions of each follower
- Exhibit genuine care and concern

Idealized Influence (IIA/IIB)

- Be a role model; set the example
- Be a lifelong learner; gain the knowledge
- Create an environment of trust and respect

Steps for Transformational Leadership

Developing Vision

- Starts with creating a vision
- Vision that is exciting and attract potential followers

Selling the Vision

- Constantly sell the vision and build a strong network of high potential

Road Map

- Transformation leaders have a clear vision
- Have a fair idea about the direction to move and want followers to move in the same direction
- Ready to accept failures

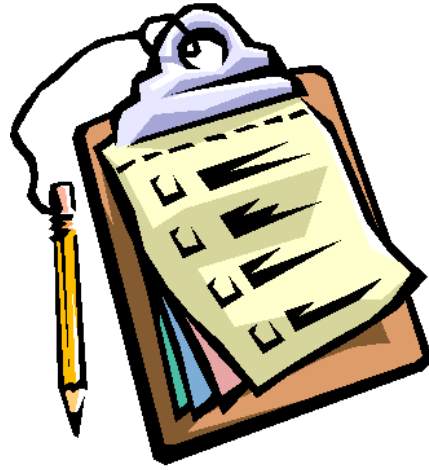
Leading the charge

- Transformation leaders remain in the fore-front during the action
- Always visible and stand up to be counted
- Bolster and infect the followers with commitment and enthusiasm



“Another Week Doin’ Enterprise DevOps”

Agenda



- ▶ Definition of DevOps
- ▶ Delivery Pipeline
- ▶ Automation
- ▶ Metrics
- ▶ Culture
- ▶ Benefits
- ▶ Leadership

▶ Healthsystems DevOps

- ▶ Material

DevOps Initiatives at Healthsystems

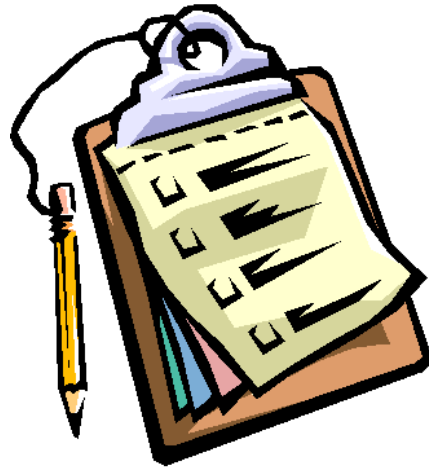
- ▶ Platform as a Service/Infrastructure as a Service/Software as a Service
 - ▶ Empower delivery teams to create their own environments and data; repeatable process
- ▶ Automated Test Coverage
 - ▶ Emphasize the need for this on teams and to build it out
- ▶ Integration Environment
 - ▶ Continuous Integration
 - ▶ Automated Tests for team deployments, team tests and End to End test
- ▶ Quality Gates
 - ▶ Pass Functional tests, Performance tests, Security tests
 - ▶ Unit test coverage, Code quality
- ▶ Blue/Green Environments
 - ▶ Allow us to test production configurations, test production deployments during the day and have deployments be a simple switch over to the new server

DevOps Initiatives at Healthesystems

- ▶ Steering Committee meets weekly
 - ▶ Representatives from:
 - ▶ IT Operations
 - ▶ Application Development
 - ▶ Architecture
 - ▶ Reporting
 - ▶ Database
 - ▶ Release Engineers
 - ▶ Security
 - ▶ Methodology
- ▶ Embedded operation members on agile teams
- ▶ Laid out a roadmap for the initiatives



