Two talks:

1. “State of DevOps”
2. “We Fear Change”
Three views of DevOps in 2024

What DevOps means to people.

Coté – March 15th, 2024
1. DevOps is

A centralized developer tools group
Market Definition/Description

This Magic Quadrant for DevOps Platforms is the first version of this Magic Quadrant. It replaces the Market Guide for Value Stream Delivery Platforms.

Gartner defines DevOps platforms as those that provide fully integrated capabilities to enable continuous delivery of software using Agile and DevOps practices. These capabilities run the gamut of the software development life cycle (SDLC) and include product planning, version control, continuous integration, test automation, continuous deployment, release orchestration, automating security and compliance policies, monitoring, and observability. DevOps platforms support team collaboration, secure software development and measurement of software delivery metrics.

DevOps platforms simplify creation, maintenance and management of the components...
2.

DevOps is

Culture
<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 inquiry</td>
</tr>
<tr>
<td>Novelty crushed</td>
<td>Novelty leads to problems</td>
<td>Novelty implemented</td>
</tr>
</tbody>
</table>

DevOps is

Simplifying yaml for developers (i.e., “Kubernetes”)
What benefits has your organization realized from operating Kubernetes?

(Choose all that apply)

- Improved resource utilization
- Eased application upgrades and maintenance
- Enabled the process of moving to the cloud
- Increase efficiency of hybrid model
- Shortened software development cycles
- Containerized monolithic applications
- Reduced public cloud costs

“The initial experience, that ‘wall of yaml,’ as we like to say, when you configure your first application can be a little bit daunting. And, I’m sorry about that. **We never really intended folks to interact directly with that subsystem.** It’s, more or less, developed a life of its own over time.”

Craig McLuckie, SpringOne 2021
4.

DevOps is

Building & running platforms
DevOps Is Dead. Embrace Platform Engineering

Platforms provide golden paths, with recommended tools and best security practices built in, reducing cognitive load while preserving developer freedom.

Sep 22nd, 2022 7:10am by Aenis Stewart
<table>
<thead>
<tr>
<th>Platform Capabilities</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Infrastructure</strong></td>
<td><strong>Data</strong></td>
<td><strong>Messaging</strong></td>
<td><strong>Boundaries</strong></td>
</tr>
<tr>
<td>compute, network,</td>
<td>databases,</td>
<td>queues,</td>
<td>automate build,</td>
</tr>
<tr>
<td>storage</td>
<td>caches, buckets</td>
<td>brokers</td>
<td>test &amp; delivery</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>store artifacts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>in registries</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and repos</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>observe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>workloads</td>
</tr>
</tbody>
</table>

Understanding why people resist using your platform

Coté - February 6th, 2024
We all know that

Changing organizations fails 70% of the time.

Actually,

We have no idea how frequently changing organizations succeeds or fails.

Evolution of the *Star Trek* Analogy
Little bit weird  
Sits closer to the boss  
Thinks too hard  
Pulls levers & turns knobs  
Easily excited  
Yells a lot in emergencies

DevSecOps

HA HA!

BUSINESS
“This is a 1 ½ CIO Job.”

Management & workers often have different incentives & motivations

**Adding a focus on opportunities to software developer productivity metrics can offer clearer paths to improvement.**

<table>
<thead>
<tr>
<th>Focus areas by level</th>
<th>DORA metrics</th>
<th>SPACE metrics</th>
<th>Opportunity-focused metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outcomes focus</strong></td>
<td>Are you delivering products satisfactorily?</td>
<td>Code-review timing</td>
<td>Satisfaction with engineering system</td>
</tr>
<tr>
<td>System level</td>
<td>Deployment frequency</td>
<td>Velocity/flow through the system</td>
<td>Inner/outer loop time spent</td>
</tr>
<tr>
<td></td>
<td>Customer satisfaction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reliability (uptime)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Team level</strong></td>
<td>Lead time for changes</td>
<td>Story points completed</td>
<td>Quality of documentation</td>
</tr>
<tr>
<td></td>
<td>Change failure rate</td>
<td>Handoffs</td>
<td>Developer Velocity Index benchmark*</td>
</tr>
<tr>
<td></td>
<td>Time to restore service</td>
<td></td>
<td>Contribution analysis</td>
</tr>
<tr>
<td></td>
<td>Code-review velocity</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Individual level</strong></td>
<td>Developer satisfaction</td>
<td>Interruptions</td>
<td>Contribution analysis</td>
</tr>
<tr>
<td></td>
<td>Retention</td>
<td></td>
<td>Talent capability score</td>
</tr>
</tbody>
</table>

1. Google’s DevOps research and assessment team, which developed these outcome metrics.
2. Satisfaction and well-being, performance, activity, communication and collaboration, and efficiency and flow. Github and Microsoft Research developed these metrics, which aim to look at developer well-being as a measurement at the individual level.
3. Non exhaustive.

