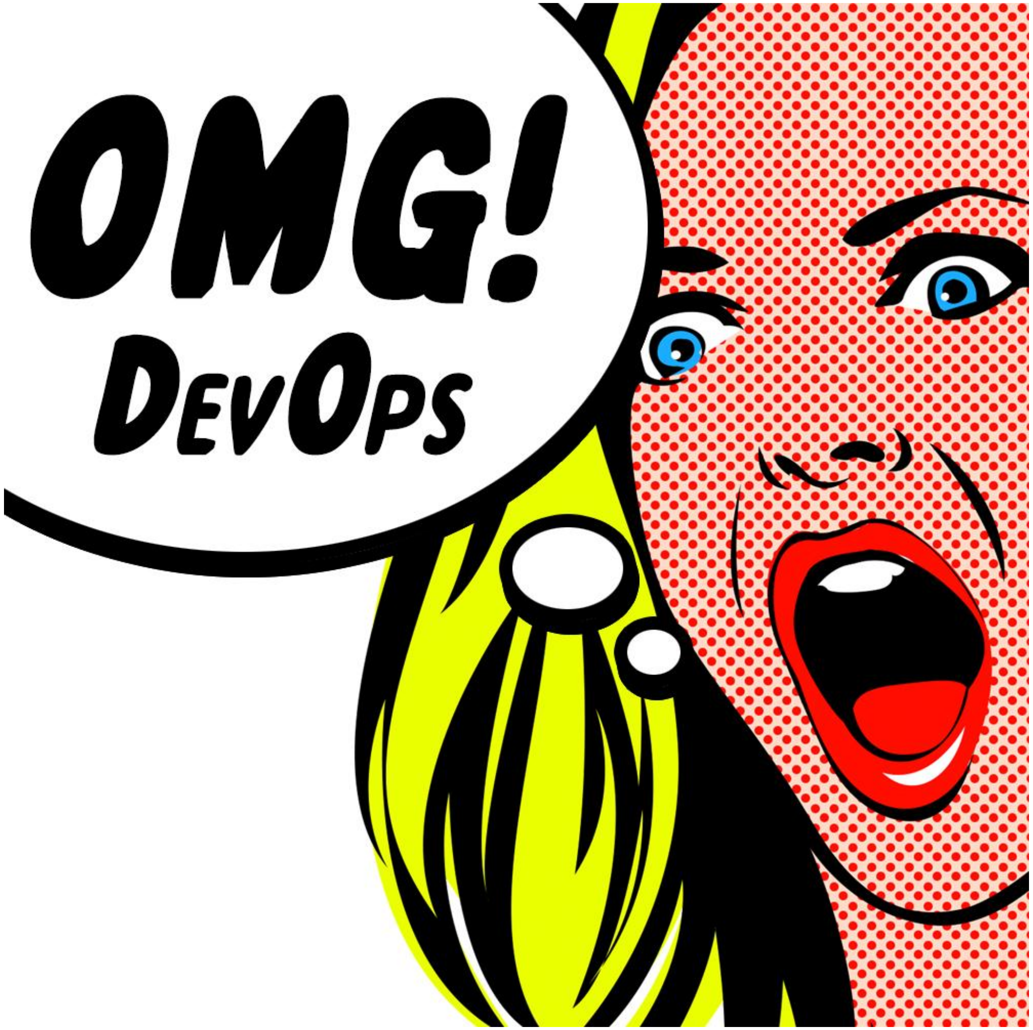
Transforming to DevOps
Creating a Collaborative Project Culture

Experis™
ManpowerGroup

Topics to discuss

- Dispelling DevOps Myths
- So, What is DevOps and Why is it Important?
- Moving to DevOps (Creating a Collaborative Culture)
  - How do you create a collaborative Business, IT and Operations culture?
  - Who are the stakeholders and their Responsibilities?
  - What are the Risks and Exposures?
- DevOps Best Practices to Mitigate Risk and Maximize Value
  - Continuous Agile
  - Community of Practice
  - Enable Automation
  - DevOps Roadmap

