



Welcome to DevOps Victoria

State of DevOps 2018

[All links are up on GitHub DevOps Victoria](#)



Thank you Red Hat!



redhat.

BUSINESS PARTNER



Agenda



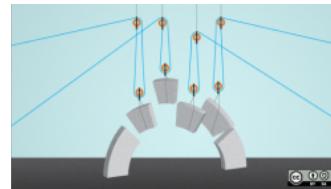
- Welcome & Introduction
- State of DevOps 2018 Overview
- From the Frontlines: **Your State of DevOps for 2018 interactive session**

This presentation has been modified (removed animations for PDF thus reference only)



Upcoming Events

- Willy-Peter Schaub – CSI Lab Jan 22 Mindset Essentials
- January / February Meetup – DevOps Manifesto ([OpenSource.com](https://www.opensource.com))
- KubeCon & CloudNativeCon - Dec 10 – 13 (sold out)



- Training – DA and DevOps – Jan, Feb, March, April
- “Cat out of the bag” – Poking the hornets nest
 - Survey (we need your help! See link below)
 - <https://willys-cave.ghost.io/poking-the-devops-hornets-nest/>

Sponsors



theDock
CENTRE FOR SOCIAL IMPACT
thedockvictoria.com



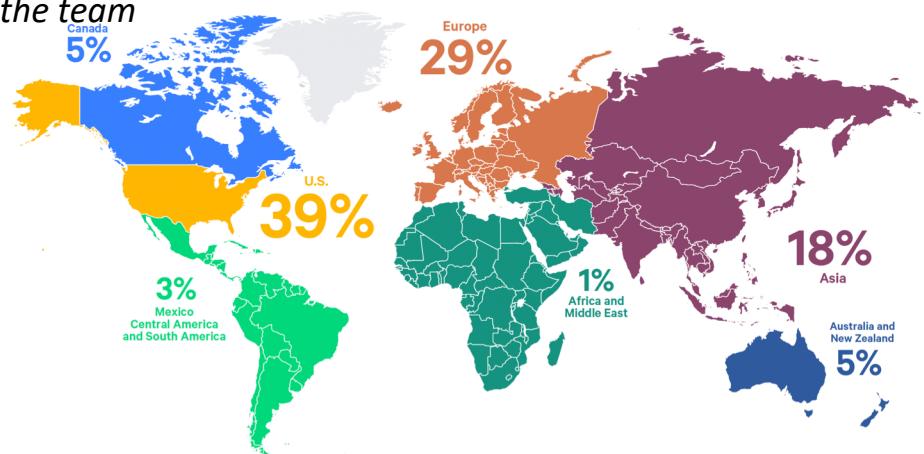
State of DevOps 2018



In a DevOps evolution, there are many paths to success, but even more that lead to failure.

Executives have a rosier view of their DevOps progress than the teams they manage.

64 percent of C-suite respondents believe security teams are involved in technology design and deployment versus 39 percent at the team



Start with the practices that are closest to production; then address processes that happen earlier in the software delivery cycle.

We are often asked “Where do we start?” We recommend starting where the pain is most acute and visible, which is typically application deployments — the boundary between Dev and Ops. Let’s face it: You’re not going to magically fix your organization’s culture overnight. But you can start by improving collaboration (and results) across this one critical functional boundary.

Cross-team sharing is key to scaling DevOps success.

We discovered that the foundational practices — the practices with the most significant impact across the entire DevOps evolutionary journey — are dependent on sharing, one of the key pillars of DevOps. Organizations that have small pockets of DevOps success, yet never manage to spread that success further, are stalled and can’t progress to higher levels of automation and self-service. So the business impact of their DevOps success may not be felt where it matters.

To ensure you can scale your early success, prioritize the building blocks that can be reused and consumed across teams, such as deployment patterns. Promoting reuse of successful patterns, enabling teams to contribute improvements to other teams’ tooling, and sharing both successes and failures are all critical to expanding the other three pillars of DevOps: culture, automation and measurement.

Automating security policy configurations is mission-critical to reaching the highest levels of DevOps evolution.

Highly-evolved organizations are 24 times more likely to always automate security policy configurations compared to the least evolved organizations. As organizations evolve, security policy becomes part of operations, not just an afterthought when an audit looms. This requires first breaking down boundaries between ops and security teams (which are further from production). As we see with all the fundamental practices of DevOps, this practice evolves from resolving immediate pain to a more strategic focus — in this case, from “keep the auditors off my back” to “keep the business and our customer data secure.” In other words, teams automate security policy configurations initially for their own benefit, and as their understanding evolves, the automation evolves to benefit the entire organization.

Talk, walk, sit, work together, its no longer pair programming its pair people ...

