



# Research into DevOps bottlenecks



Who am I

Baruch Sadogursky

Head of Developer Relations @jfrog

@jbaruch



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus





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
  - Architect, US Army (Contractor)

@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



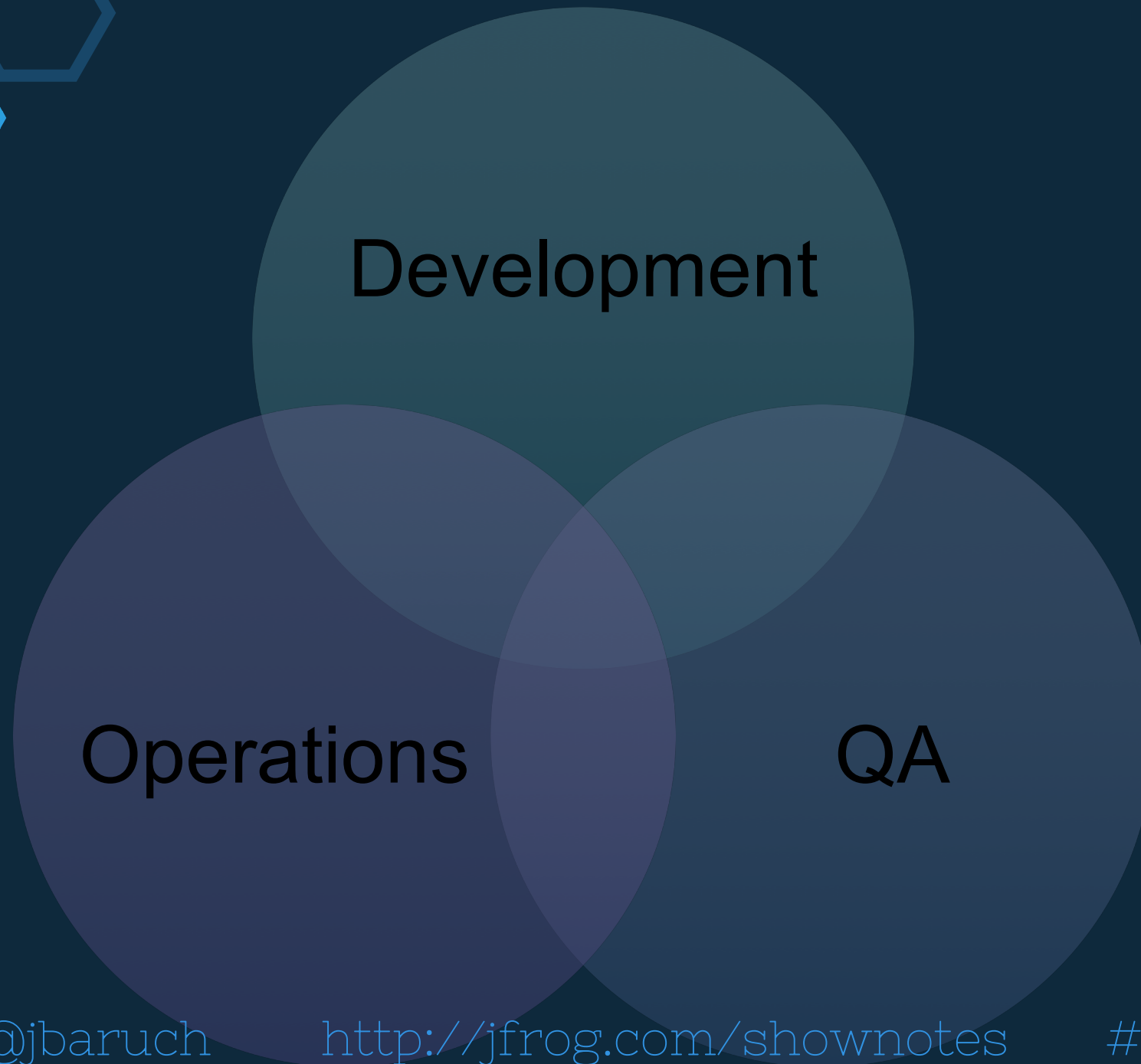
# What is DevOps?



