



# Research into DevOps bottlenecks



Who am I

Baruch Sadogursky

Developer Advocate @jfrog

@jbaruch

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld





Also,  
[jfrog.com/shownotes](http://jfrog.com/shownotes)



# Why “research”? We interview people

- 
- VP R&D @ JFrog
  - CTO @ Bugsee
  - CTO @ Top Israeli Finance Firm
  - VP Engineering at CA Technologies
  - CTO @ ScentBird
  - Architect, a top Fortune-Rated Privately-Held Tech Firm
  - Applications Engineering Manager at Fortune 500 SW Company

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



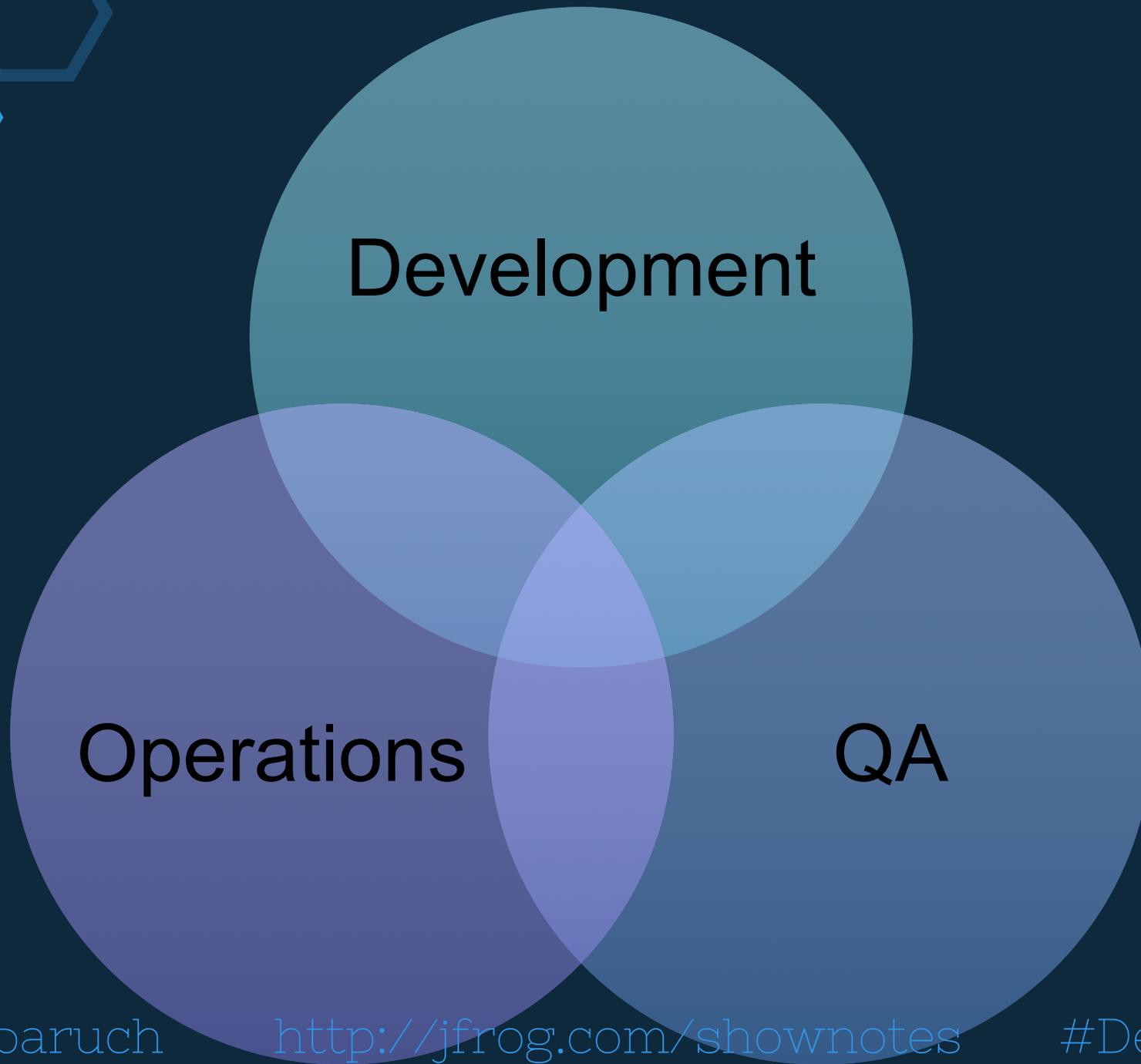
# What is DevOps?