State of DevOps 2018

SDO performance unlocks competitive advantages.

Those include increased profitability, productivity, market share, customer satisfaction, and the ability to achieve organization and mission goals.

How you implement cloud infrastructure matters.

The cloud improves software delivery performance and teams that leverage all of cloud computing's essential characteristics are 23 times more likely to be high performers.

Open source software improves performance.

Open source software is 1.75 times more likely to be extensively used by the highest performers, who are also 1.5 times more likely to expand open source usage in the future.

Outsourcing by function is rarely adopted by elite performers and hurts performance.

While outsourcing can save money and provide a flexible labor pool, low-performing teams are almost 4 times as likely to outsource whole functions such as testing or operations than their highest-performing counterparts.

Key technical practices drive high performance.

These include monitoring and observability, continuous testing, database change management, and integrating security earlier in the software development process.

Industry doesn't matter when it comes to achieving high performance for software delivery.

We find high performers in both non-regulated and highly regulated industries alike.



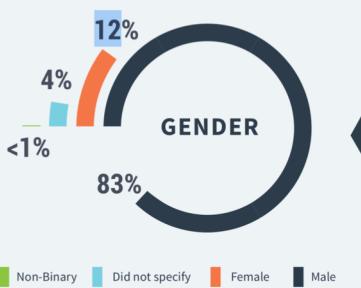
State of DevOps 2018

**30k Professionals
Worldwide > 5 Years**

This year we examine the impact that cloud adoption, use of open source software, organizational practices (including outsourcing), and culture all have on software delivery performance.

Source: Dora - Accelerate: State of DevOps 2018

DEMOGRAPHICS

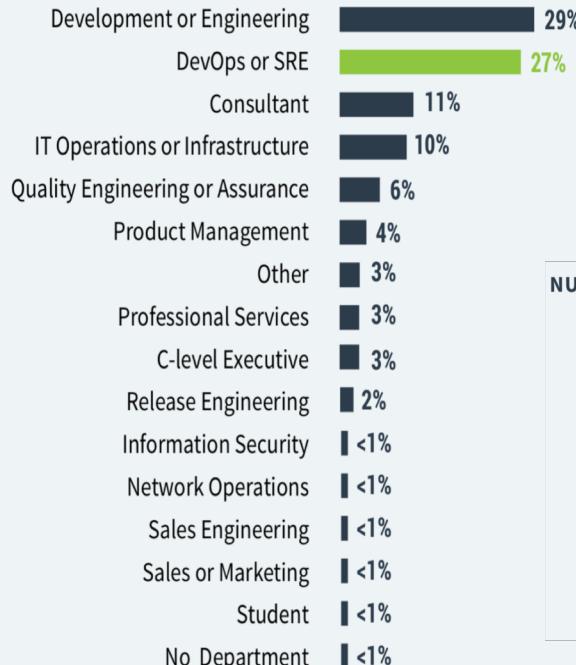


Respondents stated 25% of their teams are women

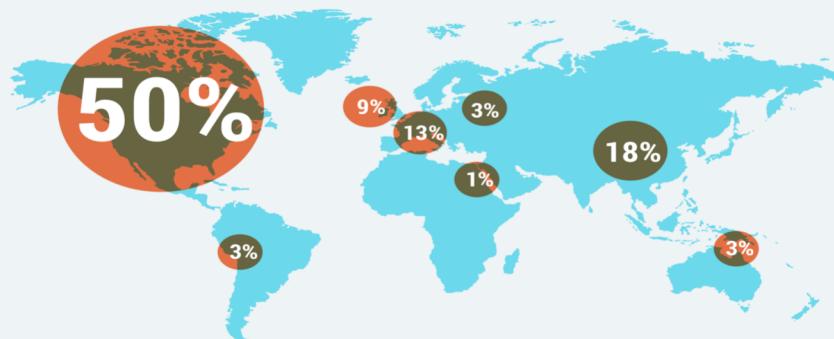
Who took this survey

Participants who work in a DevOps team have increased since we began our study, reporting 16% in 2014, 19% in 2015, and 22% in 2016, and holding steady at 27% in 2017 and 2018.