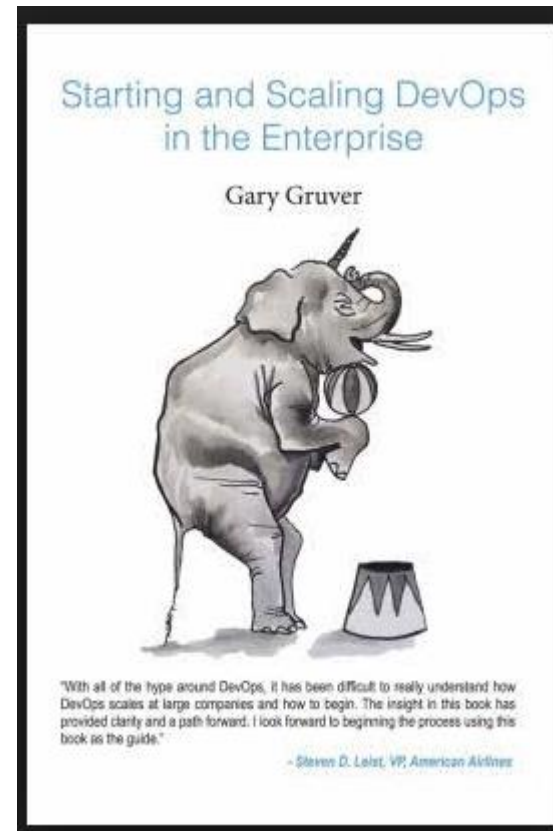
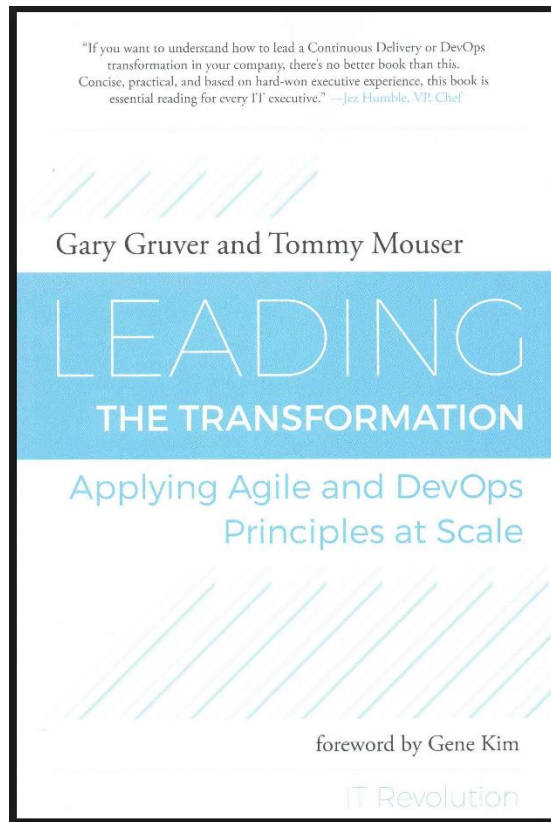
Agenda



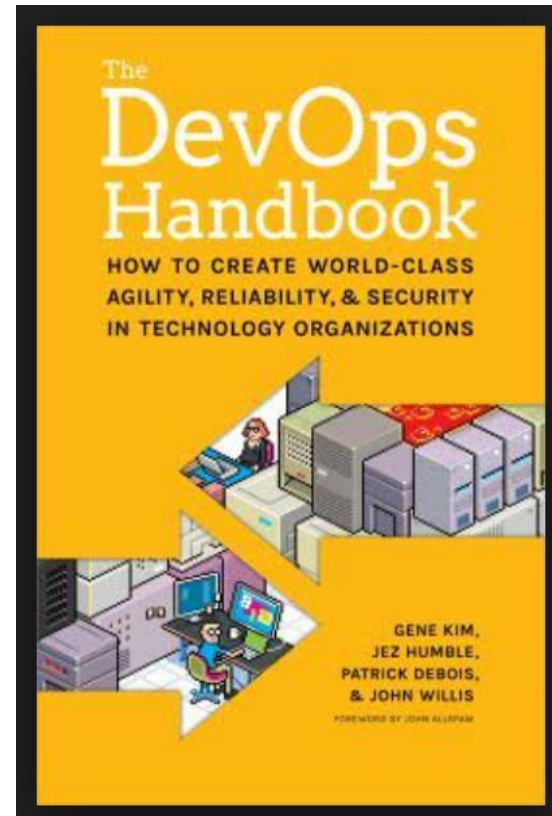
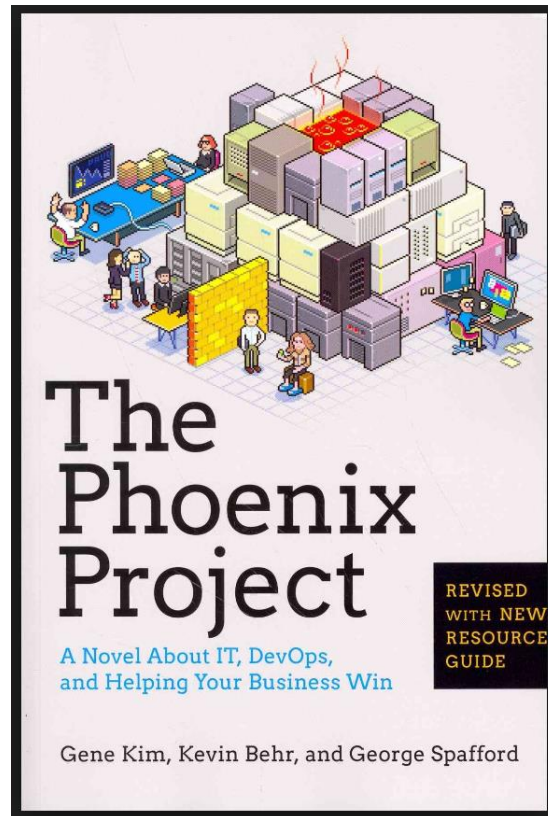
- ▶ Definition of DevOps
- ▶ Delivery Pipeline
- ▶ Automation
- ▶ Metrics
- ▶ Culture
- ▶ Benefits
- ▶ Leadership
- ▶ Healthsystems DevOps
- ▶ **Material**