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld

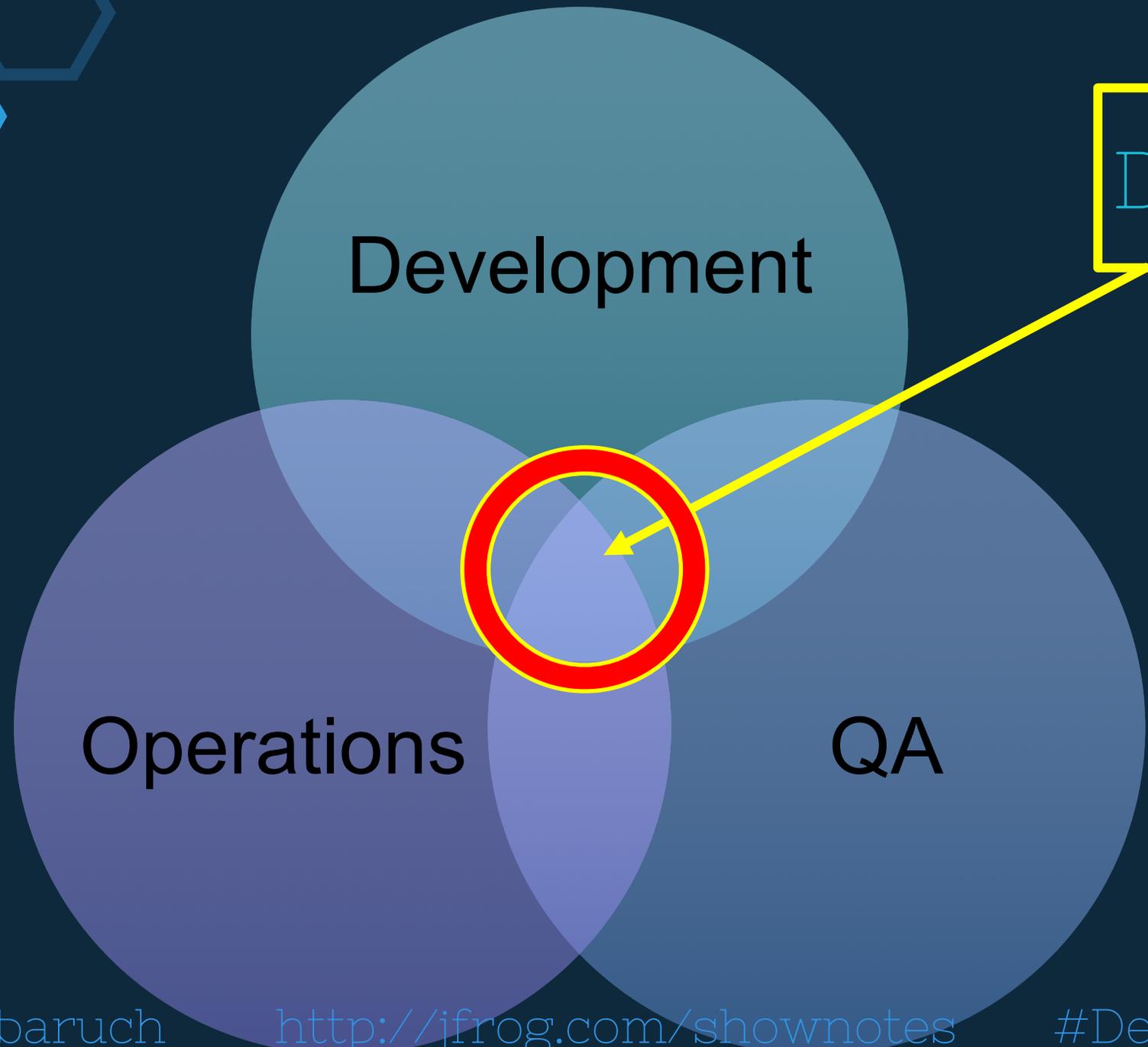


@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld





Devops!





Devops!

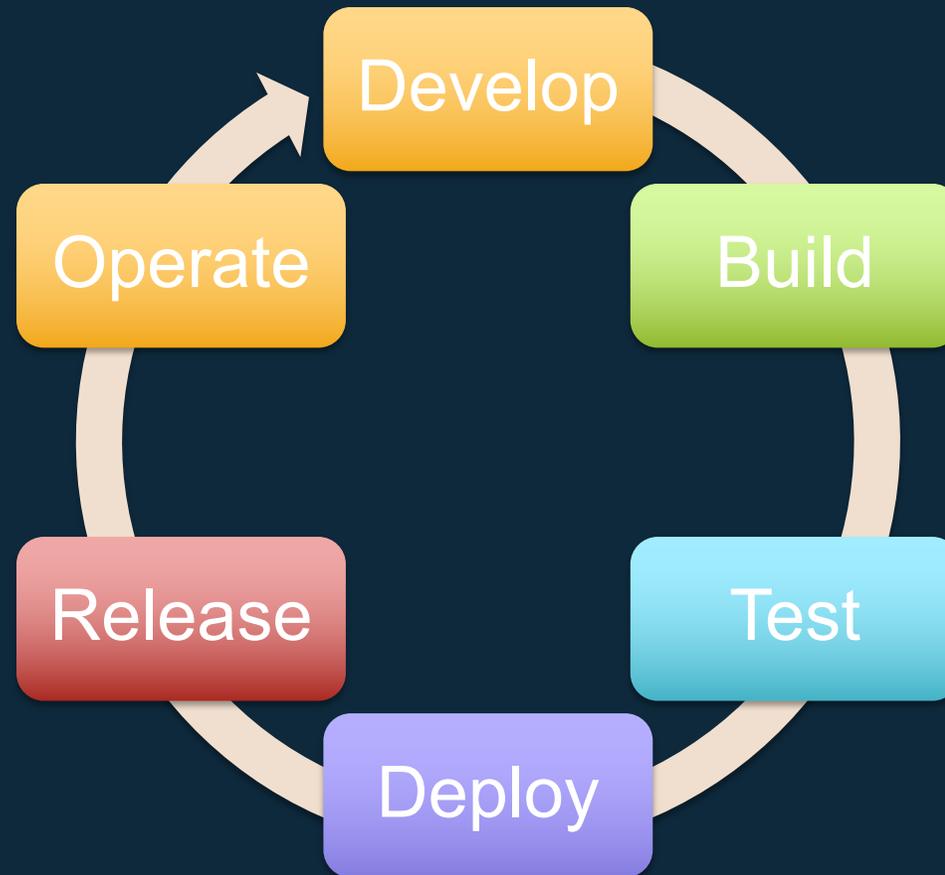




What bottlenecks?