DEPARTMENT



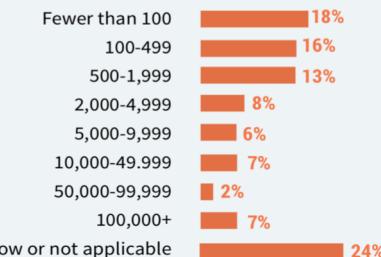
REGION



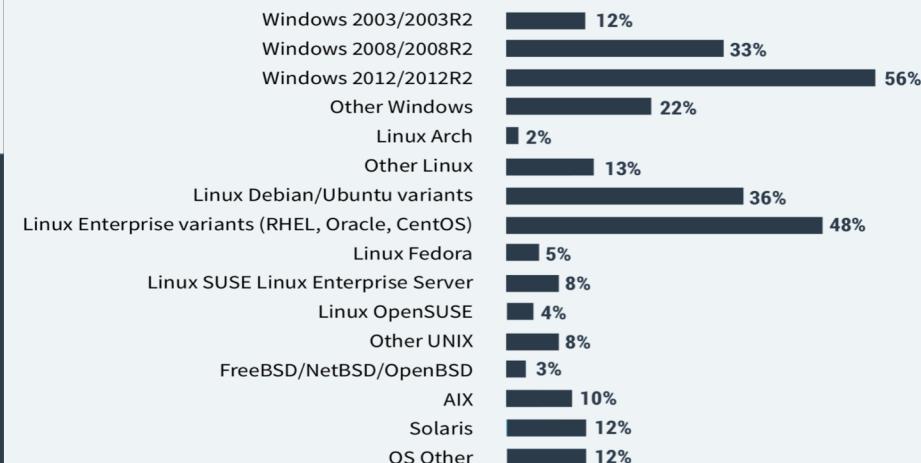
NUMBER OF EMPLOYEES



NUMBER OF SERVERS

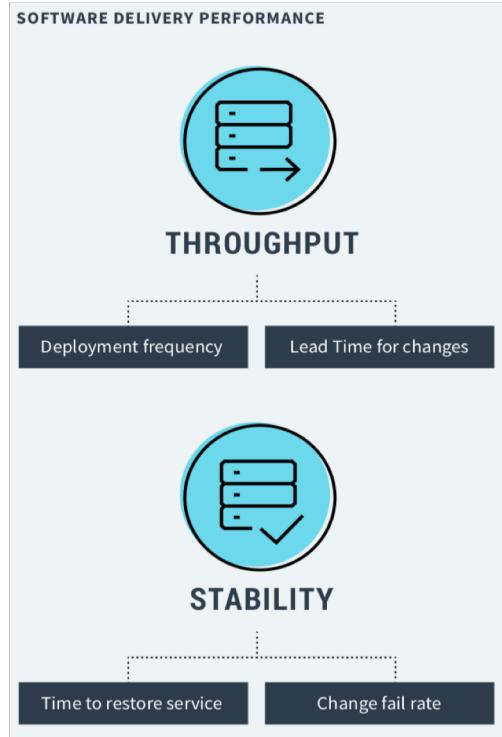


OPERATING SYSTEMS

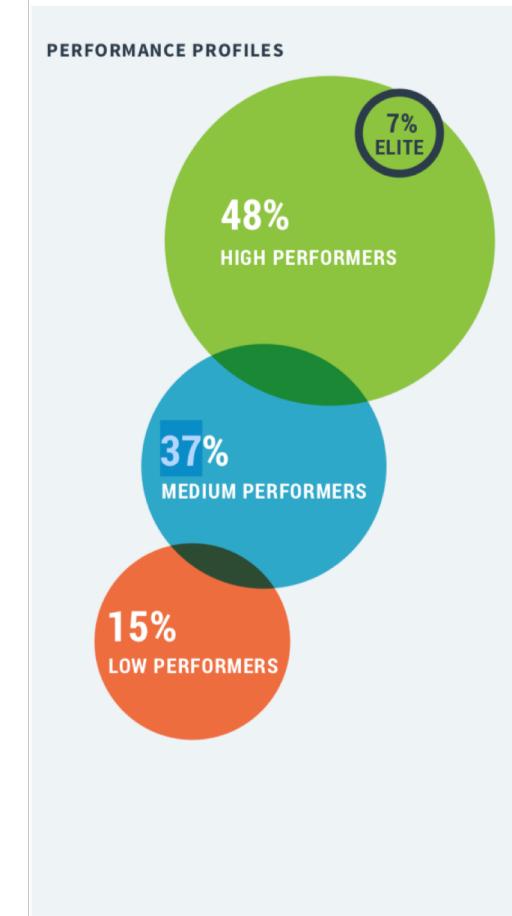


State of DevOps 2018

2018 marks industry shift in how software & technology are viewed: *Software Delivery Performance*



- Capture & Measure focus on outcomes
- No Trade-Offs (i.e. Government or highly regulated fields) backed by research
 - Throughput vs. Stability
 - Highest Performers excelling at all aspects of Throughput & Stability
 - Strong Correlation with SDO performance and Availability
 - Elite performers being 3.55x more likely (strong availability)



Continuous improvement creates better performers...