Dispelling DevOps Myths



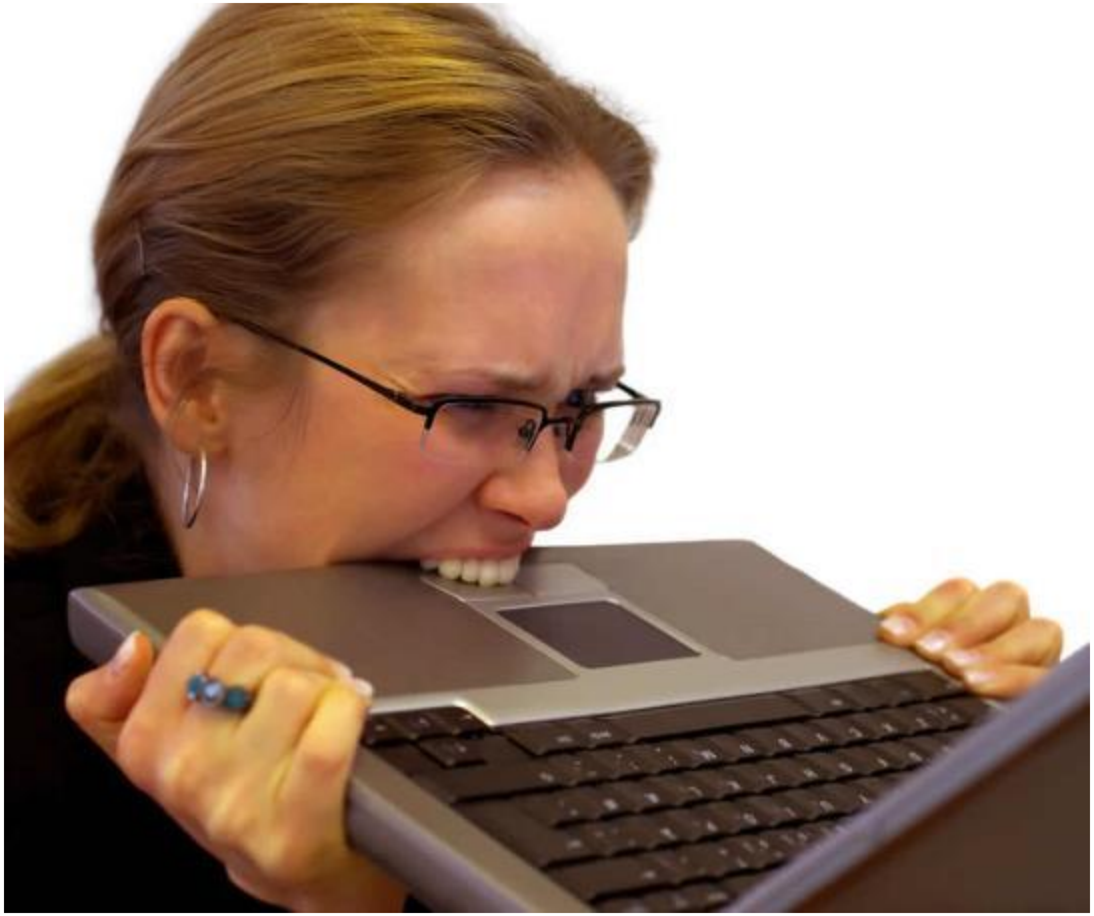
Experis™
ManpowerGroup

You Need a DevOps Guy…

DevOps is a methodology, not a skill. It is something you practice as a team not an individual.

Experis™
ManpowerGroup

DevOps = DIY…



DevOps is achieved through maximizing what you do well and improving upon your deficiencies via assistance from others.

Experis™
ManpowerGroup

DevOps supports any SDLC…

DevOps with legacy Waterfall or DevOps without Continuous Quality has a high risk of failure or not meeting customer satisfaction expectations.

DevOps Clashes with Existing Processes…

DevOps smoothly integrates with existing processes like ITIL, Agile etc.
DevOps is actually a way to improve processes like ITIL.

Experis™
ManpowerGroup

BAs “Bottleneck” DevOps Activities…

Business Analysts can ensure that focus is maintained on enterprise business or project priorities.
Teams collaborate to prioritize initiatives and/or activities that matter to the business, IT and Operations organizations.