@jbaruch

<http://jfrog.com/shownotes>

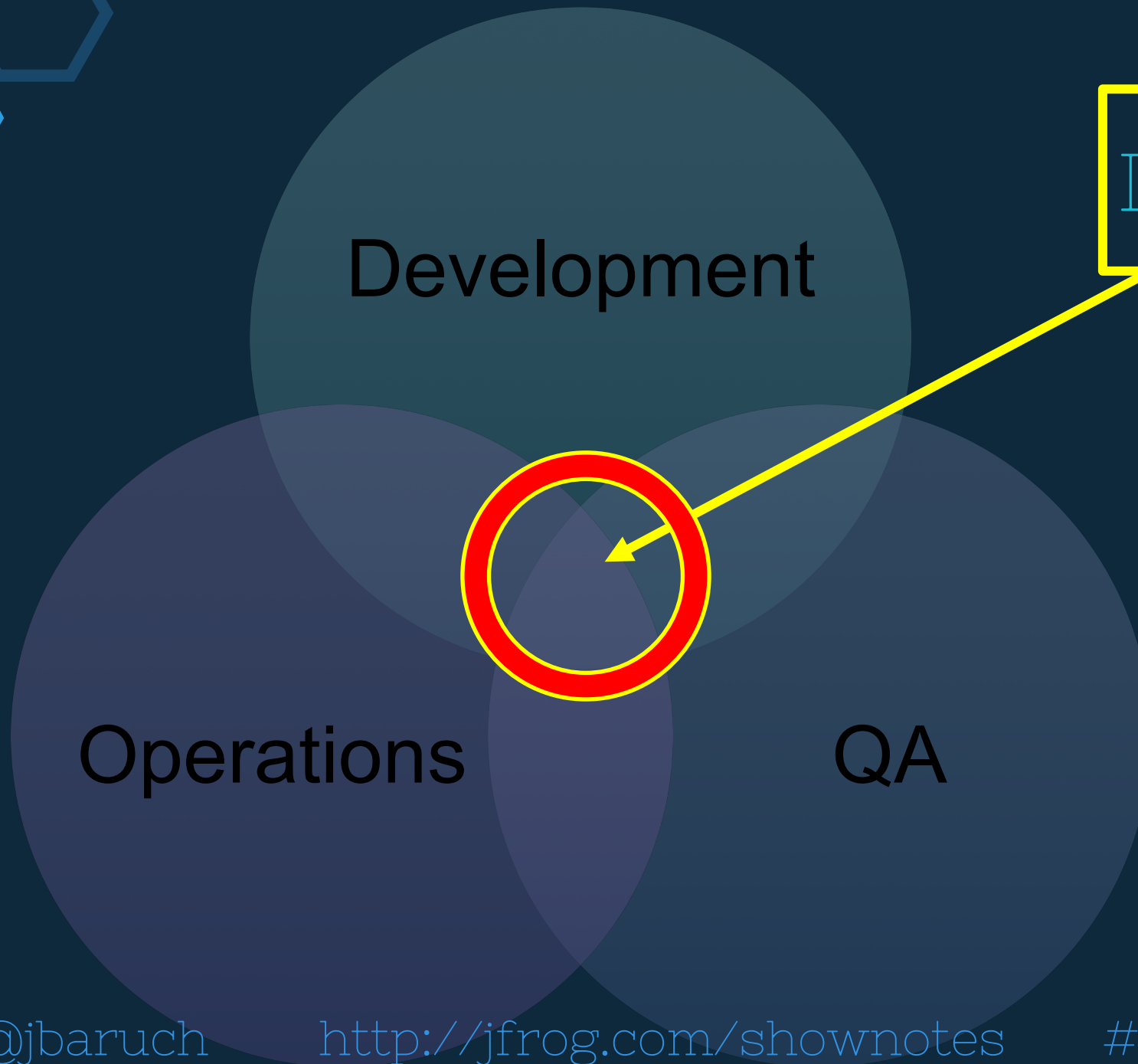
#DevNexus



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



Devops!

Development

Operations

QA





Development  
Operations QA

Devops!

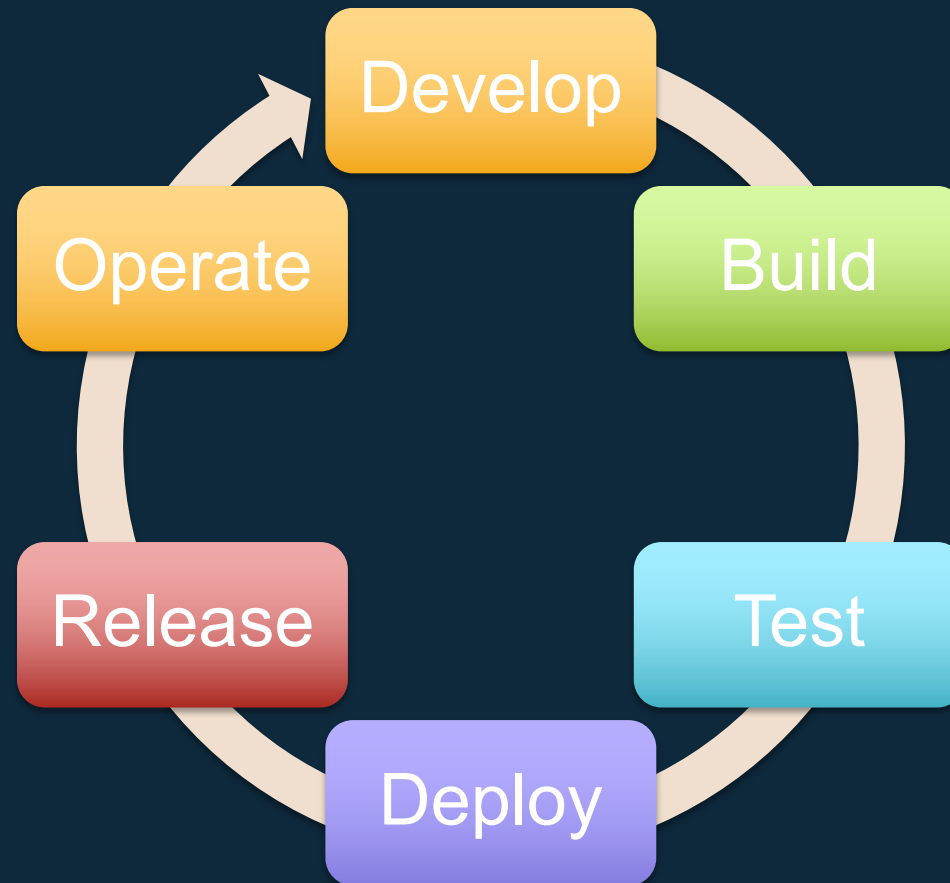


What bottlenecks?





# DevOps and Delivery



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



# Release cycles

2 years

6 months

2 weeks

1 day

2 hours





@jbaruch

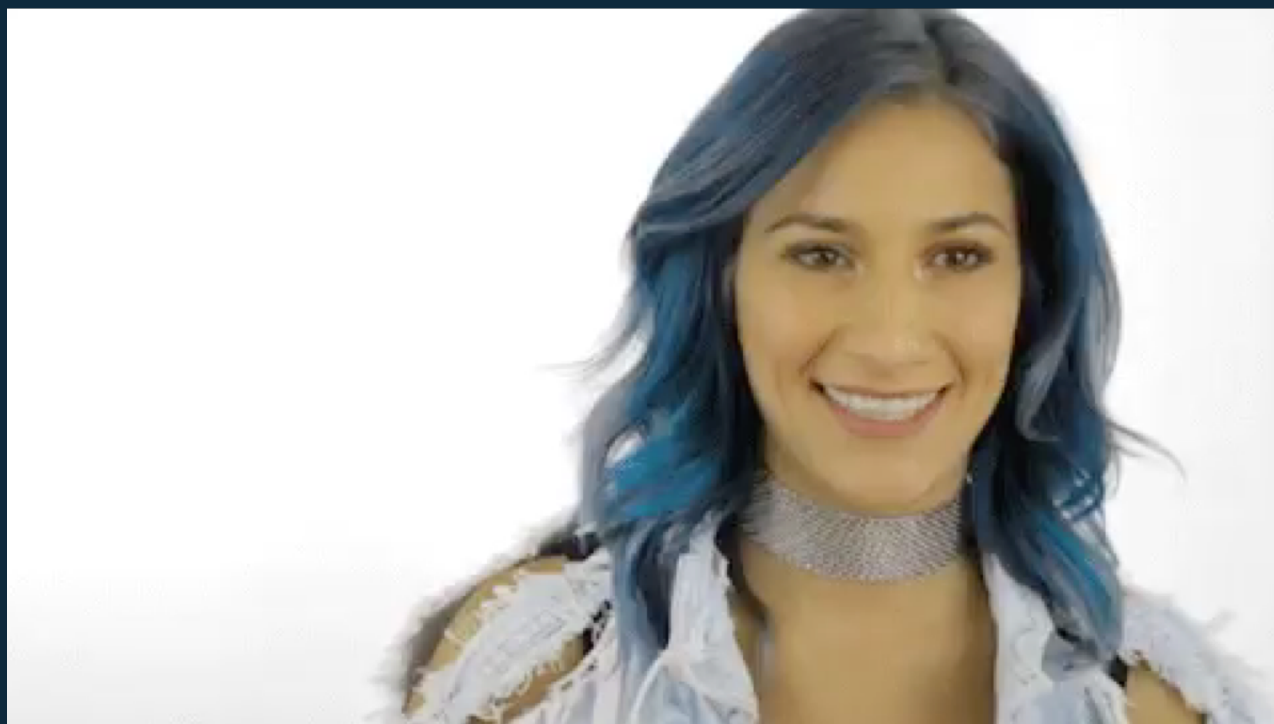
<http://jfrog.com/shownotes>

#DevNexus



Are they happy with it?