State of DevOps 2018

...the gap is widening for low performers

2018 marks industry shift in how software & technology are viewed: *Software Delivery Performance*

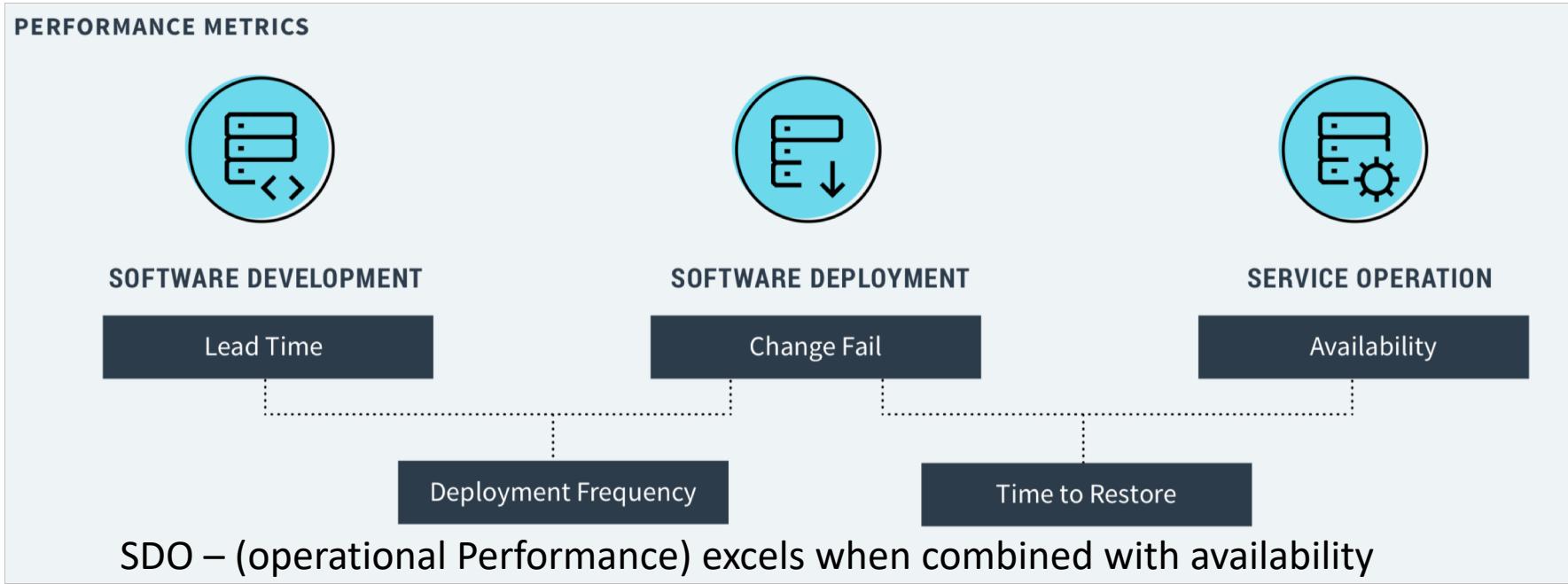
Aspect of Software Delivery Performance	Elite ^a	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code?	On-demand (multiple deploys per day)	Between once per hour and once per day	Between once per week and once per month	Between once per week and once per month
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code commit to code successfully running in production)?	Less than one hour	Between one day and one week	Between one week and one month ^b	Between one month and six months ^b
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident occurs (e.g., unplanned outage, service impairment)?	Less than one hour	Less than one day	Less than one day	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes results either in degraded service or subsequently requires remediation (e.g., leads to service impairment, service outage, requires a hotfix, rollback, fix forward, patch)?	0-15%	0-15%	0-15%	46-60%
 46 TIMES MORE <small>frequent code deployments</small>	 2,555 TIMES FASTER <small>lead time from commit to deploy</small>	 7 TIMES LOWER <small>change failure rate (changes are 1/7 as likely to fail)</small>	 2,604 TIMES FASTER <small>time to recover from incidents</small>	

MISGUIDED PERFORMERS	
Deployment frequency	Between once per month and once every six months
Lead time for changes	Between one month and six months
Time to restore service	Between one month and six months
Change failure rate	16-30%

Embrace change, failure is inevitable, so innovate or face extinction
– Brent Reed



