Research into DevOps bottlenecks

Yoav Landman
whoami

Yoav Landman
CTO@jfrog
@_yoav_
Why this “research”? 

- We speak with DevOps 
- Many sectors 
- Many types 
- Binaries -> 
  being on the release pipeline
Development

Operations
Development

Operations

QA

Devops!
How long is your release cycle?

- 6 months
- 1 month
- 2 weeks
- 1 day
- 2 hours
Are they happy with it?

- 0% Satisfied
- 20% Satisfied
- 50% Satisfied
- 80% Satisfied
- 100% Satisfied
What’s the “best” cycle?
Vendor vs. user installation

Transparency of Update

Speed of Update

- Manual Install
- Manual Update
- Fully Automated

Liquid Software
Business needs
How do you know it’s good?
I don’t measure it because I’ve had no complaints
When the bug queue is empty we are good
When I get fewer tickets on the process
Automation of every single piece
Other metrics

- Version adoption
- Bug burn-down diagram
- Speed gains “used to be 2 days, now its 2 mins”
- MTTR
- Business metrics*
84% of CEOs expect software to increase profit margins.

50% of CEOs say their industries will be digitally transformed.

46% of product value will be through software.

Gartner, ITOPs 2017

Infrastructure and Operations

2020 Expectations
Where are the bottlenecks?
Education

Automation

Trust
Pain is instructional
• You need to know,
• You need to believe,
• You need to give a sh*t
First push is hard - Keep your eye on the prize

For an older code base this could take more than a year to achieve, but it’s still worth it
What is holding you back?
HUNDREDS OF JOBS LOST AFTER COMPANY ADOPTS DEVOPS
RESIST

FAKE NEWS
Tools and complexity
Manage Complexity Growth

Software LOC doubles every seven years

50% time spent understanding code

90% of cost is maintenance

Augustine's Law

Software grows by an order of magnitude every 10 years
More automation – more liability

Super devops
"organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations"

_ M. Conway_, a computer scientist, computer programmer, and hacker who coined what's now known as Conway's Law
Guide, don’t dictate
Aim for the 80%

<table>
<thead>
<tr>
<th>Tool</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>Invest</td>
</tr>
<tr>
<td>JFrog Artifactory</td>
<td>Invest</td>
</tr>
<tr>
<td>JFrog Xray</td>
<td>Research</td>
</tr>
<tr>
<td>Gradle</td>
<td>Invest</td>
</tr>
<tr>
<td>Maven</td>
<td>Maintain</td>
</tr>
<tr>
<td>Ant</td>
<td>Divest</td>
</tr>
</tbody>
</table>
Let’s talk about trust
Quality is the bottleneck

Automatic testing is not enough

The “big green button” concept is lack of trust in quality
Automatic testing is not enough, but we can use our customers as testers.

- Canary releases
Runtime is everywhere
Runtime is everywhere
2017 Study of Vulnerabilities on DockerHub

- **3,802** Official Images on Docker Hub
  - **93**: Latest

- **127** Median Number of Vulnerabilities per Image on Docker Hub

- **0** Official Images on Docker Hub Without Known Vulnerabilities

Source: "A Study of Security Vulnerabilities on Docker Hub"
DevSecOps == runtime security control

- Getting it safely to the runtime
- Eradicate bad software from our runtime
  - Bad/unverified update
  - Vulnerabilities
  - Malicious binaries
A Question Of control