# DevOps and Delivery



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



# Release cycles

6 months

1 month

2 weeks

1 day

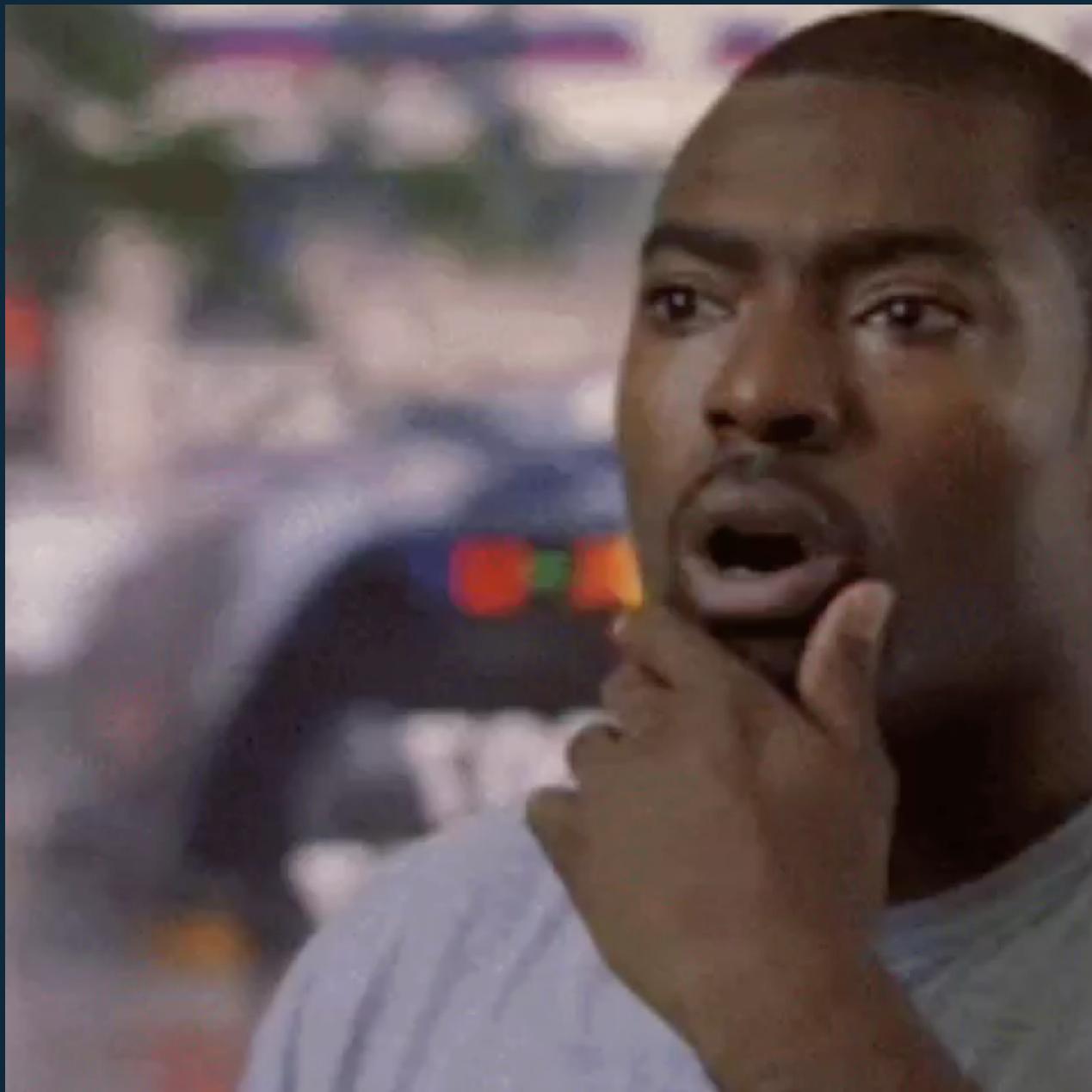
2 hours

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld





@jbaruch

<http://jfrog.com/shownotes>

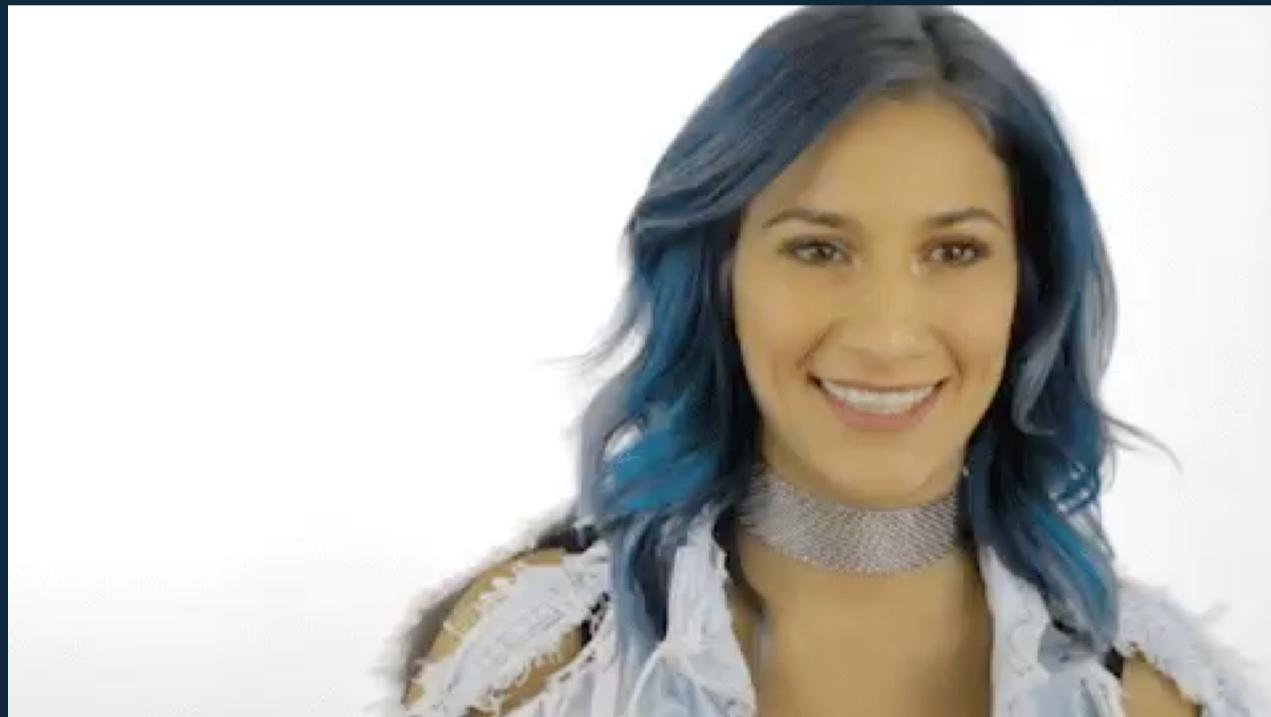
#DevOpsWorld





Are they happy with it?





@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld





How do you know it's good?