State of DevOps 2018

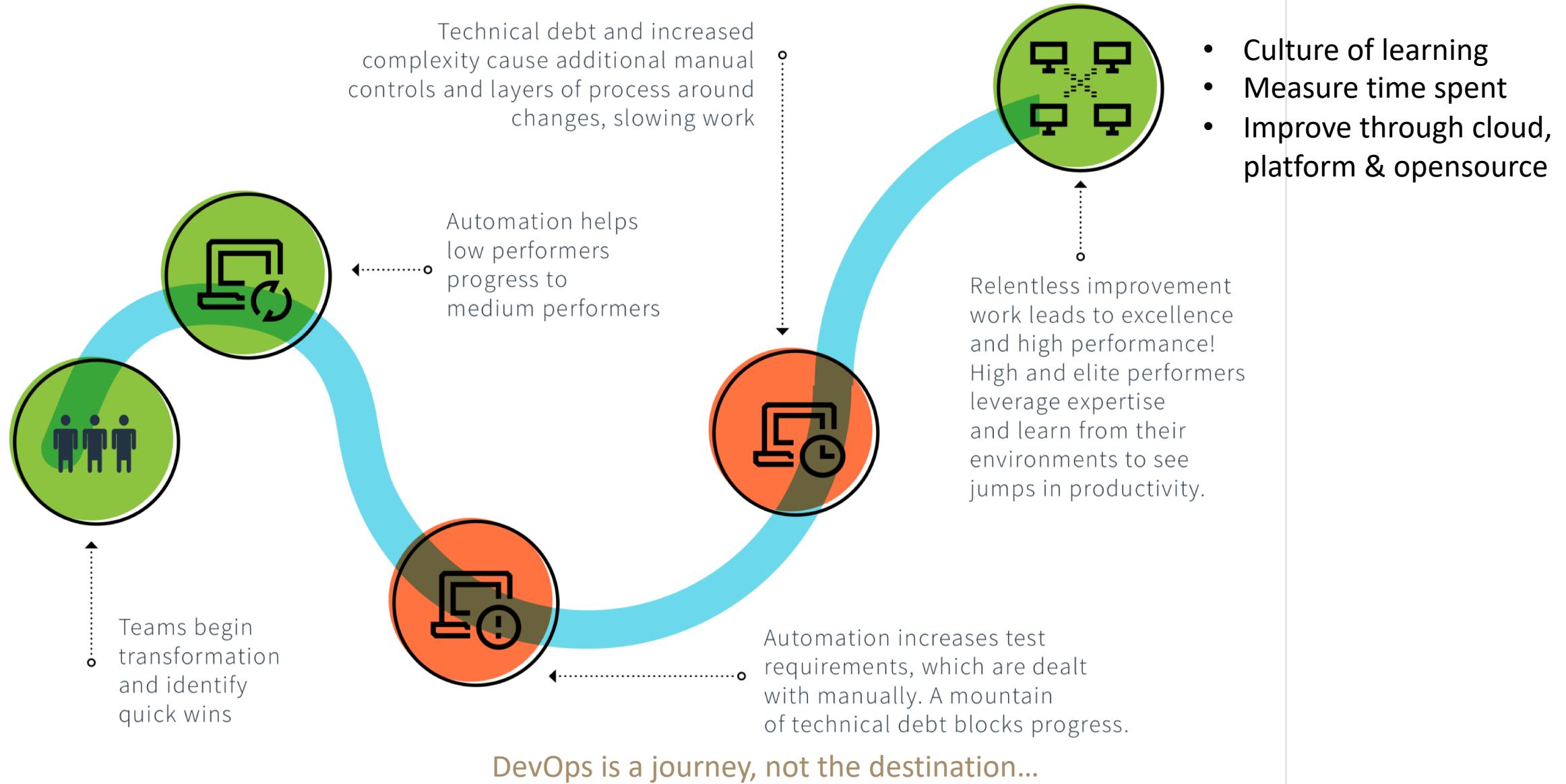


This forms Organizational Performance – the real reason for DevOps

- Profitability
- Productivity
- Market share
- Number of customers
- Quantity of products or services
- Operating efficiency
- Customer satisfaction
- Quality of products or services provided
- Achieving organization or mission goals

Build with Security in mind, InfoSec should be

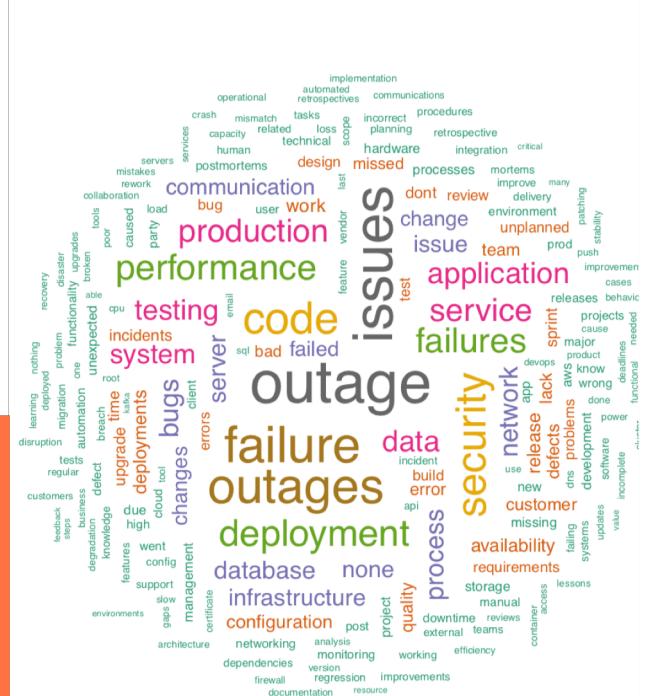
J-CURVE OF TRANSFORMATION



Following slides are copied from animations:

- For reference only these slides depict discussion points regarding the SDO and Service Delivery Organization practices that influence, or can be associated with the constructs.





- 5 Essential Characteristics of Cloud Computing
- Cloud Native
- Infrastructure as Code
- OpenSource
- Outsourcing sux (its an anchor) * more expensive, misguided

LEAN AND AGILE PRACTICES

LEAN PRODUCT MANAGEMENT

Team experimentation

Working in small batches

Gathering & implementing customer feedback

Westrum organizational culture*

SDO PERFORMANCE

Software delivery performance

AVAILABILITY

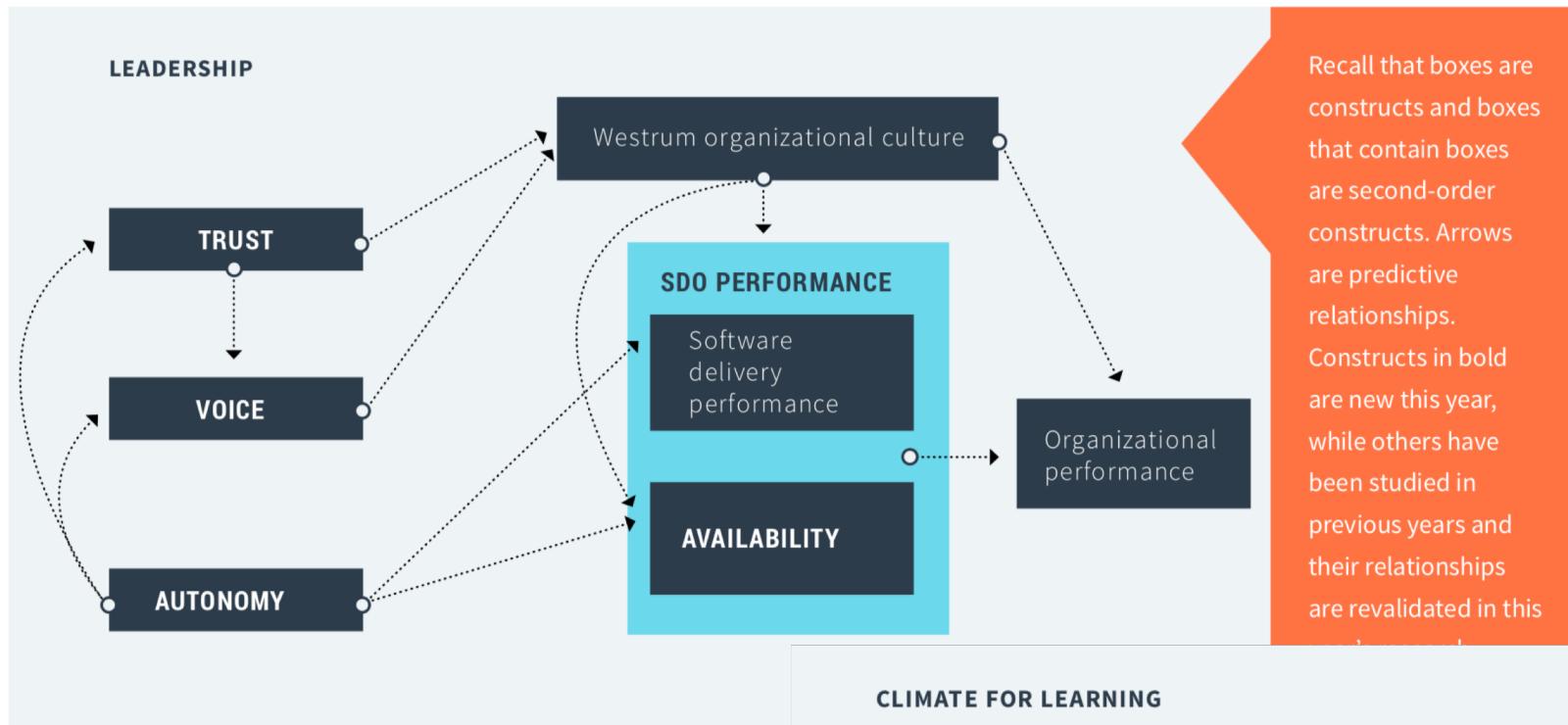
Organizational performance

OUTSOURCING

Recall that boxes are constructs and boxes that contain boxes are second-order constructs. Arrows are predictive relationships. Constructs in bold are new this year, while others have been studied in previous years and their relationships are revalidated in this year's research.