Leading the Transformation

Starting and Scaling DevOps in the Enterprise



Gene Kim books



Conferences



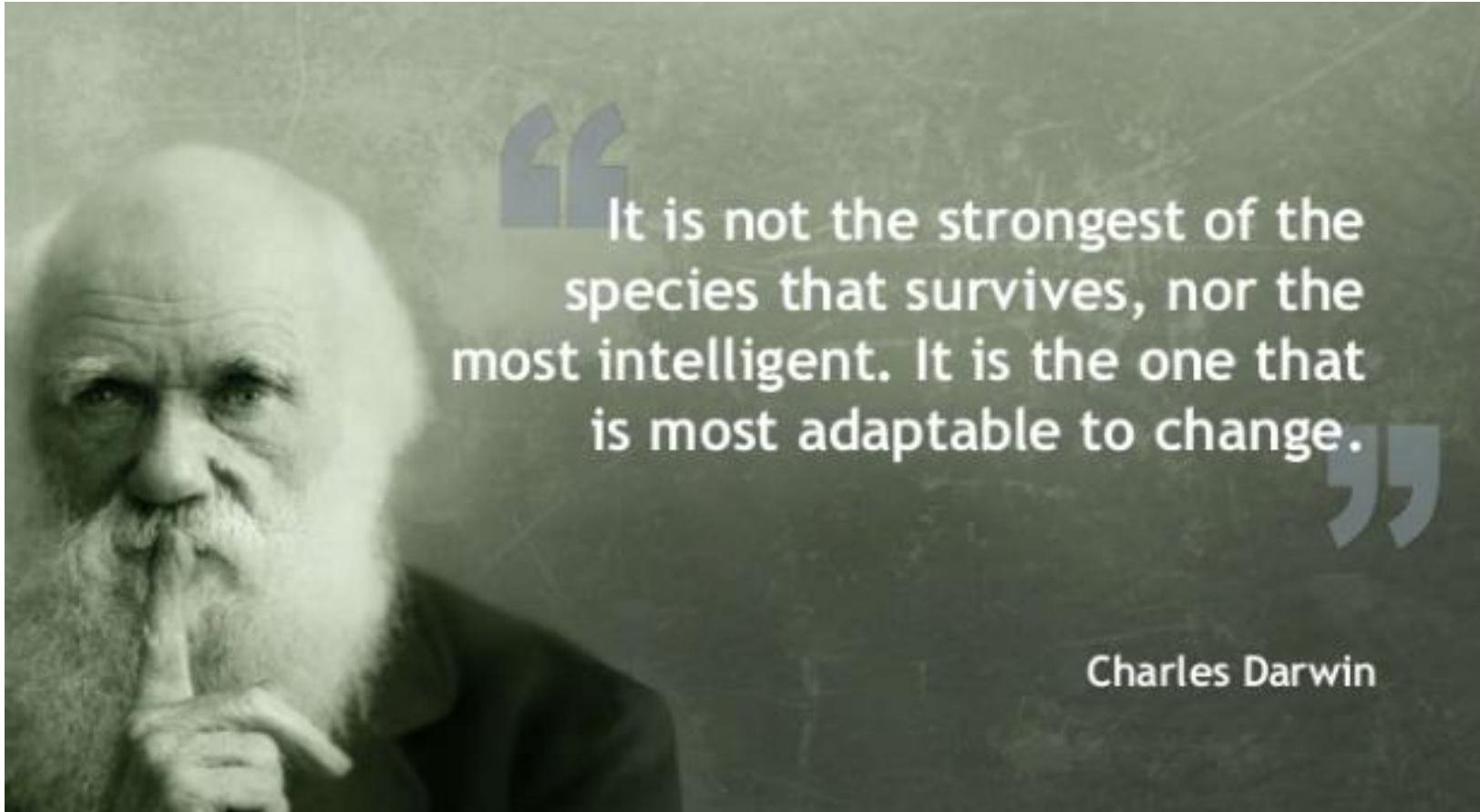
Takeaways from conferences

- ▶ DevOps is no longer a fringe movement, but a competitive necessity
- ▶ Shift-Left to identify quality, performance, security issues sooner
- ▶ Optimize your culture, organization and processes for speed
- ▶ Journey
 - ▶ Building and organizing the right teams
 - ▶ Redesign infrastructure and code
 - ▶ New skills, new tools
 - ▶ Buy In

Goal Today

- ▶ Understanding of what DevOps is
- ▶ How automation and testing fit in
- ▶ What to start measuring
- ▶ Culture and Leadership needed
- ▶ Get excited about the benefits this change can provide





“ It is not the strongest of the species that survives, nor the most intelligent. It is the one that is most adaptable to change. ”

Charles Darwin



Questions



John O'Bryant

Director of Application Development at HealthSystems



<https://www.linkedin.com/in/johnobryant>



johnob12@hotmail.com