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



I don't measure it because I've had no complaints

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



When the bug queue is empty we are good

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



When I get fewer tickets on the process

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



“ Automation of every single piece

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



## Other measures

- Version adoption
- Bug Burn-down diagram
- Productivity Gains “used to be 2 days, now its 2 minutes”
- Quality, No Regressions!
- MTTR
- Business metrics



# What's the right release cycle?







# Why not to update



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld





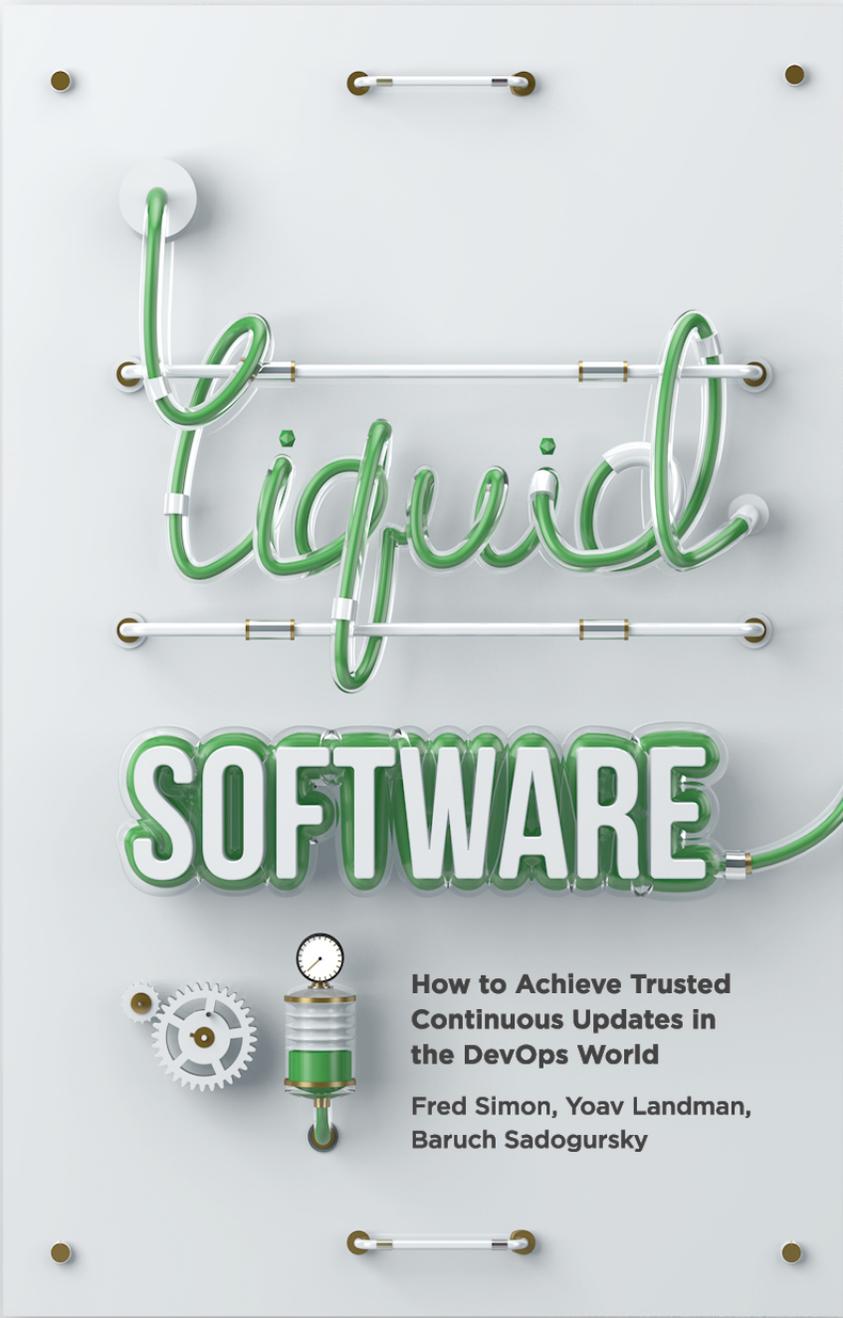
■ Acceptance tests



■ The dreadful updates

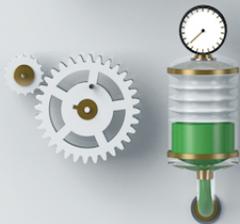


- Impossible dilemma: we need the update, but it will break things



Liquid

SOFTWARE



**How to Achieve Trusted  
Continuous Updates in  
the DevOps World**

Fred Simon, Yoav Landman,  
Baruch Sadogursky

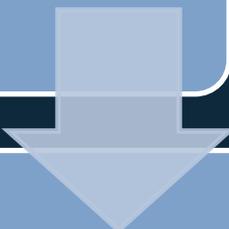




You don't offload quality tests to customers



The customers are always ready to update



You just lost the last excuse for not releasing fast





But what about the quality?...



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld

2018

*Accelerate:*  
**State of DevOps**  
Strategies for a New Economy

Presented by



Diamond Sponsor



Gold Sponsors

**Deloitte.**

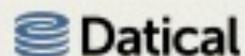


pagerduty

redgate



cloudbees



sumologic

TRICENTIS





# No, not really.

Aspect of Software Delivery Performance	Elite <sup>a</sup>	High	Medium	Low
<b>Deployment frequency</b> 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
<b>Lead time for changes</b> 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 <sup>b</sup>	Between one month and six months <sup>b</sup>
<b>Time to restore service</b> 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
<b>Change failure rate</b> 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%



# No, not really.

Aspect of Software Delivery Performance	Elite <sup>a</sup>	High	Medium	Low
<b>Deployment frequency</b> 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
<b>Lead time for changes</b> 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 <sup>b</sup>	Between one month and six months <sup>b</sup>
<b>Time to restore service</b> 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
<b>Change failure rate</b> 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%



# No, not really.

Aspect of Software Delivery Performance	Elite <sup>a</sup>	High	Medium	Low
<b>Deployment frequency</b> 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
<b>Lead time for changes</b> 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 <sup>b</sup>	Between one month and six months <sup>b</sup>
<b>Time to restore service</b> 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
<b>Change failure rate</b> 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%



# No, not really.

Aspect of Software Delivery Performance	Elite <sup>a</sup>	High	Medium	Low
<b>Deployment frequency</b> 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
<b>Lead time for changes</b> 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 <sup>b</sup>	Between one month and six months <sup>b</sup>
<b>Time to restore service</b> 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
<b>Change failure rate</b> 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%