@jbaruch

<http://jfrog.com/shownotes>

#DevNexus





How do you know it's good?



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



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 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?



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus







# Why not to update



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus





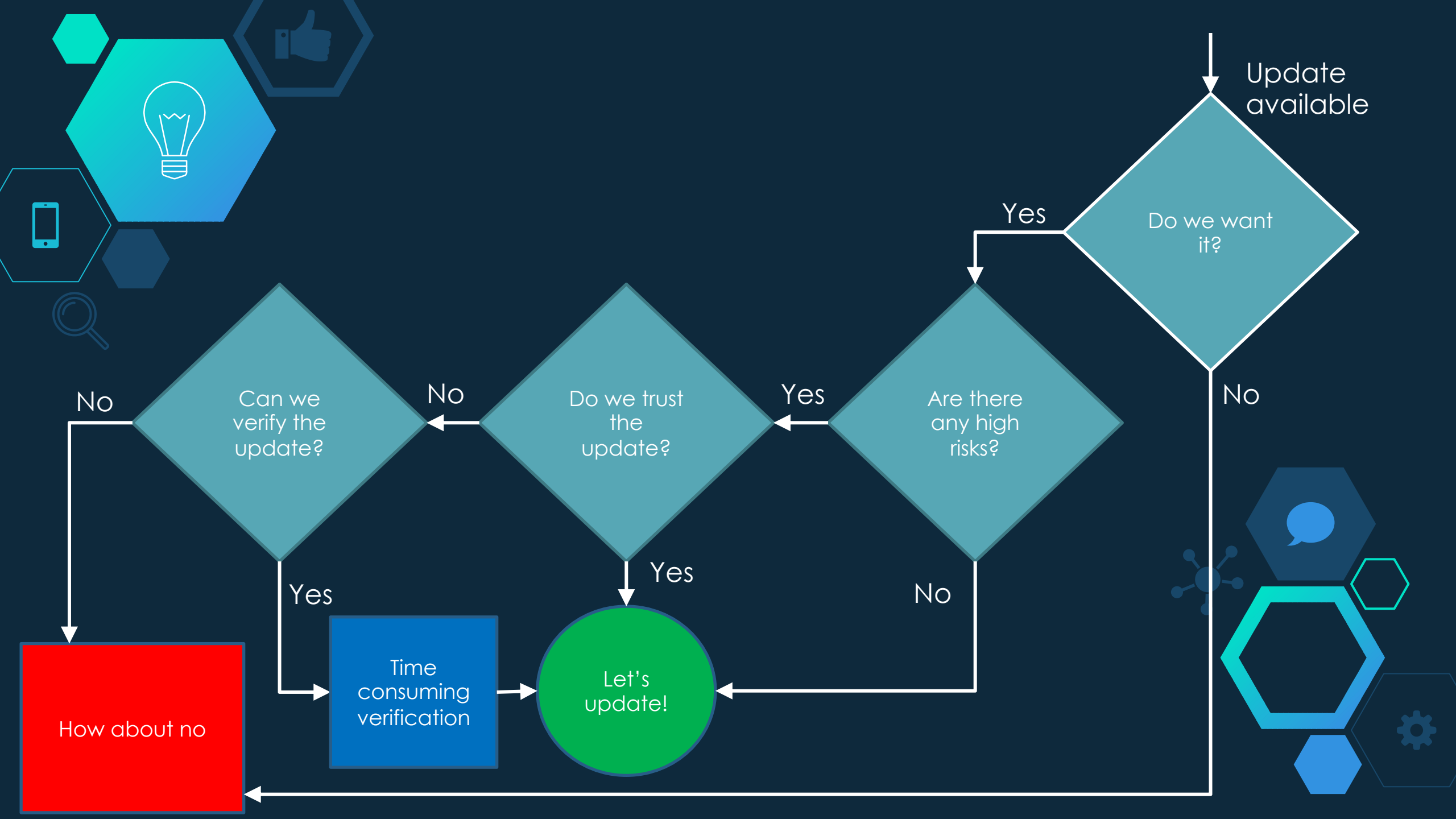
■ Acceptance tests



■ “The dreadful updates



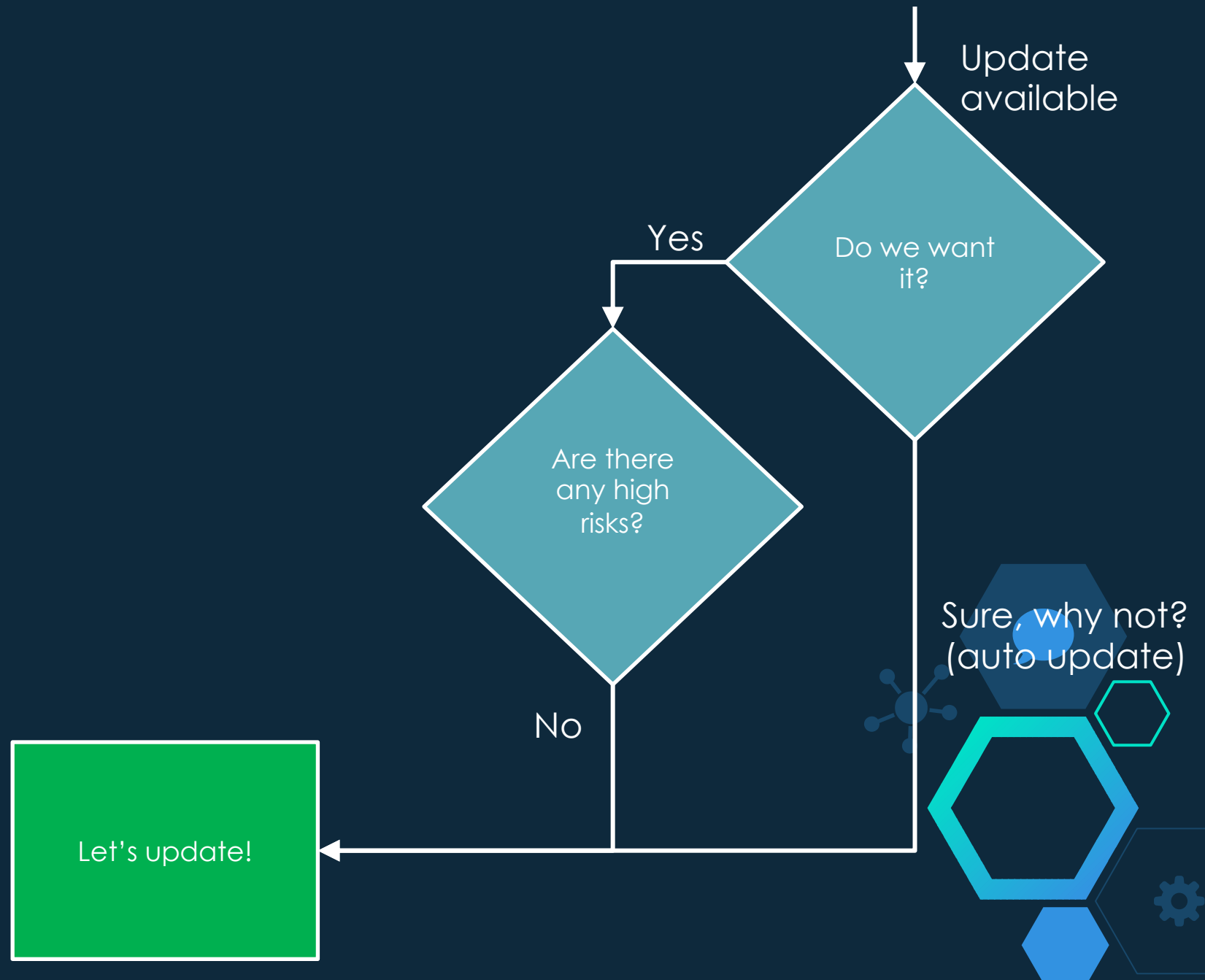
- Impossible dilemma: we need the update, but it will break things





Aspect of Software Delivery Performance	Elite <sup>2</sup>
<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)
<b>Lead time for changes</b>	





Virus Alert!



Hi, I am Albanian virus. But because of poor technology in my country unfortunately I am not able to harm your computer. Please be so kind to delete one of your important files yourself and then forward me to other users. Many thanks for your cooperation. Best regards, Albanian virus.

Yes

No

Cancel







# Google Wifi

## Sincerest Apologies for any Issues

Dear Google Wifi & OnHub customers,

Today we experienced an issue with our Google Accounts engine that may have affected your Google Wifi and OnHub devices. This caused some devices to automatically reset to the initial state you bought them in and they will unfortunately need to be set up again. This has not affected the software or performance of the device but it does need to be re-setup.

Visit this [dedicated support page](#) for more information and step-by-step instructions to get back online. If you set up your Wi-Fi with the same network name and password as before, your household connected devices should automatically reconnect.

If you do not want to set up the devices right now, you can always connect to the setup Wi-Fi network — the network name and password are printed on the base of your primary device (the one wired to the modem). This should get you connected to the internet immediately.

We know how important Wi-Fi is to you and share our sincerest apologies for the inconvenience this has caused.