Experis™
ManpowerGroup

***"It ain't what you know that gets you into trouble.
It's what you know for sure that just ain't so."***

Mark Twain

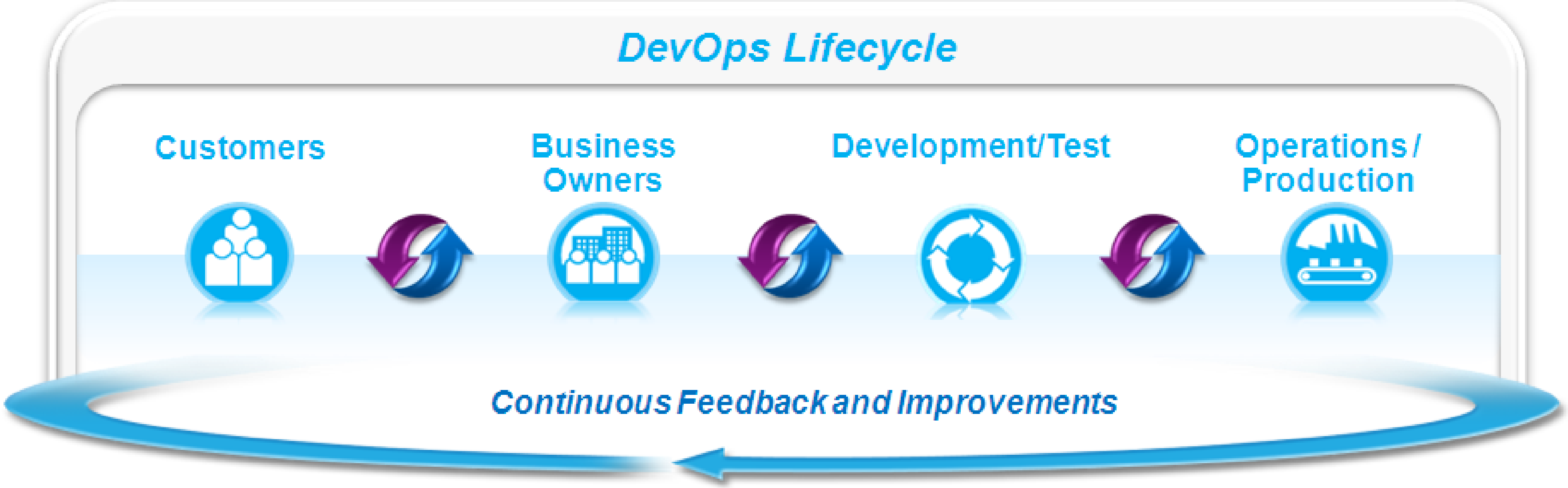
Experis™
ManpowerGroup

Experis™
ManpowerGroup

What is DevOps?

An approach for continuous delivery of software-driven innovation

dev·ops *noun* \'dev-äps\
Enterprise capability for continuous software delivery that enhances ability to seize market opportunities and reduces time-to-customer-feedback



Reduce time to obtain and respond to customer feedback

Accelerate software delivery

Balance speed, cost, quality and risk

...so should it be called

BusDevOps

What is “BusDevOps”?

- BusDevOps is a philosophy where **business teams, development teams**, and the **operations organization** continuously collaborate.
- Some call it **“Near Real Time”** development or **“Elastic”** deployment cycle.
- **Human intervention is minimized wherever possible.**