Education

Automation

Trust

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld





Pain is instructional

@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



You need to know,  
You need to believe,  
You need to give a sh\*t



Also,



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



# “Activation Energy”



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld

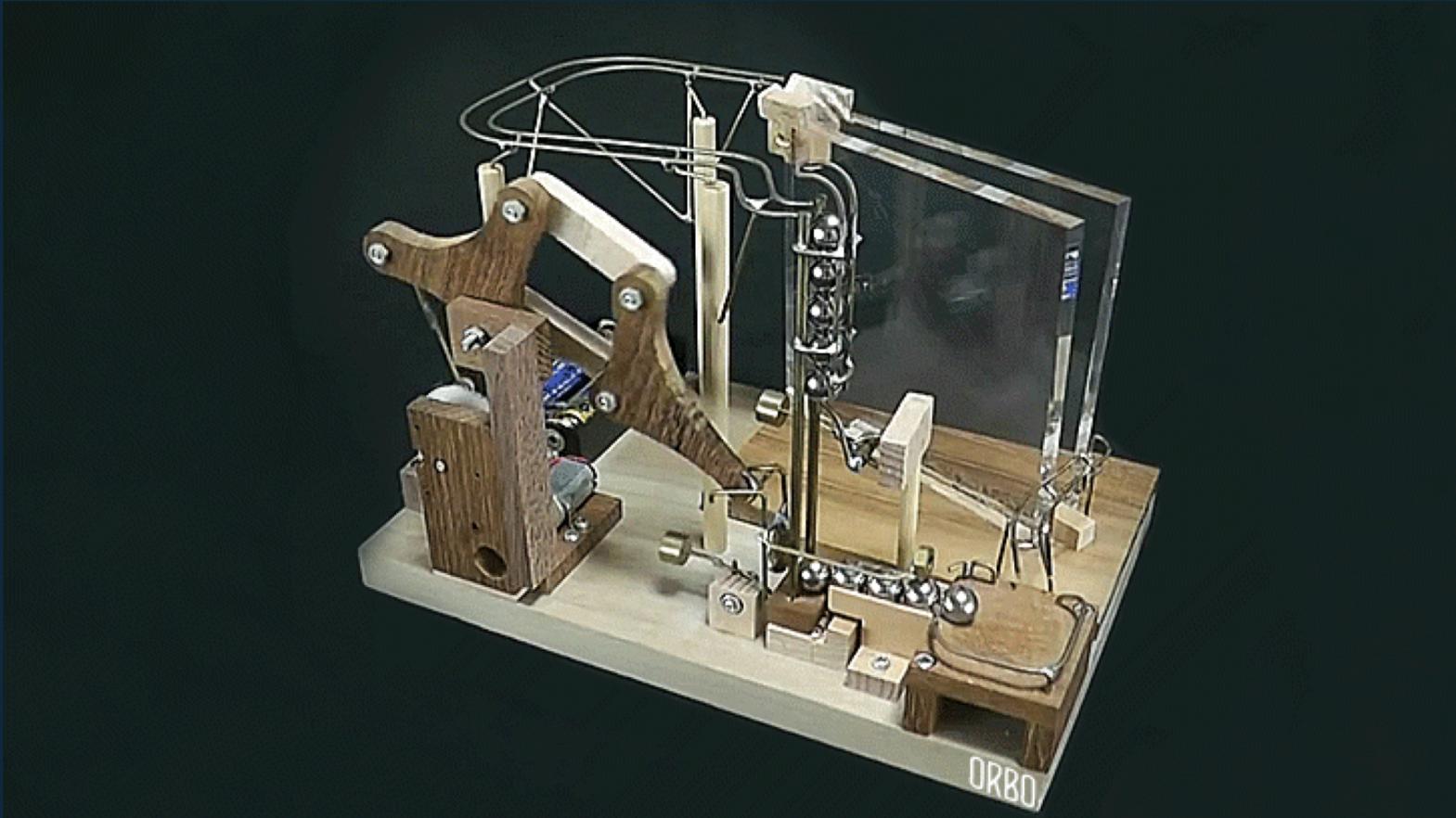


First push is hard

Keep your eye on the prize

For an older code base this could take more than a year to achieve, but its still worth it

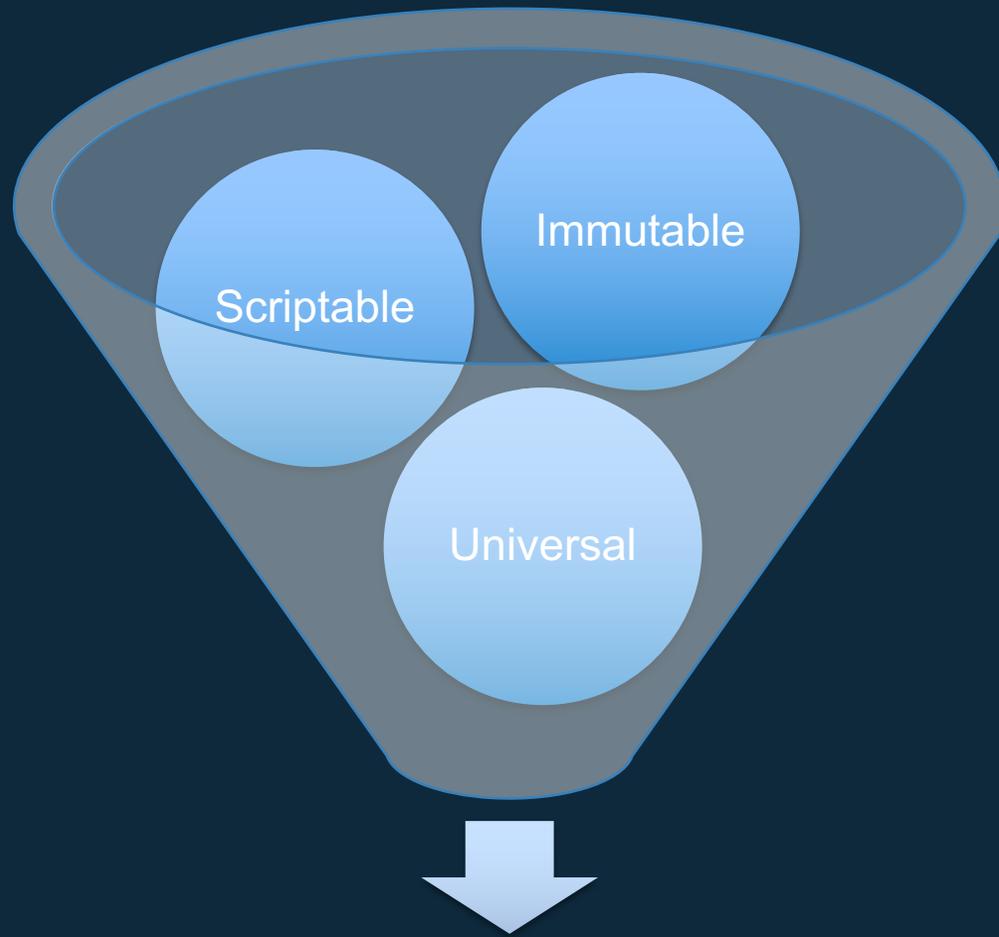
# Tools



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



**Repeatable!**



Guide, don't dictate  
Aim for the 80%

Tool	Rating
Java	Invest
JFrog Artifactory	Invest
JFrog Xray	Research
Gradle	Invest
Maven	Maintain
Ant	Divest