Ben Brown On behalf of the Google Wifi and OnHub team

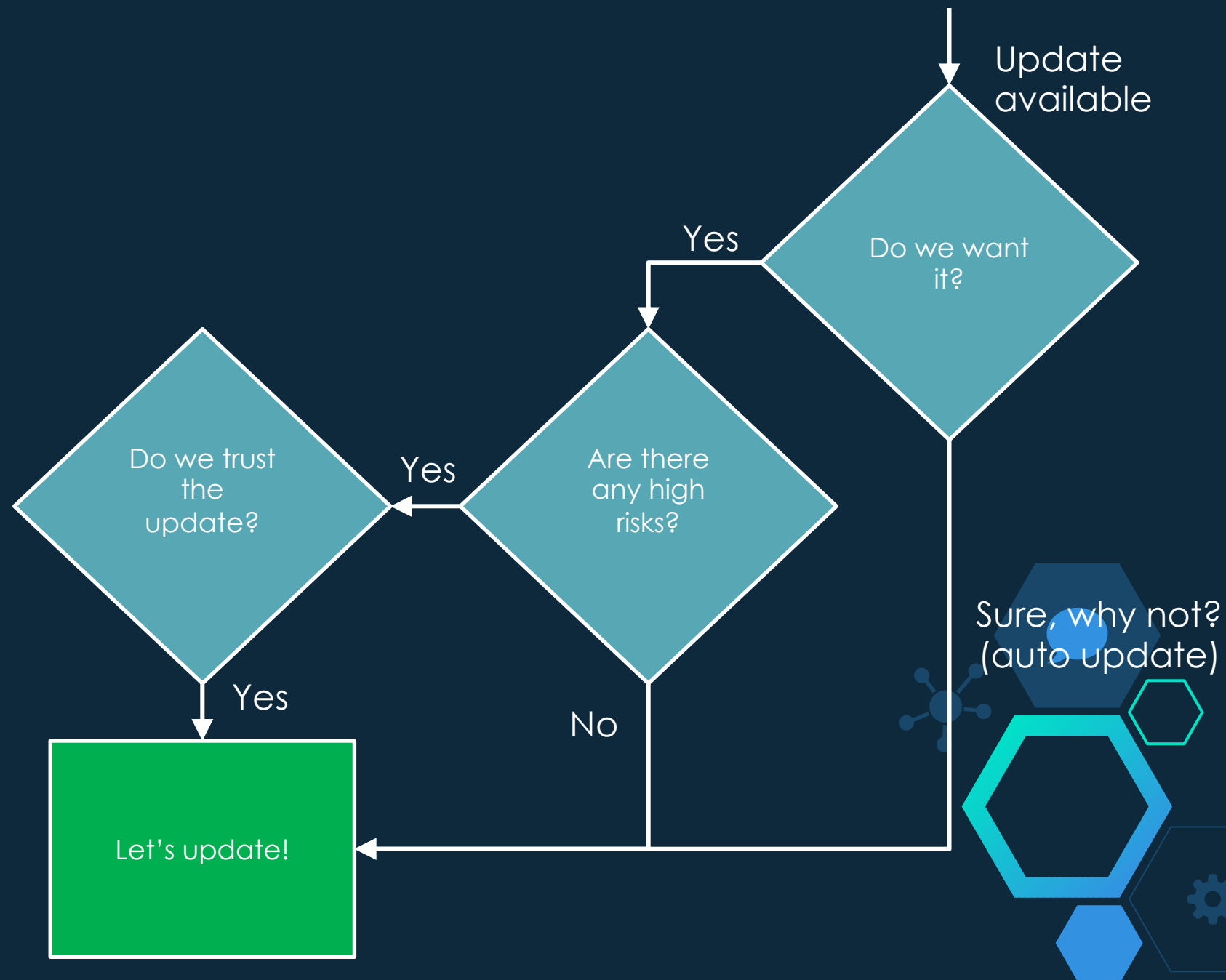


@jbaruch

@jfrog

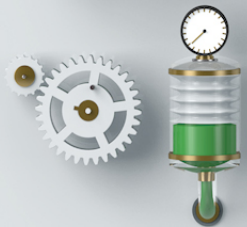
#LiquidSoftware

www.liquidsoftware.com





# 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>

#DevNexus



2018

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

Presented by  
**DORA**  
DEVOPS RESEARCH & ASSESSMENT

Diamond Sponsor  
**Google Cloud**

Gold Sponsors

**Deloitte.**

**xl** XebiaLabs  
Enterprise DevOps