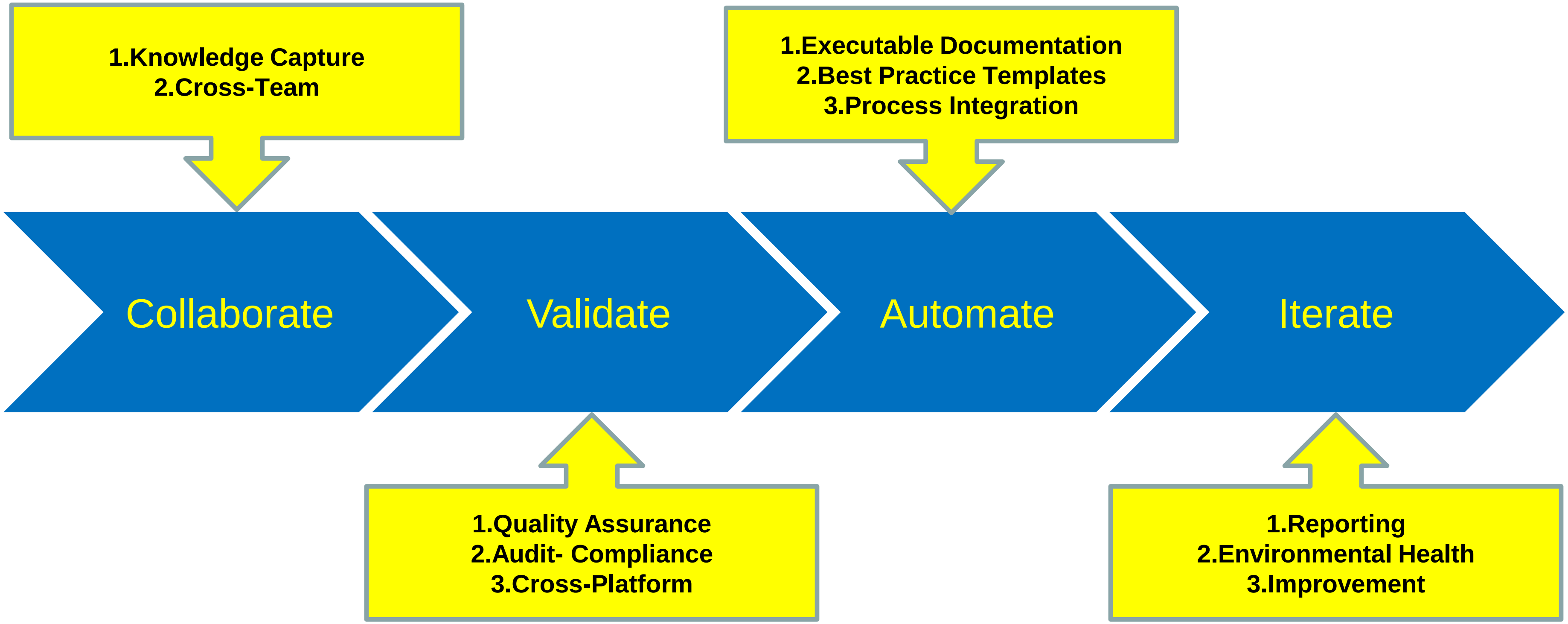
“Big” DevOps and “Little” DevOps

Business Innovation, Workflow & Collaboration

Process Automation & Software Delivery

Experis™
ManpowerGroup

Big DevOps emphasizes…



Collaborate

1.Knowledge Capture
2.Cross-Team

Validate

1.Quality Assurance
2.Audit- Compliance
3.Cross-Platform

Automate

1.Executable Documentation
2.Best Practice Templates
3.Process Integration

Iterate

1.Reporting
2.Environmental Health
3.Improvement

In essence, Big DevOps….

- Brings in cultural change aiming to deliver business system functionality faster at a higher rate of quality.
- Emphasizes operational process
  - Predictability,
  - Efficiency,
  - Security,
  - and Maintainability.
- Normally enables
  - Improved deployment frequency,
  - Lower failure rate of new releases,
  - Shortened lead time between fixes, and
  - Faster mean time to recovery in the event of a new release crashing

Experis™
ManpowerGroup

“Little” DevOps emphasizes…

Visual Studio
CHEF
puppet labs
RALLY SOFTWARE
JIRA
docker
splunk>
DEV OPS
SALTSTACK
Jenkins
ANSIBLE
now
VERSIONONE
Simplifying Software Delivery

In essence, Little DevOps…

- **Accelerates deployment**
  - IT organizations deploy code 30 times more frequently than their peers.
- **Prevents failures and streamlines recovery**
  - Production deployment involves 60 times fewer failures in the first place and recover 168 times faster when an incident does occur
- **Delivers sustainable value**
  - Production deployment involves 20 times higher change success rates and four times lower mean time to repair.
- **Can prevent IT resource “burnout”**
  - Spending less time on “fires” propagates lower incidences of employee burnout.
- **Enable teams to spend less time on administrative support**
  - Via automated tool support managing workflows, hand-offs, builds and deployments

TechBeacon – August 2015

Moving to DevOps

(and creating a collaborative culture)

Why are companies moving to DevOps?

Business Dynamics

- Competitive Pressures
- Early and Often Delivery Benefits of Agile
- Expedited Release Cadence
- Release Consistence and Predictability
- Transparency for audit and regulatory compliance

Technology Dynamics

- Juggling many platforms, teams, tools, and infrastructure to support Software Releases
- Use technology to manage using technology.

Experis™
ManpowerGroup

Why is This Important?

IT-enabled innovation is a competitive differentiator

DevOps can break down barriers and conflicting priorities that often exist between business, development and operations teams.

Google

… so, how effective have companies been at establishing a DevOps culture?

What companies want to achieve

1. Communication of Knowledge and Integration of People
2. Better Process
3. Reality-based Measurements

What companies encounter instead

1. Distracted by day-to-day delivery pressures – 78%
2. Tools don't integrate properly – 62%
3. Lack the necessary internal expertise – 56%

Source: Forrester study commissioned by Wipro, 2008

How do you Create a Collaborative Business, IT and Operations Culture?