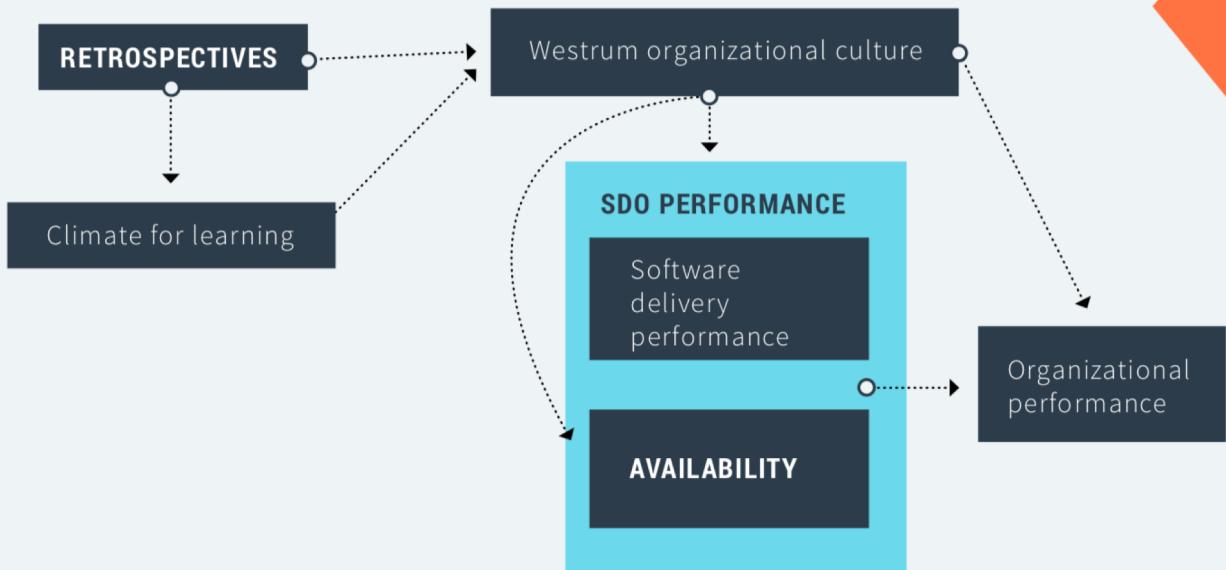
* Westrum is a measure of organizational culture particularly well-suited to DevOps. We discuss this in detail [on page 62](#) in our Culture section.

LEADERSHIP



Recall that boxes are constructs and boxes that contain boxes are second-order constructs. Arrows are predictive relationships. Constructs in bold are new this year, while others have been studied in previous years and their relationships are revalidated in this

CLIMATE FOR LEARNING



Recall that boxes are constructs and boxes that contain boxes are second-order constructs. Arrows are predictive relationships. Constructs in bold are new this year, while others have been studied in previous years and their relationships are revalidated in this year's research.

TECHNICAL PRACTICES

Deployment automation

Continuous integration

Trunk-based development

Loosely coupled architecture

Version control

CONTINUOUS TESTING

MONITORING & OBSERVABILITY

DATABASE

SECURITY

Continuous delivery

CLOUD INFRASTRUCTURE

Westrum organizational culture

SDO PERFORMANCE

Software delivery performance

AVAILABILITY

Organizational performance

Deploy pain

Burnout

Recall that boxes are constructs and boxes that contain boxes are second-order constructs, while arrows are predictive relationships. Constructs in **bold** are new this year, while others have been studied in previous years and their relationships are revalidated in this year's research.

Time Spent	Elite	High	Medium	Low
NEW WORK	50%	50%	40%	30%
Unplanned work and rework	19.5%	20% ^a	20% ^a	20% ^a
Remediating security issues	5%	5% ^b	5% ^b	10%
Working on defects identified by end users	10%	10% ^c	10% ^c	20%
Customer support work	5%	10%	10%	15%

Medians reported because distributions are not normal.

^a Significantly different when testing for differences using Tukey's post hoc analysis

^{b, c} Not significantly different when testing for differences using Tukey's post hoc analysis



23 TIMES

Teams that adopt essential cloud characteristics are 23 times more likely to be elite performers.