**pagerduty**

**redgate**

**4T REV**

**cloudbees**

**Datical**

**sumologic**

**TRICENTIS**

**GitLab**



# 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>

#DevNexus





Pain is instructional



“

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





Also,



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



# “Activation Energy”



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus





@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



First push is hard

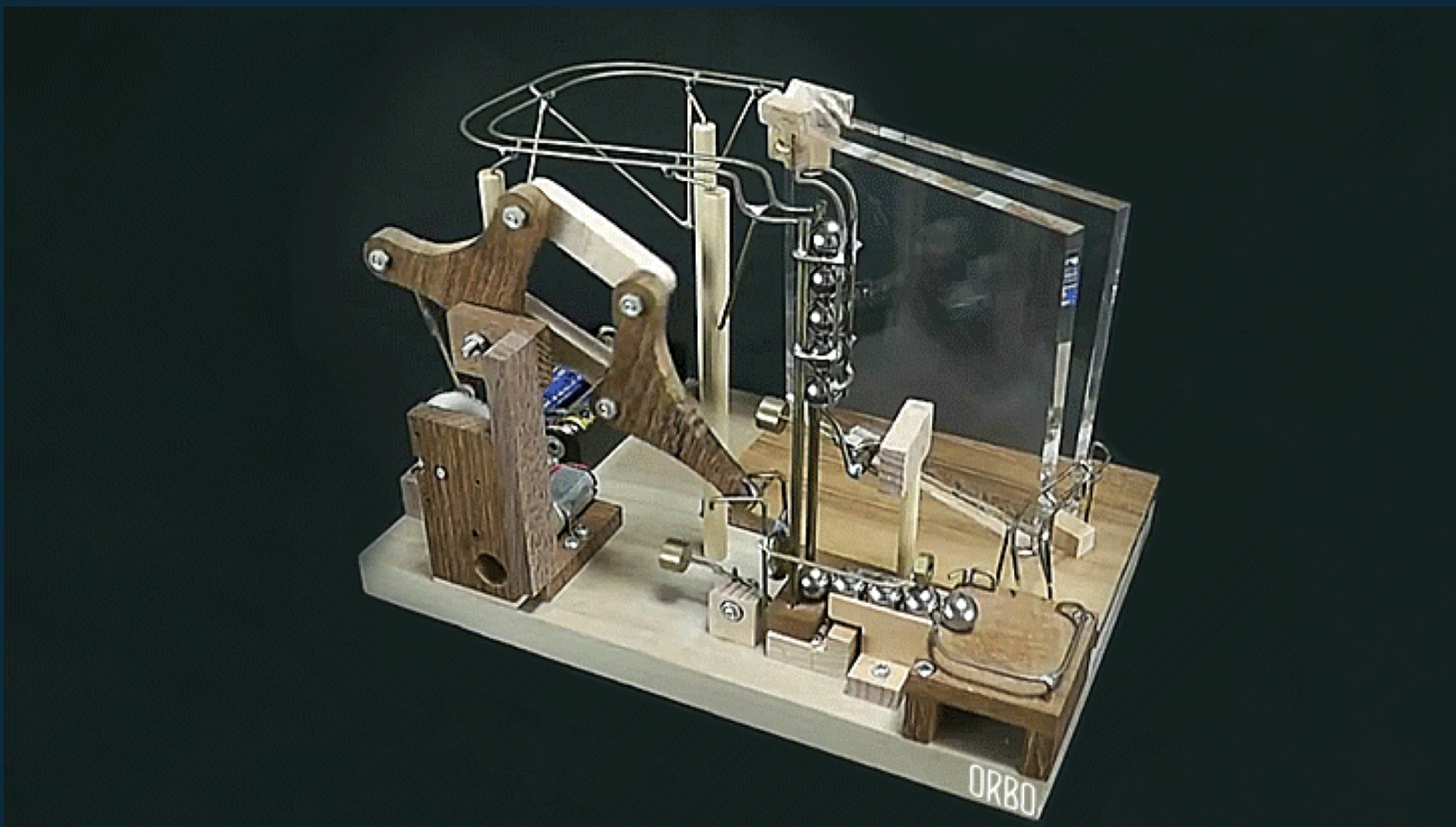
Keep your eye on the prize

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





# Tools

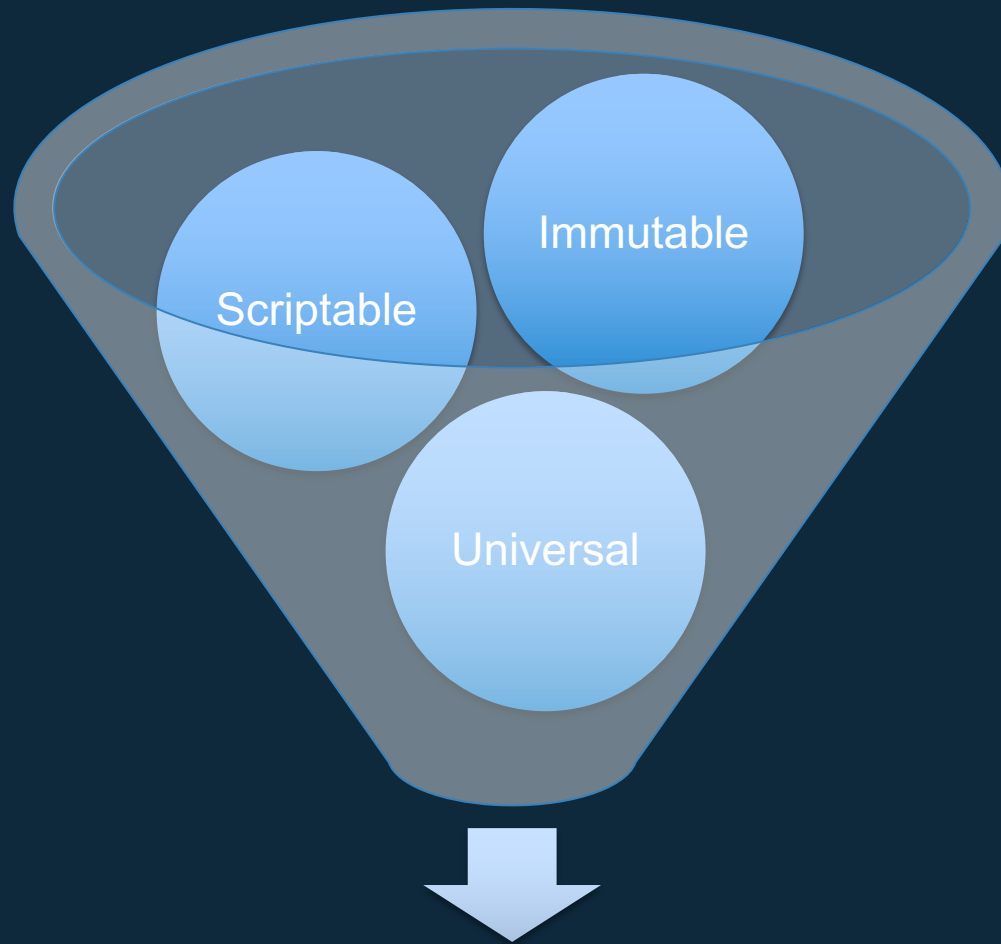


@jbaruch


<http://jfrog.com/shownotes>

#DevNexus






**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

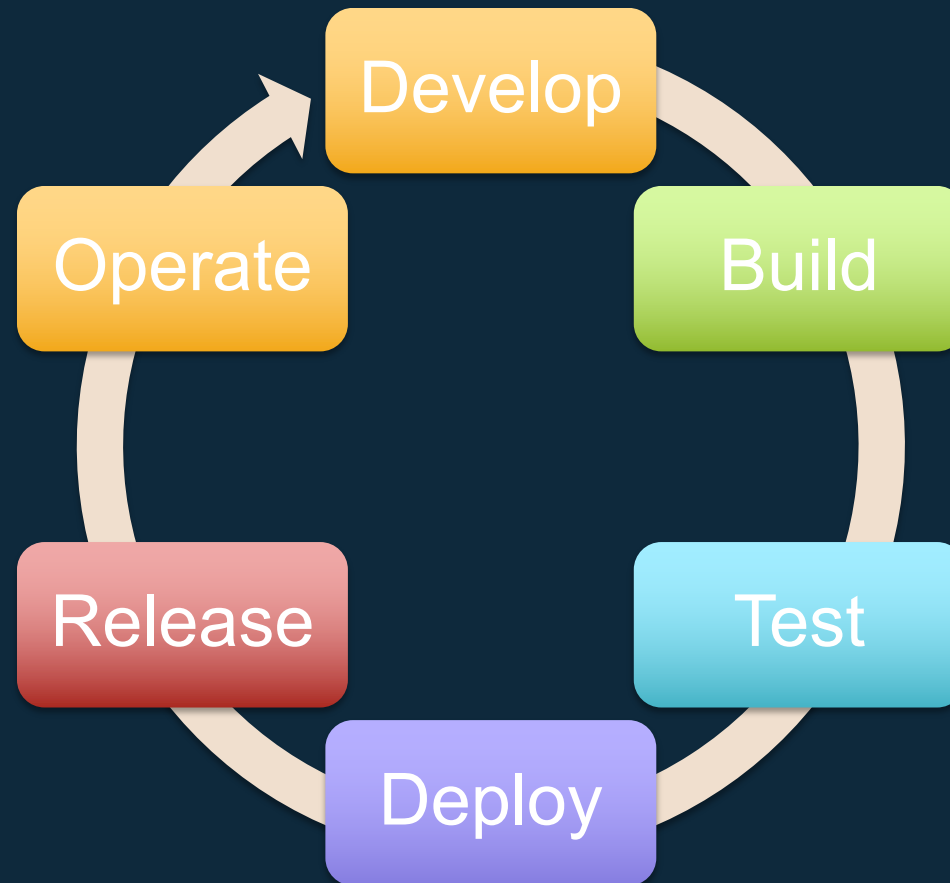


@jbaruch

<http://jfrog.com/shownotes>

#DevNexus

Remember this?



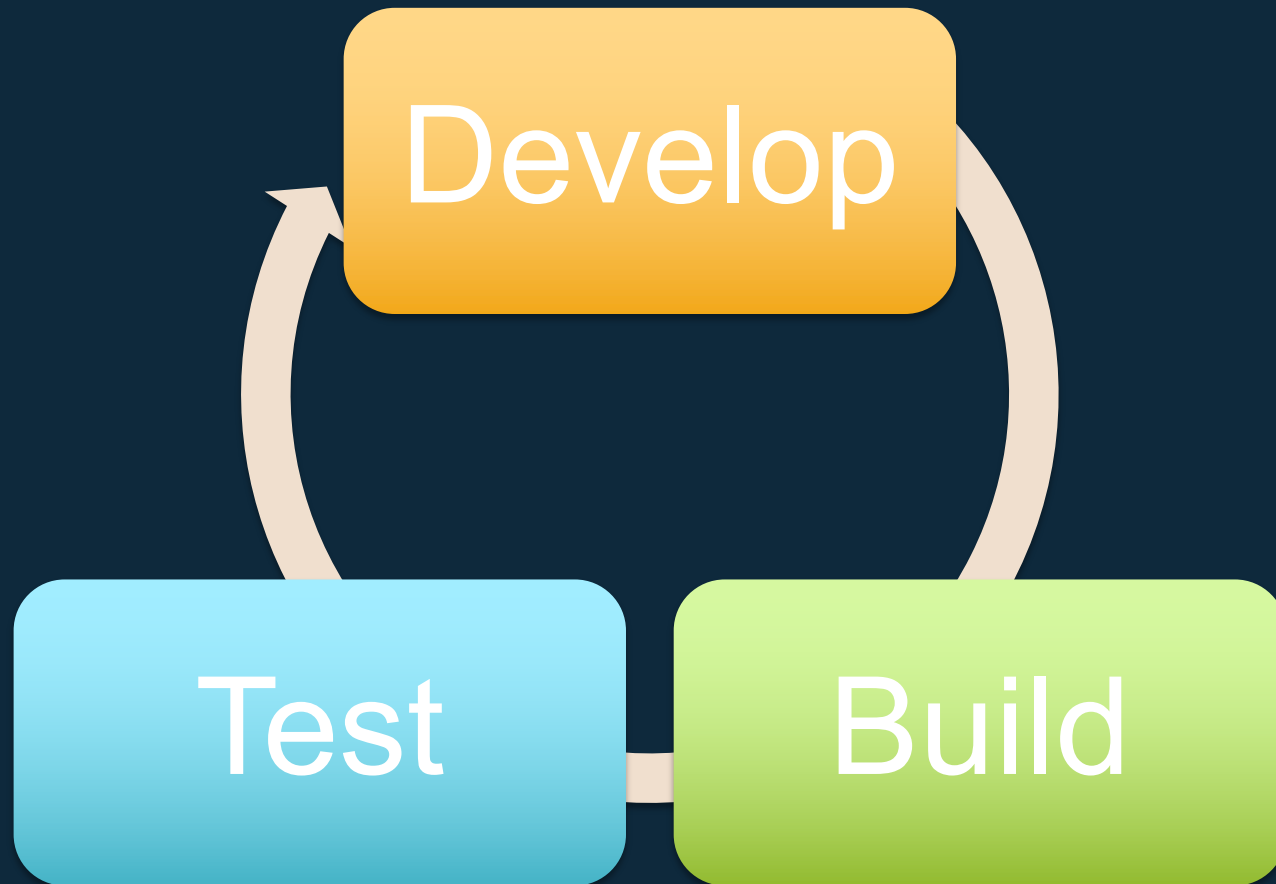
@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