Experis™
ManpowerGroup

Who are the Stakeholders?

DevOps is a change to both culture and mindsets

- **Executives**
- **Business Stakeholders**
- **IT & Infrastructure Stakeholders**
- **Business & IT Operational Managers**
- **Business & IT Operational Participants**
- **Coaches/SMEs (Process Analysts)**

Executives Have a Big Role

- **Focus needed on:**
  - Prioritizing project backlog
  - Limiting concurrent assignments (WIP) to project teams
  - Ensuring consistent and ongoing organization support

Experis™ ManpowerGroup

Work within a DevOps Roadmap

1. **Get Executive Buy-In**
   - Common business objective(s) understanding
   - Senior leadership collaborates with middle management
2. **Extend "Agile Principals" to Delivering Systems & Services**
   - Break down large and/or complex deliverables into modular components
3. **Identify & Remove Waste**
   - Map SDLC processes & handoffs
   - Identify points that are cumbersome or manually intensive
   - Apply automation for workflow & time savings
   - Create new RACIs
4. **Socialize, Evangelize & Train**
   - Communicate IT & business improvements
   - Train to "get everyone on the same page"
5. **Be Able to Measure Success**
   - Change Fail Rates
   - Mean Time to Recover (MTTR)
   - Lead Time for Changes
   - Number and frequency of software releases
   - Volume of defects
   - Time/cost per release
   - Number and frequency of outages / performance issues
   - Revenue/profit impact of outages / performance issues
   - Number and cost of resources

Experis™
ManpowerGroup

DevOps “Ins and Outs”

Out: Organizing employees based on department
In: Integrating everyone into a single team

“Standard project management philosophies don’t always apply to DevOps,” That means dropping the usual hierarchical approach to management and carrot-and-stick standbys when it comes to deadlines. Step one is to bring everyone under one roof.

“The DevOps folks should not be treated as a separate organization with their own deadlines and responsibilities. It is absolutely critical that DevOps folks be embedded with your development teams or at least be working very closely with them in a collaborative fashion.”

Craig Schneider

Experis™
ManpowerGroup

DevOps “Ins and Outs”

Out: Deferring quality assurance to the end of the project
In: Building in time to test as you go, at every step

Gone are the formal QA-driven days of develop, then test and then revise. Successful DevOps deployments require organizations to embrace test-driven development and continuous integration approaches to keep projects running swiftly.

Sandeep Sood

DevOps “Ins and Outs”

Out: Telling
In: Showing

Talking to the team about progress on a project is ineffective in today’s complex, visually driven environment. Require all team members to demonstrate their progress, rather than giving verbal updates.

“One of the most common mistakes in the field is being too afraid or lazy to demand that team members are fully transparent about their progress. Ask for an actual demo every week, at a set time and day. This can feel uncomfortable at first, but it gradually becomes habitual. Because progress is fully transparent, people get help early, and it’s completely clear when a piece of a project is at risk.”

Sandeep Sood

DevOps “Ins and Outs”

Out: Prioritizing requirements based on importance
In: Treating nonfunctional requirements as equals

Many developers knows that as a project approaches launch, "wish-list" requirements that don’t affect the functionality of the product often drop off the to-do list. This is how products go out the door with poor documentation, an unfinished user interface, or temporary graphical elements.

“The implementation phase is usually the longest and most tedious. Short deadlines, team friction, and management pressure lead to a strain on resources, and this is where most nonfunctional requirements are dropped on the floor. Adding them ‘later’ is a recipe for disaster.”

Ivo Vachkov

DevOps “Ins and Outs”

Out: Hoping for the best
In: Expecting the worst

Finally, it's key to remember that the first iteration of any new product isn’t going to work properly. Planning for testing is a good start, but planning for recovering from a disaster is also critical. The team should build potential failure directly into the project timeline.

“Design for failure. Always assume it will happen, and design the recovery process.”

Ivo Vachkov

DevOps Risks & Exposures

Experis™
ManpowerGroup

5 Easiest Ways DevOps Fails

Failure #1: Fail to define what DevOps means to your organization

Before you start introducing technology and process under a DevOps initiative make sure to define a baseline for your DevOps initiative:

1. What is your definition of DevOps?
2. What are you trying to accomplish? And, most importantly,
3. Where do you draw the boundaries between DevOps and more structured approaches to IT service management.

5 Easiest Ways DevOps Fails

Failure #2: Focus on tools and techniques, forget about people.

