The key to high performing tech organizations

@jezhumble #agileaus
“the enterprise”

Business

Project C

Project B

Engineering

Project A

Operations

DBAs

Infrastructure team

Service desk

Value stream

Ping!
Let's create a new product
We're going agile!

Oh no!

Value stream

Business

Project D

Project A

Project B

Engineering

Operations

DBAs

Infrastructure team

Service desk
Our test-driven code follows SOLID principles.

Shame it doesn't work.
Amazon May Deployment Stats
(production hosts & environments only)

11.6 seconds
Mean time between deployments (weekday)

1,079
Max # of deployments in a single hour

10,000
Mean # of hosts simultaneously receiving a deployment

30,000
Max # of hosts simultaneously receiving a deployment

software delivery as a competitive advantage

“Firms with high-performing IT organizations were *twice as likely* to exceed their *profitability, market share* and *productivity* goals.”

software delivery as a competitive advantage

high performers were more than twice as likely to achieve or exceed the following objectives:

- Quantity of products or services
- Operating efficiency
- Customer satisfaction
- Quality of products or services provided
- Achieving organizational and mission goals
- Measures that demonstrate to external parties whether or not the organization is achieving intended results

software delivery performance

lead time for changes (version control to production)

deploy frequency

time to restore service

change fail rate

# 2018 performance benchmarks

<table>
<thead>
<tr>
<th>Aspect of Software Delivery Performance</th>
<th>Elitea</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment frequency</td>
<td>On-demand (multiple deploys per day)</td>
<td>Between once per hour and once per day</td>
<td>Between once per week and once per month</td>
<td>Between once per week and once per month</td>
</tr>
<tr>
<td>Lead time for changes</td>
<td>Less than one hour</td>
<td>Between one day and one week</td>
<td>Between one week and one monthab</td>
<td>Between one month and six monthsab</td>
</tr>
<tr>
<td>Time to restore service</td>
<td>Less than one hour</td>
<td>Less than one day</td>
<td>Less than one day</td>
<td>Between one week and one month</td>
</tr>
<tr>
<td>Change failure rate</td>
<td>0-15%</td>
<td>0-15%</td>
<td>0-15%</td>
<td>46-60%</td>
</tr>
</tbody>
</table>

Data shows a new 4th high performance group: **elite performers**

Proportion of high performers has grown YoY, but the bar for excellence remains high.

Elite performers are still able to optimize for throughput and stability.

firmographics

Accelerate: The Science of Lean Software and DevOps, Forsgren, Humble and Kim 2018
availability

Ability for teams to ensure their product or service can be accessed by end users

Software delivery + availability = SDO performance

Elite performers are 3.5X more likely to have strong availability practices

capabilities that drive high performance
what is continuous delivery?

The ability to get changes—features, configuration changes, bug fixes, experiments—into production or into the hands of users safely and quickly in a sustainable way.

https://continuousdelivery.com/
build quality in

“Cease dependence on mass inspection to achieve quality. Improve the process and build quality into the product in the first place”

W. Edwards Deming
key finding: architectural outcomes

can my team...

...make large-scale changes to the design of its system without the permission of somebody outside the team or depending on other teams?

...complete its work without needing fine-grained communication and coordination with people outside the team?

...deploy and release its product or service on demand, independently of other services the product or service depends upon?

...do most of its testing on demand, without requiring an integrated test environment?

...perform deployments during normal business hours with negligible downtime?
key finding: doing cloud right

On-demand self-service: 46% agreed or strongly agreed. Only 22% of teams are doing cloud right!

Broad network access: 46% agreed or strongly agreed. Teams that use these essentials characteristics are 23X more likely to be elite performers.

Resource Pooling: 43% agreed or strongly agreed.

Rapid elasticity: 45% agreed or strongly agreed.

Measured service: 48% agreed or strongly agreed.

cloud in regulated environments

https://devops-research.com/research.html
monitoring and observability

Teams with a comprehensive monitoring and observability solution were **1.3 times more likely to be an elite performer.**

Having a monitoring and observability solution **positively contributed to SDO performance.**

**MONITORING** is tooling or a technical solution that allows teams to watch and understand the state of their systems and is based on gathering predefined sets of metrics or logs.

**OBSERVABILITY** is tooling or a technical solution that allows teams to actively debug their system and explore properties and patterns they have not defined in advance.

Fun stats fact: monitoring and observability load together.
which of these measure effective test practices?

• Developers primarily create & maintain acceptance tests
• QA primarily create & maintain acceptance tests
• Primarily created & maintained by outsourced party
• When automated tests pass, I’m confident the software is releasable
• Test failures are likely to indicate a real defect
• It’s easy for developers to fix acceptance tests
• Developers share a common pool of test servers to reproduce failures
• Developers create on demand test environments
• Developers use their own dev environments to reproduce failures
which of these measure effective test practices?

• Developers primarily create & maintain acceptance tests
• QA primarily create & maintain acceptance tests
• Primarily created & maintained by outsourced party
• When automated tests pass, I’m confident the software is releasable
• Test failures are likely to indicate a real defect
• It’s easy for developers to fix acceptance tests
• Developers share a common pool of test servers to reproduce failures
• Developers create on demand test environments
• Developers use their own dev environments to reproduce failures
continuous testing

previous practices plus...