Some commits should fail!







Quality is the bottleneck

Automatic testing is not enough

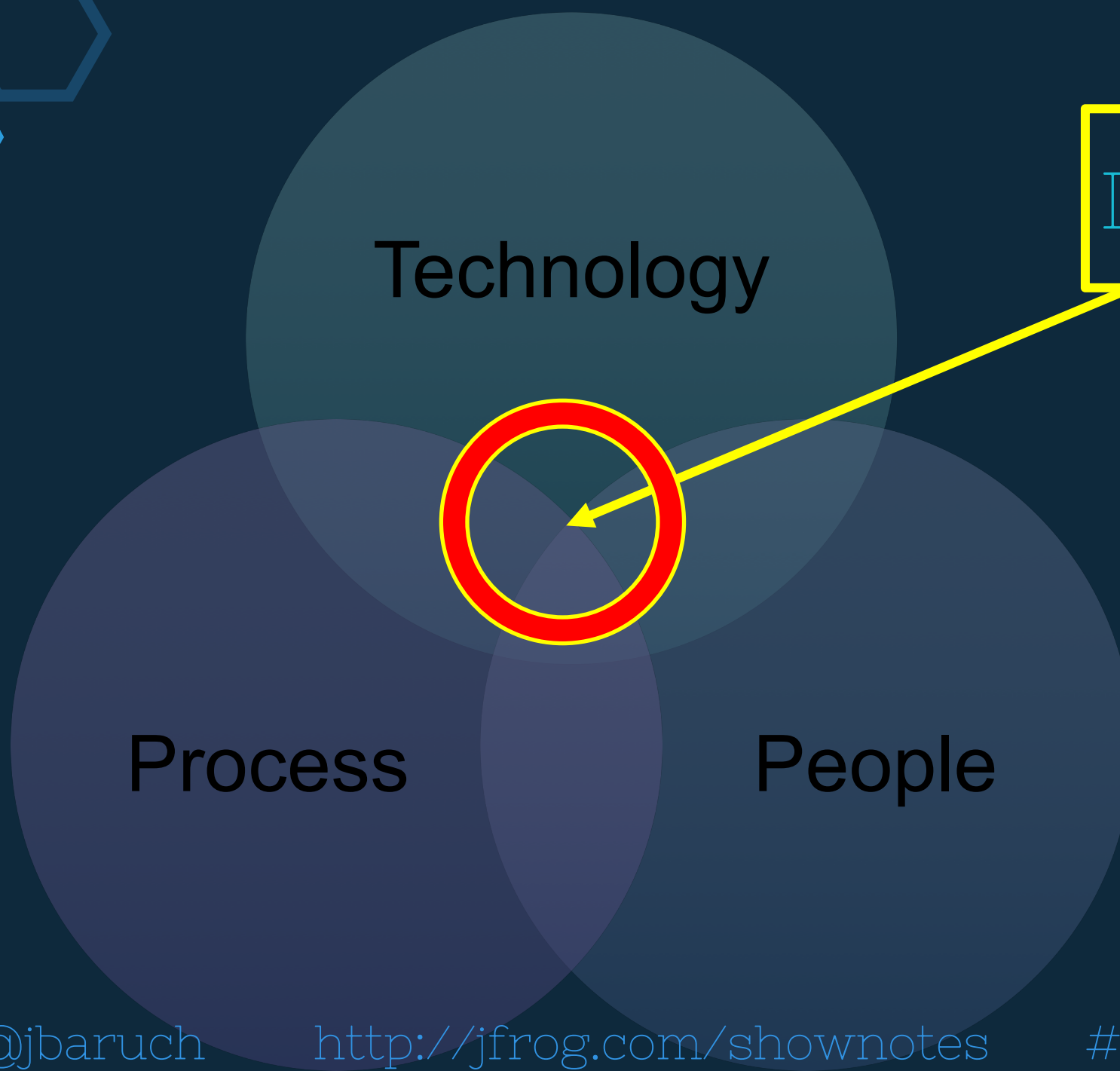
The “big green button” concept is lack of trust in quality