You can’t just hire a bunch of “DevOps People” put them in a room and step away expecting them to work “Magic” and make everything more efficient.
Instead what you need to ensure in any DevOps initiative is that you are taking human-driven processes in account.

Many companies that adopt DevOps, move to faster, more frequent releases driven by the needs of individual projects only to realize that an increase cadence of software delivery can lead to QA and release management burnout. If you are introducing more automation to speed time to market make sure you also think of the impact any DevOps initiative is going to have on people.

5 Easiest Ways DevOps Fails

Failure #3: Ignore governance entirely.

DevOps in the enterprise tends to emerge from one or two group deciding to stage a revolution against an ineffective IT organization. An enterprise without common standards for software architecture, release management, and environment management isn’t an enterprise at all – it’s an awful mess.

The largest companies in the industry want to enable teams to work faster, but also understand that DevOps isn’t about reducing the number of governance gates. On the contrary, if anything DevOps enables more effective, more frequent governance gates.

5 Easiest Ways DevOps Fails

Failure #4: Fail to account for risk.

More frequent releases, self-service provisioning of infrastructure, infrastructure automation, continuous delivery pipelines: all of these common factors of DevOps initiatives lead to faster time to market, but at the tail end of a release process the business risks remain unchanged.

Changes to production facing systems still require rigorous change management and when multiple teams feel empowered to push to production every week (or every day) you still need some release management function tracking conflicts and risk.

When organizations adopt DevOps they often lose the built-in “checks and balances” that came with ITIL. Software can be delivered faster, but the enterprise still require governance gates.

5 Easiest Ways DevOps Fails

Failure #5: Run DevOps without metrics.

Enterprises expect to see hard data to back up staffing and infrastructure decisions. If you are invested in the success of a DevOps initiative make sure that you are collecting statistics that justify your investment.

Keep track of release and environment metrics that are involved with DevOps and teams not involved with DevOps and use the data to make informed decisions to dial up or dial down particular initiatives from your DevOps teams.

Common exposures preventing DevOps?

- Critical Business Objective with no plan or an ability/available cycles to develop one
- Agile enablement (or a tailored version) for customers predominantly embracing Waterfall SDLCs
- Current state of Agile does not embrace a “continuous” culture
- Collaboration is critical and needs to be “delicately” managed
- Automation selection and configuration

DevOps Best Practices to Mitigate Risk & Maximize Value

Experis™
ManpowerGroup

Leverage or Consider Continuous Agile

- **Continuous Development** via Lean - Kanban function boards or Test-Driven Development (TDD)
- **Continuous Integration** via builds after each change or frequent builds on a daily basis
- **Continuous Quality** through automated unit, regression and integration testing
- **Continuous Improvement** through sprint retrospectives and frequent process improvement and/or tailoring
- **Continuous Delivery** with releases to production after each functional change
- **Continuous Project Management** with Product Owners contributing and collaborating with Agile development teams on a daily basis towards
  - What is the right thing to build,
  - What does it look like and
  - How does it work