# Let's talk about trust



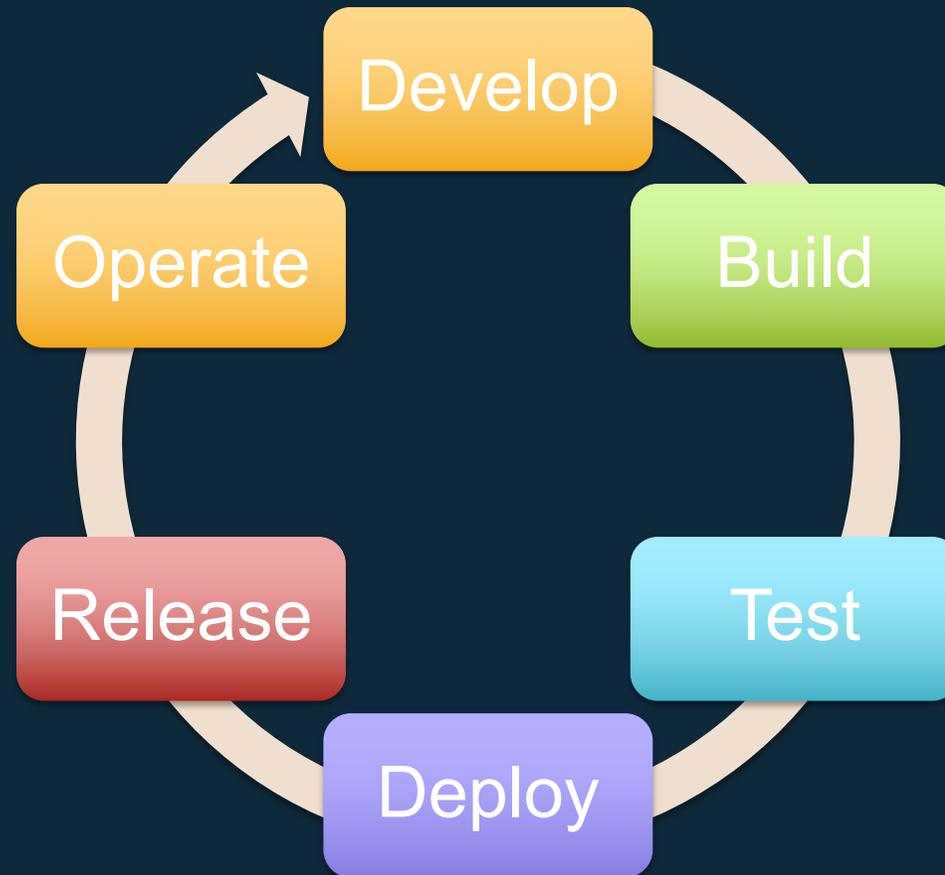
@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



Remember this?

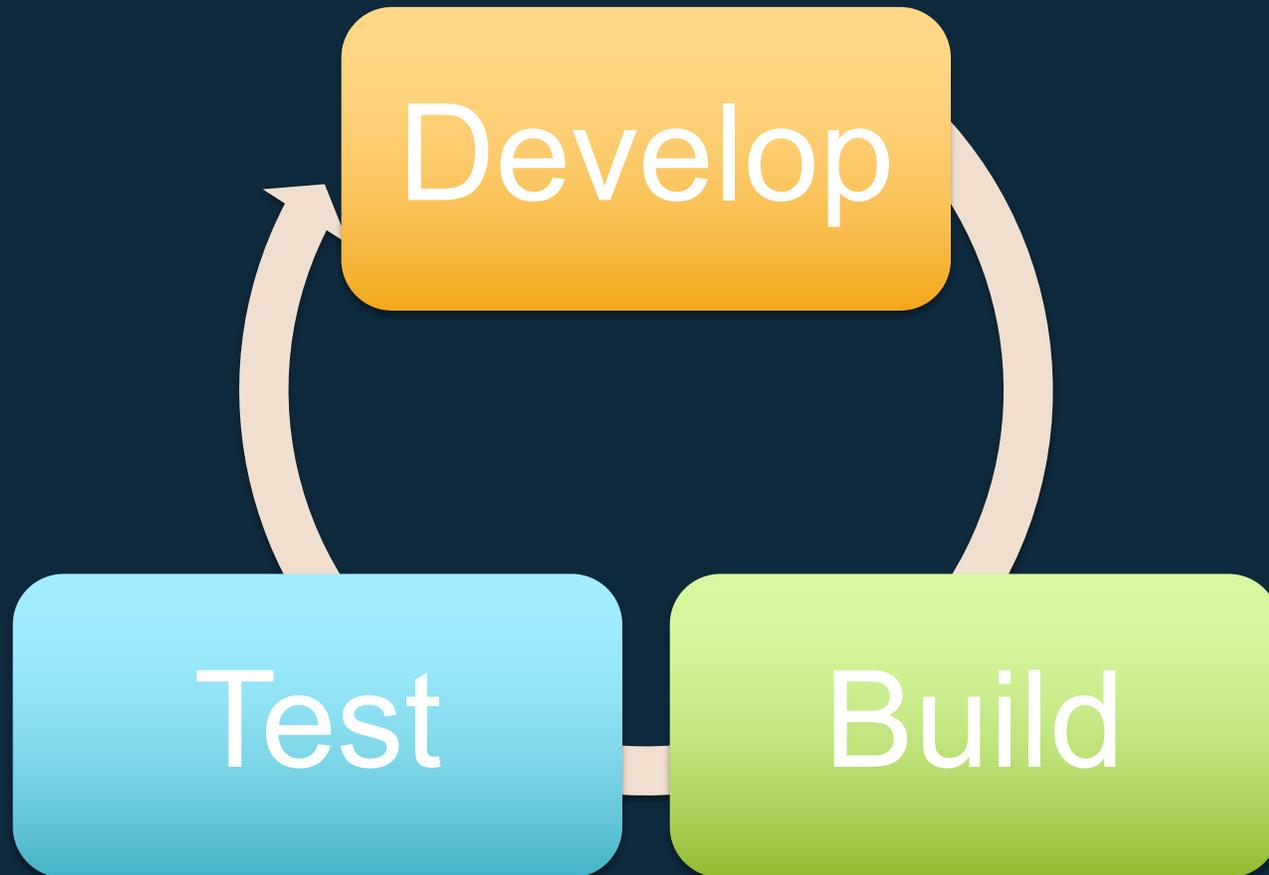


@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld

Some commits should fail!