FIVE ESSENTIAL CHARACTERISTICS OF CLOUD COMPUTING

% AGREED OR STRONGLY AGREED

On-demand self-service
Consumers can provision computing resources as needed, automatically, without any human interaction required.

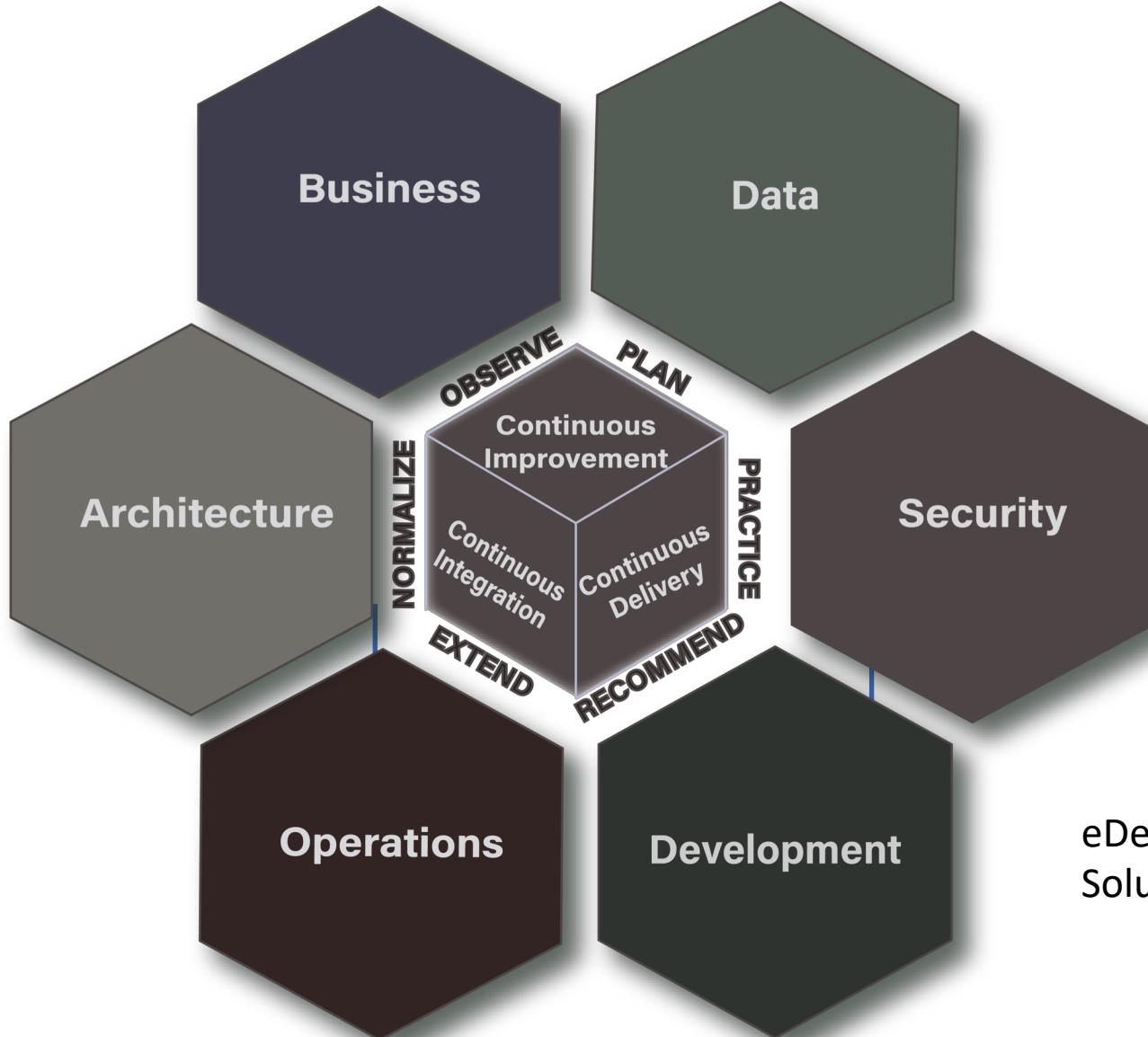
Broad network access
Capabilities are widely available and can be accessed through heterogeneous platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling
Provider resources are pooled in a multi-tenant model, with physical and virtual resources dynamically assigned and reassigned on-demand. The customer generally has no direct control over the exact location of provided resources, but may specify location at a higher level of abstraction (e.g., country, state, or datacenter).

Rapid elasticity
Capabilities can be elastically provisioned and released to rapidly scale outward or inward commensurate with demand. Consumer capabilities available for provisioning appear to be unlimited and can be appropriated in any quantity at any time.

Measured service
Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported for transparency.

eDevOps



eDevOps © Tactec Strategic Solutions Inc.



Your experiences...

