If You Don’t Know Where You’re Going, It Doesn’t Matter How Fast You Get There

Nicole Forsgren, PhD @nicolefv
Jez Humble @jezhumble
© 2018 DevOps Research and Assessments LLC. CC-BY-SA
Outline

Where am I going?

Why should I care?

How do I improve performance & quality?

How should I measure performance?

What is this culture thing (and how do I measure it)?
Where am I going?
Maturity models are for CHUMPS
Maturity models are for CHUMPS
Where am I going?

Direction. Not a destination.

But what direction?

Is there “one metric that matters?”
IT performance

lead time for changes
release frequency
time to restore service
change fail rate
<table>
<thead>
<tr>
<th>Survey questions</th>
<th>High IT performers</th>
<th>Medium IT performers</th>
<th>Low IT performers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment frequency</strong></td>
<td>On demand (multiple deploys 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>For the primary application or service you work on, how often does your organization deploy code?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Lead time for changes</strong></td>
<td>Less than one hour</td>
<td>Between one week and one month</td>
<td>Between one week and one month*</td>
</tr>
<tr>
<td>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)?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mean time to recover (MTTR)</strong></td>
<td>Less than one hour</td>
<td>Less than one day</td>
<td>Between one day and one week</td>
</tr>
<tr>
<td>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)?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Change failure rate</strong></td>
<td>0-15%</td>
<td>0-15%</td>
<td>31-45%</td>
</tr>
<tr>
<td>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)?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Note: Low performers were lower on average (at a statistically significant level), but had the same median as the medium performers.
IT performance matters!

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


...for nonprofits too

high performers were also twice as likely to exceed objectives in:

- quantity of goods and services
- operating efficiency
- customer satisfaction
- quality of products or services
- achieving organization or mission goals.
The DevOps Movement

A cross-functional community of practice dedicated to the study of building, evolving and operating rapidly changing, secure, resilient systems at scale.
Quality

New work vs. unplanned work

High performers
- New work: 49%
- Unplanned work: 21%
- Other work: 30%

Low performers
- New work: 38%
- Unplanned work: 27%
- Other work: 35%

Legend:
- New work
- Unplanned work or rework
- Other work (meetings, routine maintenance, etc.)
How should I measure performance?
Common Mistakes

▪ Outputs vs. Outcomes
▪ Individual/local vs. Team/global
▪ Some common examples:
  Lines of code
  Velocity
  Utilization
Common Mistakes: Lines of Code

▪ More is better?
  – Bloated software
  – Higher maintenance costs
  – Higher cost of change
▪ Less is better?
  – Cryptic code that no one can read
▪ Ideal: solve business problems with most efficient code
Common Mistakes: Velocity

- Agile: problems are broken down into stories, which are assigned “points” of estimated effort to complete
- At end of sprint, total points signed off by customer is recorded = velocity
- Velocity is a capacity planning tool. NOT a productivity tool.
- Why doesn’t this work for productivity?
  - Velocity is a relative measure, not absolute. So: bad for comparing teams
  - Gaming by inflating estimates
  - Focus on team completion at the expense of collaboration (a global goal)
Common Mistakes: Utilization

- Utilization is only good up to a point
- Higher utilization is better?
  - High utilization doesn’t allow slack for unplanned work
  - Queue theory: as utilization approaches 100%, lead times approach infinity
  - Once you hit higher and higher levels of utilization (a poor goal of productivity), teams will take longer and longer to get work done
# High Trust Culture

## How Organizations Process Information

<table>
<thead>
<tr>
<th>Pathological (<em>power oriented</em>)</th>
<th>Bureaucratic (<em>rule oriented</em>)</th>
<th>Generative (<em>performance oriented</em>)</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
# Likert-type scale

How strongly do you agree or disagree with the following? On my team...

<table>
<thead>
<tr>
<th></th>
<th>Strongly disagree</th>
<th>Disagree</th>
<th>Somewhat disagree</th>
<th>Neither agree nor disagree</th>
<th>Somewhat agree</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information is actively sought.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Messengers are not punished when they deliver news of failures or other bad news.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Responsibilities are shared.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Cross-functional collaboration is encouraged and rewarded.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Failure causes inquiry.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>New ideas are welcomed.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Failures are treated primarily as opportunities to improve the system.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
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?
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

Kripa Krishnan | http://queue.acm.org/detail.cfm?id=2371297
Conclusions

We CAN have it all, or at least tempo AND stability.

DevOps culture & practices have a measurable impact on IT & org perf & quality

Culture can be measured and changed

Technology and agility do matter - but it’s not enough
Want more Measurement Goodness?

To receive the following:
- A 93-page excerpt of *Accelerate: The Science of DevOps*
- This presentation
- DORA’s ROI whitepaper: Forecasting the Value of DevOps Transformations
- Metrics Guidance whitepaper
- Tactics for Leading Change whitepaper
- My ACM Queue article on DevOps Metrics with Mik Kersten: Your Biggest Mistake Might Be Collecting the Wrong Data

Just grab your phone and send an email:
- To: nicolefv@sendyourslides.com
- Subject: devops