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld





Quality is the bottleneck

Automatic testing is not enough

The “big green button” concept is lack of trust in quality

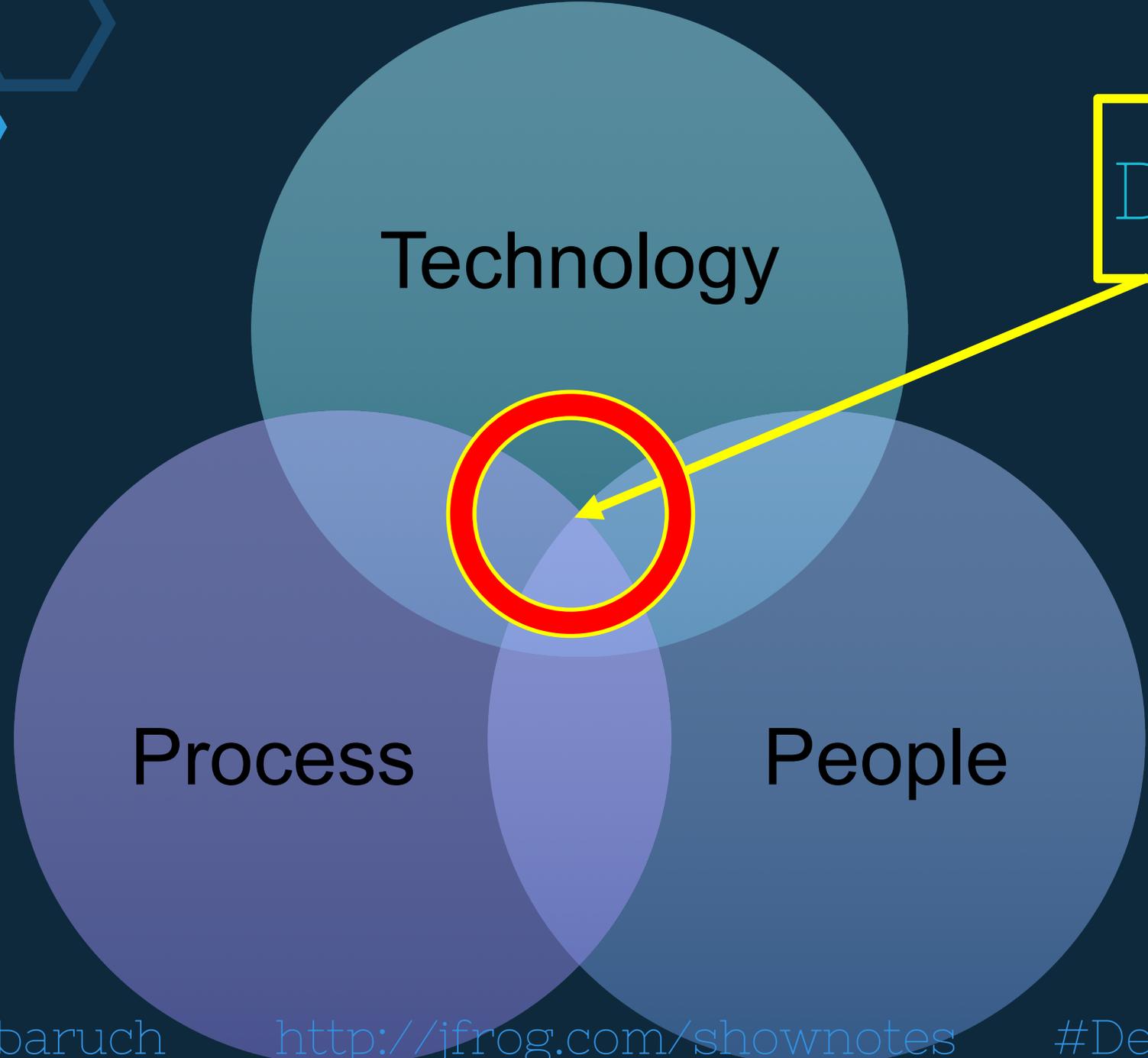


Quality is the bottleneck

“

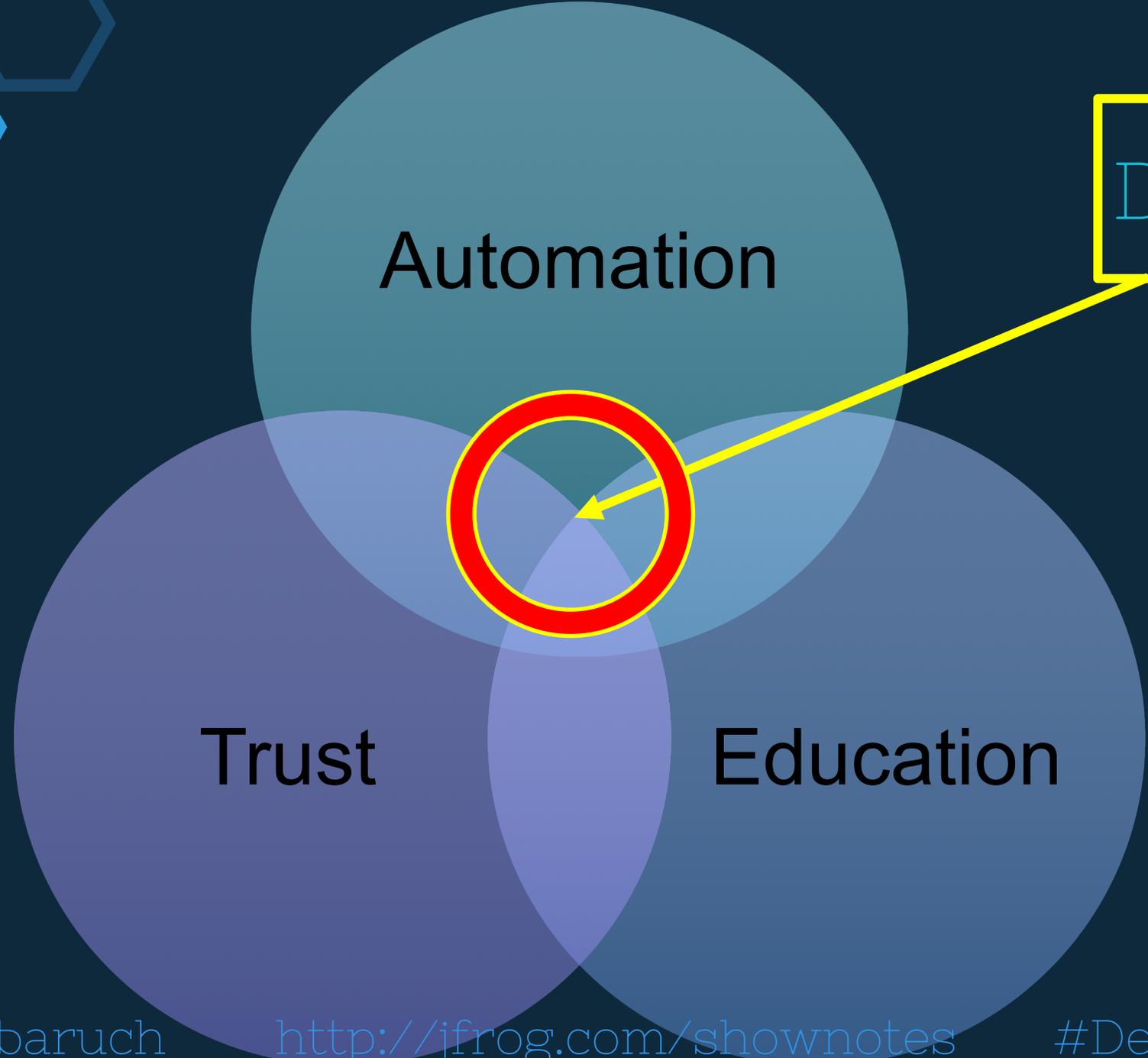
Automatic testing is not enough, but we can use our customers as testers