• continuously reviewing and improving test suites to better find defects and keep complexity and cost under control

• allowing testers to work alongside developers throughout the software development and delivery process

• performing manual test activities such as exploratory testing, usability testing, and acceptance testing throughout the delivery process

• having developers practice test-driven development by writing unit tests before writing production code for all changes to the codebase

• being able to get feedback from automated tests in less than ten minutes both on local workstations and from a CI server
lean management

Effective WIP limits that drive process improvement

Use of visual displays to monitor quality, productivity and work in process

Use of app perf and infra monitoring tools to make business decisions

Together, the factors on the left model "Lean Management", which leads to...

A generative, performance-oriented culture (per Westrum's model)

Higher levels of IT performance (higher throughput and stability)

Higher levels of org performance (productivity, market share, profitability)

Lower levels of burnout
lean product management

Regularly gathering, broadcasting, & implementing customer feedback

Splitting work into small batches that can be completed in <1 week & released frequently

Teams have a good understanding of the flow of work from business to customers

Teams have the authority to create and change specifications as part of the dev process

Westrum culture

Software delivery performance

Less burnout

Organizational performance
culture impacts performance

A generative, performance-oriented culture (per Westrum's model)

Higher levels of IT performance (higher throughput and stability)

Higher levels of org performance (productivity, market share, profitability)
## high trust culture

### how organizations process information

<table>
<thead>
<tr>
<th>Pathological (power oriented)</th>
<th>Bureaucratic (rule oriented)</th>
<th>Generative (performance oriented)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low cooperation</td>
<td>Modest cooperation</td>
<td>High cooperation</td>
</tr>
<tr>
<td>Messengers shot</td>
<td>Messengers neglected</td>
<td>Messengers trained</td>
</tr>
<tr>
<td>Responsibilities shirked</td>
<td>Narrow responsibilities</td>
<td>Risks are shared</td>
</tr>
<tr>
<td>Bridging discouraged</td>
<td>Bridging tolerated</td>
<td>Bridging encouraged</td>
</tr>
<tr>
<td>Failure leads to scapegoating</td>
<td>Failure leads to justice</td>
<td>Failure leads to enquiry</td>
</tr>
<tr>
<td>Novelty crushed</td>
<td>Novelty leads to problems</td>
<td>Novelty implemented</td>
</tr>
</tbody>
</table>

Westrum, “A Typology of Organizational Cultures” | [http://bmj.co/1BRGh5q](http://bmj.co/1BRGh5q)
effective teams

1. Psychological Safety
   Team members feel safe to take risks and be vulnerable in front of each other.

2. Dependability
   Team members get things done on time and meet Google’s high bar for excellence.

3. Structure & Clarity
   Team members have clear roles, plans, and goals.

4. Meaning
   Work is personally important to team members.

5. Impact
   Team members think their work matters and creates change.

https://rework.withgoogle.com/blog/five-keys-to-a-successful-google-team/
dealing with failure

in a complex, adaptive system failure is inevitable

when accidents happen, human error is the starting point of a blameless post-mortem

ask: how can we get people better information?

ask: how can we detect and limit failure modes?
The immediate response from everyone around was to ask, “What help do you need?”
disaster recovery testing

“For DiRT-style events to be successful, an organization first needs to accept system and process failures as a means of learning... We design tests that require engineers from several groups who might not normally work together to interact with each other. That way, should a real large-scale disaster ever strike, these people will already have strong working relationships”

—Kripa Krishnan, Director, Cloud Operations, Google

Highly Aligned, Loosely Coupled

• Highly Aligned
  – Strategy and goals are clear, specific, broadly understood
  – Team interactions focused on strategy and goals, rather than tactics
  – Requires large investment in management time to be transparent and articulate and perceptive

• Loosely Coupled
  – Minimal cross-functional meetings except to get aligned on goals and strategy
  – Trust between groups on tactics without previewing/approving each one – so groups can move fast
  – Leaders reaching out proactively for ad-hoc coordination and perspective as appropriate
  – Occasional post-mortems on tactics necessary to increase alignment

https://www.slideshare.net/reed2001/culture-1798664/94-Highly_Aligned_Looseley_Coupled_Highly
“If it's a good idea, go ahead and do it. It is much easier to apologize than it is to get permission.”

—Rear Admiral Grace Hopper, USN, 1906-1992
thank you!

To receive the following:

- An excerpt from *Accelerate*
- 30% off Jez's new video course: creating high performance organizations
- 50% off Jez's CD video training, interviews with Eric Ries, and more
- A copy of this presentation
- A 100 page excerpt from *Lean Enterprise*
- An excerpt from *The DevOps Handbook*
- A 20m preview of Jez's Continuous Delivery video workshop

Just pick up your phone and send an email
To: jezhumble@sendyourslides.com
Subject: devops

© 2016-8 DevOps Research and Assessment LLC and Jez Humble and Associates LLC