Quality is the bottleneck

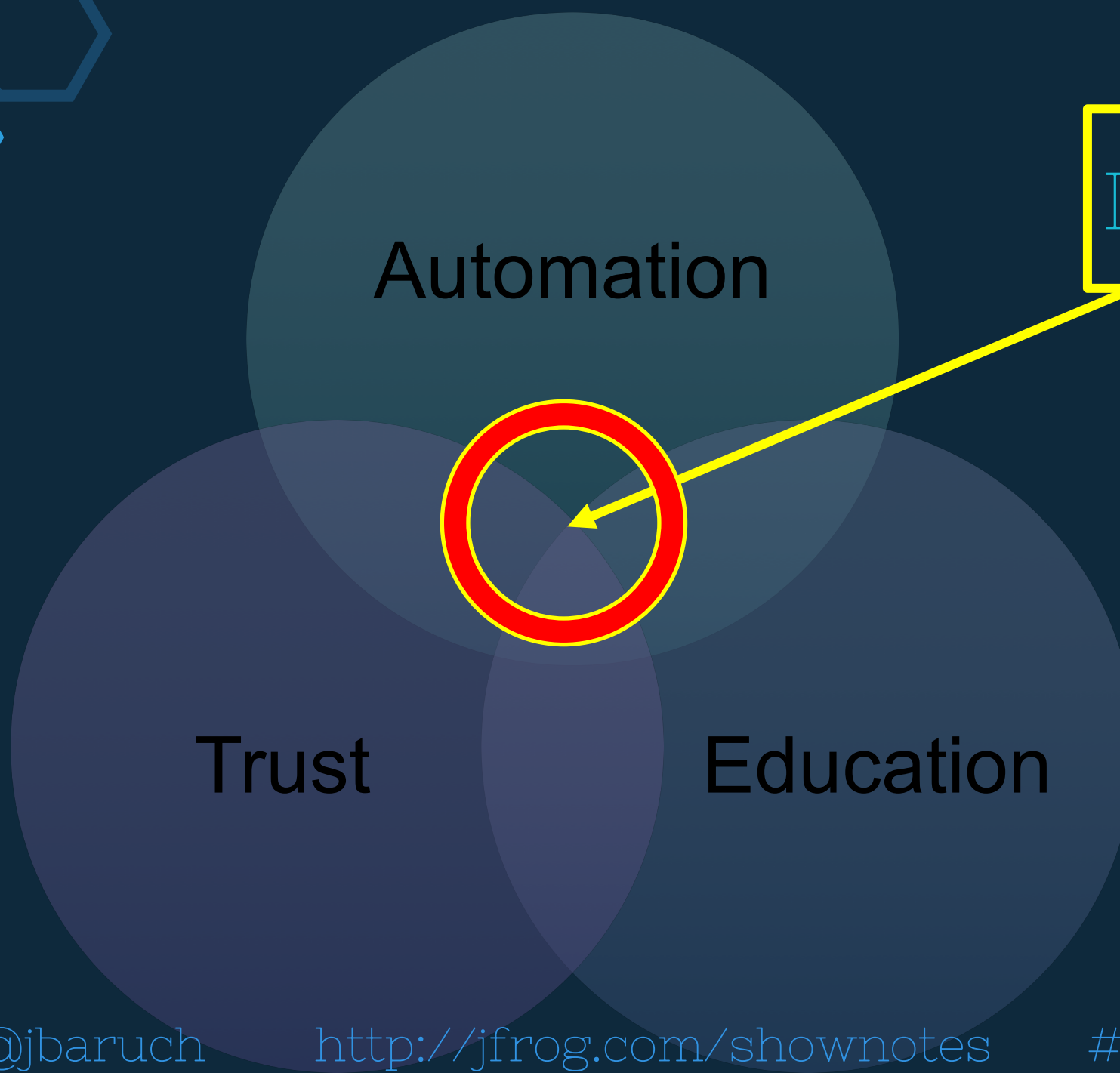
Automatic testing is not enough, but  
we can use our customers as testers

Canary releases



Devops!





Devops!

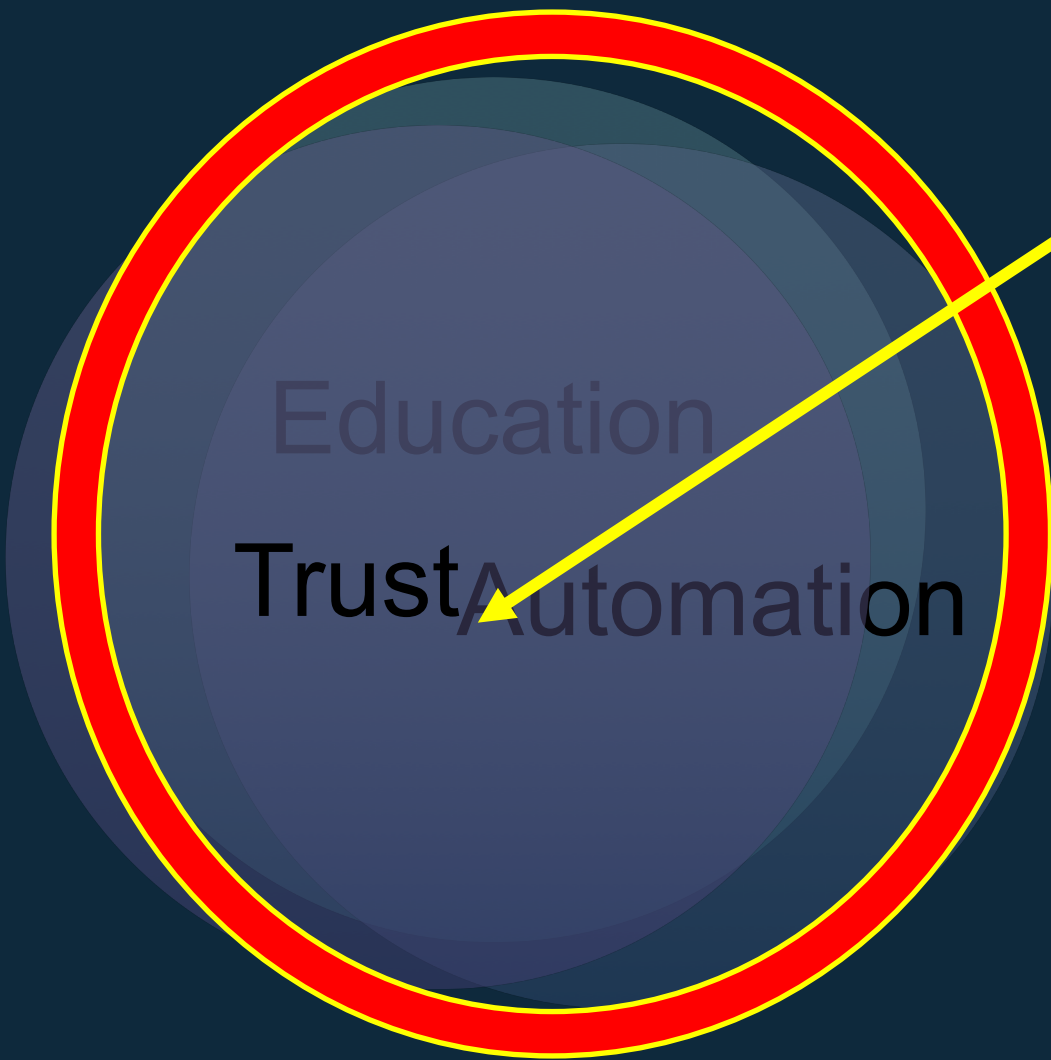




@jbaruch

<http://jfrog.com/shownotes>

#DevNexus



Devops!





# QA & Show notes

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