Canary releases



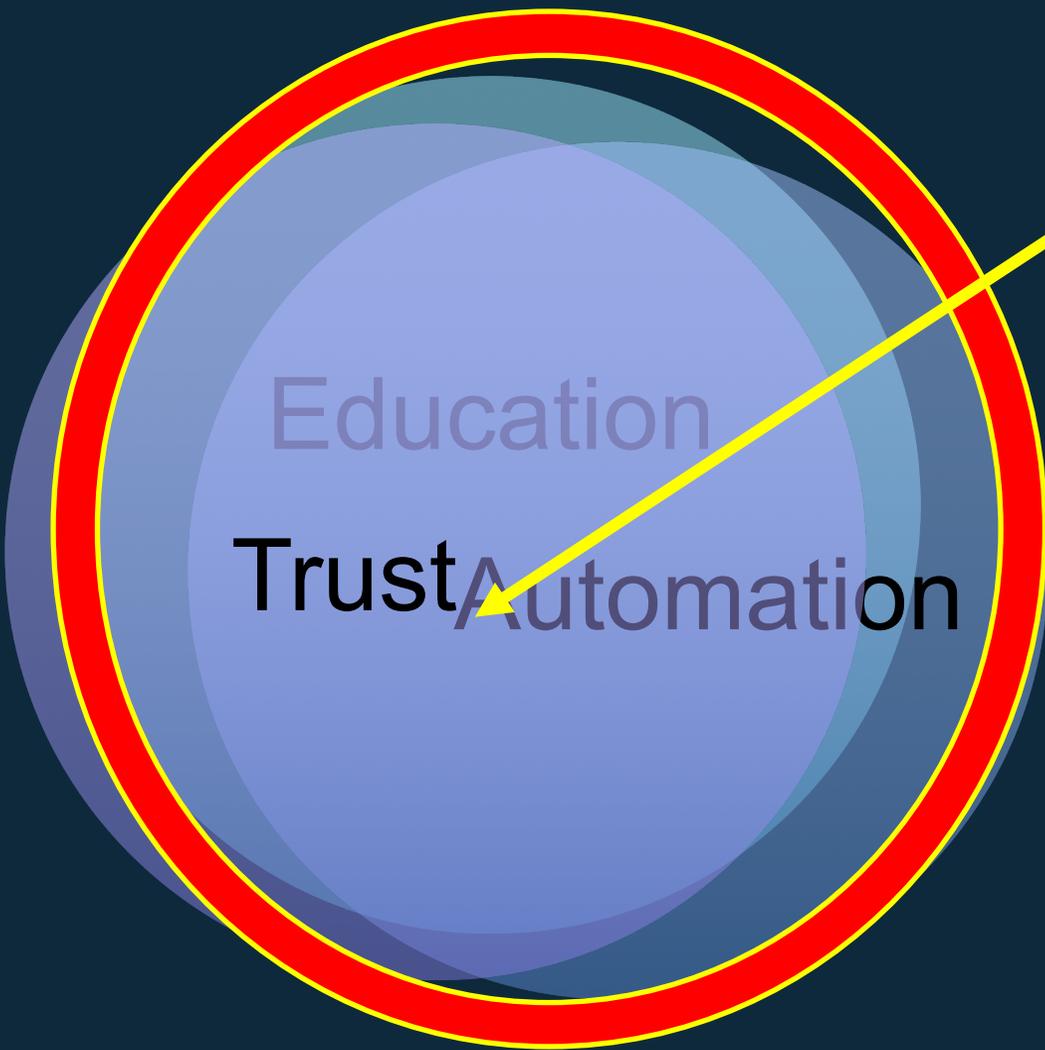
Devops!





Devops!





Devops!



@jbaruch

<http://jfrog.com/shownotes>

#DevOpsWorld



# QA & Show notes

- ◇ @jbaruch
  - ◇ #DevOpsWorld
  - ◇ <http://jfrog.com/shownotes>
    - Video
    - Slides
    - Links
    - Feedback
    - Raffle! (come early)
- 