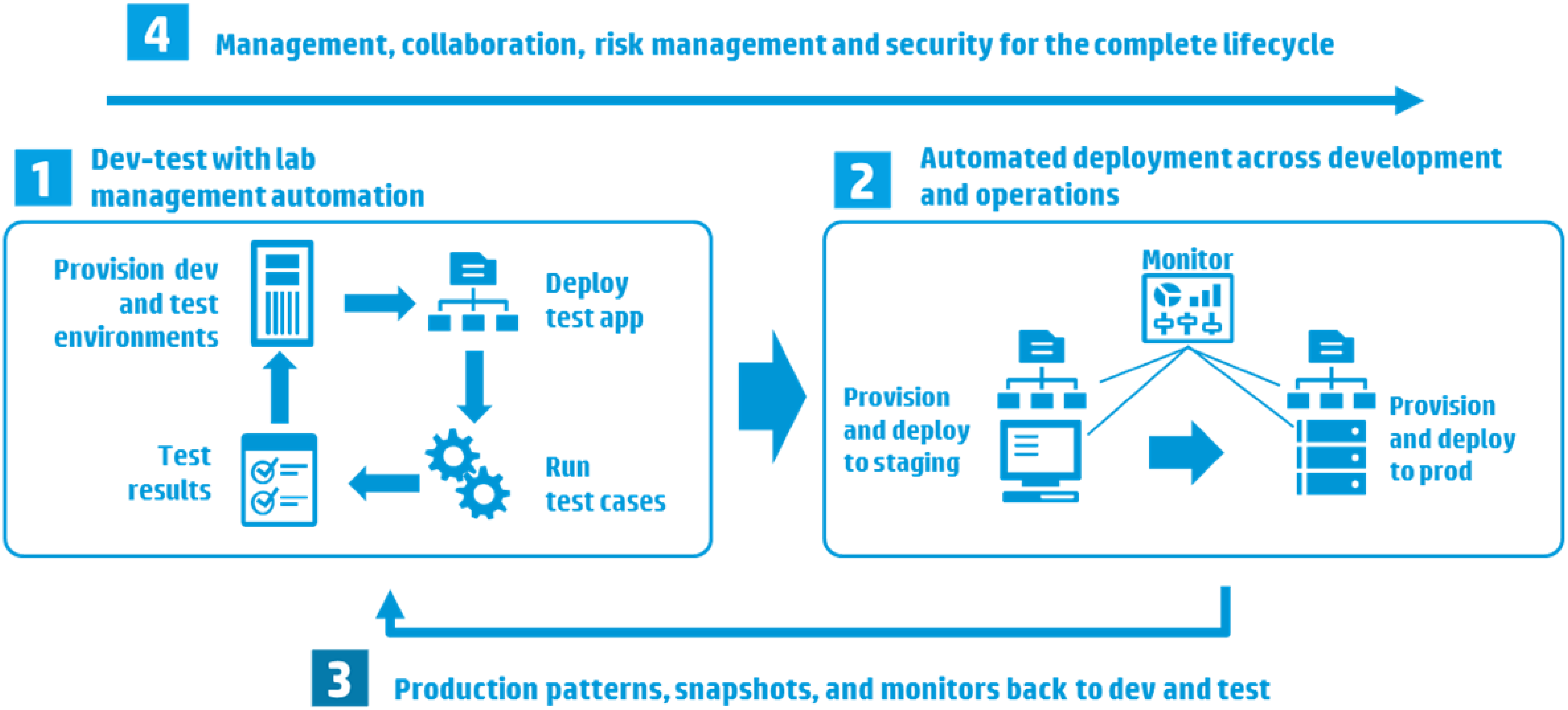
PMO Community of Practice

- The PMO Community of Practice (CoP) helps bridge a critical organizational gap between delivery and strategy. This focus enables the translation of organizational strategies into actionable plans. The CoP facilitates collaboration, knowledge sharing, and professional development opportunities for a growing community of project, program, and portfolio professionals.
- **Benefits**
  - Promotes knowledge sharing
  - Addresses problems impacting the community
  - Addresses not having a centralized PMO for an initiative crossing several organizational boundaries.