A thriving organization focuses on satisfaction, flow, ease, happiness

<table>
<thead>
<tr>
<th>Causes of thriving</th>
<th>Because a developer is…</th>
</tr>
</thead>
</table>
| Agency             | 1) able to voice disagreement with team definitions of success  
|                    | 2) has a voice in how their contributions are measured |
| Motivation & Self-Efficacy | 1) motivated when working on code at work  
|                    | 2) can see tangible progress most of the time  
|                    | 3) is working on the type of code work they want to work on  
|                    | 4) is confident that even when working in code is unexpectedly difficult, they will solve their problems |
| Learning Culture   | 1) learning new skills as a developer  
|                    | 2) able to share the things they learn at work |
| Support & Belonging| 1) supported to grow, learn, and make mistakes by their team  
|                    | 2) agrees they are accepted for who they are by their team |

**TABLE 1: EXAMPLE DEVEX METRICS**

<table>
<thead>
<tr>
<th>FEEDBACK LOOPS</th>
<th>COGNITIVE LOAD</th>
<th>FLOW STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PERCEPTIONS</strong></td>
<td><strong>WORKFLOWS</strong></td>
<td><strong>KPIs</strong></td>
</tr>
<tr>
<td>Human attitudes and opinions</td>
<td>System and process behaviors</td>
<td>North star metrics</td>
</tr>
</tbody>
</table>
| - Satisfaction with automated test speed and output  
| - Satisfactory with time it takes to validate a local change  
| - Satisfactory with time it takes to deploy a change to production | - Time it takes to generate CI results  
| - Code review turnaround time  
| - Deployment lead time (time it takes to get a change released to production) | - Time it takes to get answers to technical questions  
| - Manual steps required to deploy a change  
| - Frequency of documentation improvements | - Overall perceived ease of delivering software  
| - Employee engagement or satisfaction  
| - Perceived productivity |
| - Perceived complexity of codebase  
| - Ease of debugging production systems  
| - Ease of understanding documentation | | |
| - Perceived ability to focus and avoid interruptions  
| - Satisfactory with clarity of task or project goals | |
| - Perceived disruptiveness of being on-call |

Management vs. workers often have different urgency & motivation to change

<table>
<thead>
<tr>
<th>Exec’s View</th>
<th>Work the Same</th>
<th>Transform!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensation</td>
<td>$</td>
<td>$$$$</td>
</tr>
<tr>
<td>Risk</td>
<td>HIGH</td>
<td>HIGH</td>
</tr>
<tr>
<td>Outcome</td>
<td>💣</td>
<td>👍</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Staff’s View</th>
<th>Work the Same</th>
<th>Transform!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensation</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td>Risk</td>
<td>LOW</td>
<td>HIGH</td>
</tr>
<tr>
<td>Outcome</td>
<td>👍</td>
<td>🤷</td>
</tr>
</tbody>
</table>
The people who do the work (should) decide how to change the work.

Sources: "DevOps is Enterprise Wide," Nigel Thurlow, DevOpsDays Dallas 2022.
We believe that we need to reimagine banking to make banking simple, seamless, as well as invisible to allow our customers to live more bank less.

Siew Choo Soh, DBS Bank
Containers will not fix your broken culture (and other hard truths)

BRIDGET KROMHOUT

We focus so often on technical anti-patterns, neglecting similar problems inside our social structures. Spoiler alert: the solutions to many difficulties that seem technical can be found by examining our interactions with others. Let’s talk about five things you’ll want to know when working with those pesky creatures known as humans.
“We are building this platform not for us, we are building it for Mercedes-Benz developers.”

Thomas Müller, Mercedes-Benz
Find the Developer Toil, Confusion, Blockers

What are we making?
- We have a strong vision for our product, and we’re doing important work together every day to fulfill that vision.
- I have the context I need to confidently make changes while I’m working.
- I am proud of the work I have delivered so far for our product.
- I am learning things that I look forward to applying to future products.
- My workstation seems to disappear out from under me while I’m working.
- It’s easy to get my workstation into the state I need to develop our product.
- What aspect of our workstation setup is painful?
- It’s easy to run our software on my workstation while I’m developing it.
- I can boot our software up into the state I need with minimal effort.
- What aspect of running our software locally is painful? What could we do to make it less painful?
- It’s easy to run our test suites and to author new ones.
- Tests are a stable, reliable, seamless part of my workflow.
- Test failures give me the feedback I need on the code I am writing.
- What aspect of production support is painful?

What aspect of our release process is painful?
- Our release process (CI/CD) from source control to our story acceptance environment is fully automated.
- If the release process (CI/CD) fails, I’m confident something is truly wrong, and I know I’ll be able to track down the problem.

What aspect of our release process (CI/CD) is painful?
- Our team releases new versions of our software as often as the business needs us to.
- We are meeting our service-level agreements with a minimum of unplanned work.
- When something is wrong in production, we reproduce and solve the problem in a lower environment.

TECHNICAL IMPROVEMENTS

Daily deploys
+30% developer productivity
+78% operational efficiency
60% reduction in incidents
Repaving prod months->weeks->daily

BUSINESS IMPROVEMENTS

65% shift to in-app ordering
+46% enrollment rates
3 ½ weeks to retool loan program
6 months to launch a new business
142% ROI on platform investment
A fine selection of pre-shaved yaks
https://tanzu.vmware.com
Thanks!

https://newsletter.cote.io/

https://cote.io/platform

cote@broadcom.com