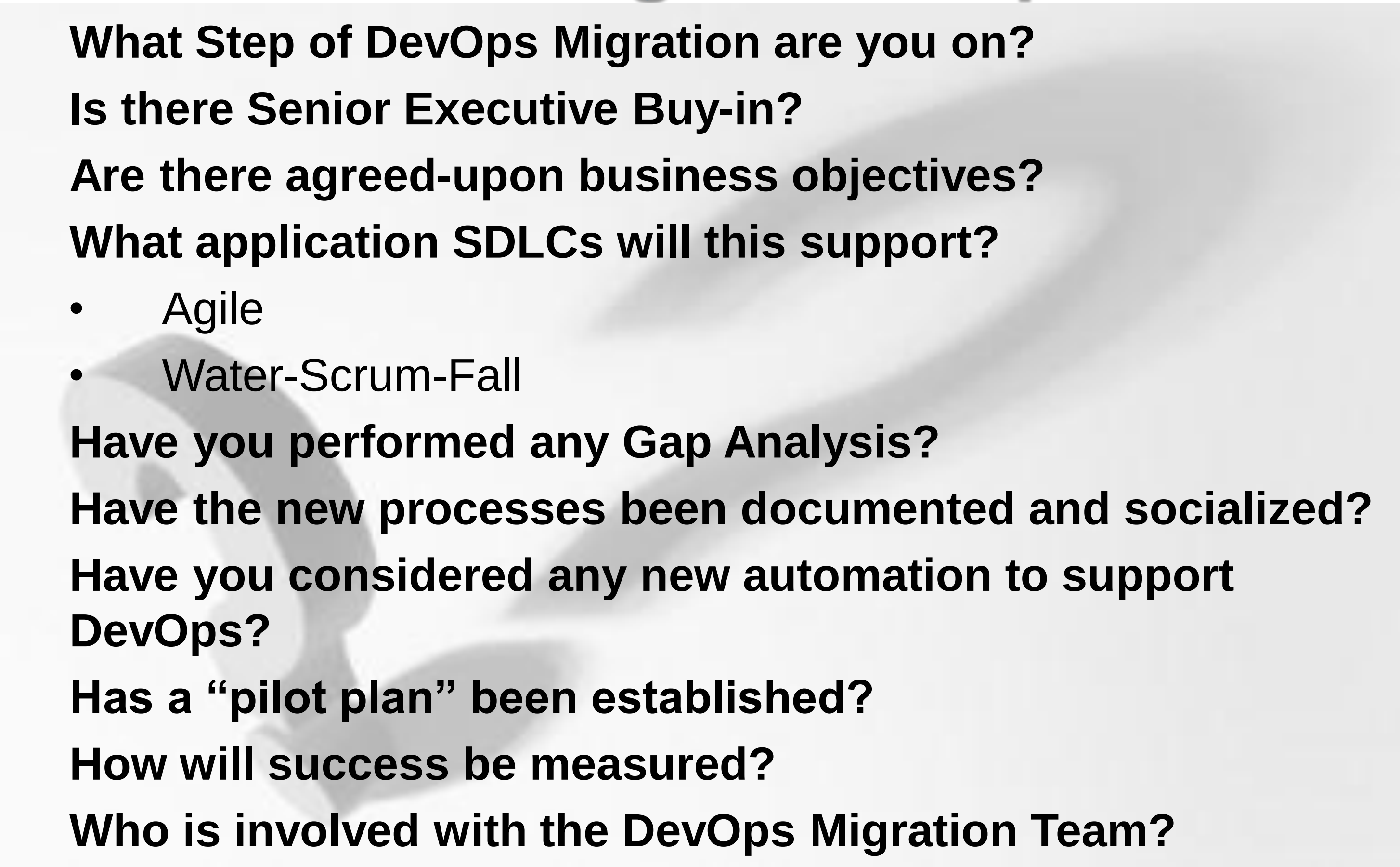
Thoughts on Tools…

TINY-GIANT-BOOKS.COM
THEY WANTED MORE AGILITY
SO WE SOLD THEM $2 MILLION IN TOOLS

Thoughts on Tools…


4
Management, collaboration, risk management and security for the complete lifecycle
1
Dev-test with lab management automation
Provision dev and test environments
Deploy test app
Run test cases
Test results
2
Automated deployment across development and operations
Monitor
Provision and deploy to staging
Provision and deploy to prod
3
Production patterns, snapshots, and monitors back to dev and test

Understand where you are with respect to moving to DevOps



- **What Step of DevOps Migration are you on?**
- **Is there Senior Executive Buy-in?**
- **Are there agreed-upon business objectives?**
- **What application SDLCs will this support?**
  - Agile
  - Water-Scrum-Fall
- **Have you performed any Gap Analysis?**
- **Have the new processes been documented and socialized?**
- **Have you considered any new automation to support DevOps?**
- **Has a “pilot plan” been established?**
- **How will success be measured?**
- **Who is involved with the DevOps Migration Team?**

Experis™
ManpowerGroup

DevOps & Business Analysis

- **DevOps is NOT Agile, but is an extension of Agile thinking or an Agile culture**
  - Agile enables the provision of Production-Ready software
  - DevOs deploys into Production
  - DevOps involves functional, data and process/workflow analysis

- **Imbed Operations at the same time as the Business**
  - DevOps is About Culture and Quality
  - Early involvement of experts Ops = experts in maintainability and deployability
  - Break down organizational silos
  - Enable and require constant communication

- **Role Diversification**
  - Test Drive Everything
  - Leverage automated tests for everything: application, infrastructure, everything.
  - Enable and require constant communication

…so what are the outcomes with a collaborative DevOps culture?

- DevOps oriented teams spend slightly more time automating tasks
- Both Traditional IT and DevOps oriented teams communicate actively
- DevOps oriented teams fight fires less frequently
- DevOps oriented teams spend less time on administrative support
- DevOps oriented teams work fewer days after-hours

Contact Information

- Speaker: Dennis L. Baldwin
- Company: Experis
- Website: www.experis.com
- Phone: (214) 202-9363
- E-mail: dennis.baldwin@experis.com
- Linkedin Address: https://www.linkedin.com/pub/dennis-baldwin/0/806/66a

Thank You