SESSION NUMBER ARIAL (TBD)

What Observing 30,000 Development Teams Revealed About the Future of Machines Making Software

Derek E. Weeks

Vice President, Sonatype
Co-founder, All Day DevOps

@weekstweets

2014: Derek Weeks

Everyone has a software supply chain.

faster is better
for the enterprise

2019: Nicole Forsgren

ELITE PERFORMERS

Comparing the elite group against the low performers, we find that elite performers have…

208
TIMES MORE
frequent code deployments

106
TIMES FASTER
lead time from commit to deploy

2,604
TIMES FASTER
time to recover from incidents

7
TIMES LOWER
change failure rate
(changes are 1/7 as likely to fail)

Throughput Stability

Source: Accelerate: State of DevOps 2019

2017: Jeff Bezos

without adversaries,
there would be
no need for security

faster is better for adversaries

Adversarial Tactic: Wait and Prey

March 7
Apache Struts releases updated version to thwart vulnerability CVE-2017-5638

March 8
NSA reveals Pentagon servers scanned by nation-states for vulnerable Struts instances

Struts exploit published to Exploit-DB.

March 9
Cisco observes "a high number of exploitation events."

March 10
- Equifax
- Canada Revenue Agency
- Canada Statistics
- GMO Payment Gateway

March 13
Okinawa Power
Japan Post

April 13
India Post

March '18
India's AADHAAR

3 Days in March

The Rest of the Story

December '17
Monero Crypto Mining

Today
65% of the Fortune 100 download vulnerable versions

sonatype

@weekstweets

27 open source breaches in May

MARCH 12
Vulnerability found in SaltStack open source configuration framework, available as a PyPI package.

According to Flexera, Salt is used by around 17 percent of organizations with cloud deployments.

APRIL 15
F-secure informs SaltStack of 6,000 publicly exposed Salt Masters at risk of compromise.

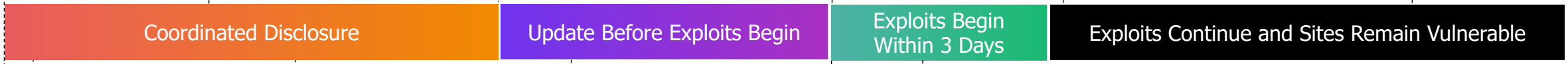
APRIL 29
SaltStack publishes version 3000.2 and 2019.2.4 to fix issue and shares identifiers: CVE-2020-11651 and CVE-2020-11652.

F-Secure: "We expect that any competent hacker will be able to create 100% reliable exploits for these issues in under 24 hours."

MAY 2
LineageOS, a maker of an open source operating system based on Android, said it detected the intrusion on May 2nd at around 8 pm Pacific Time.

MAY 7
Cisco discovered the compromise of six of their Salt master servers, which are part of the Cisco VIRL-PE (Internet Routing Lab Personal Edition) service infrastructure.

MAY 12
Censys reports the number stands at **2,928** Salt servers still exposed — a 21% reduction from last week, and a **50% reduction overall** since the CVE was announced.



MARCH 24
SaltStack confirms receipt of vulnerability report.

APRIL 23
SaltStack publishes advance notice to their users urging them not to expose Salt Masters to the internet and prepare to apply patch on April 29th.

APRIL 30
Sonatype ingests the CVE information.

MAY 2

18 breaches noted on GitHub accounts

- **xiaopanggege**: an unknown program suddenly ran today
- **atuchak**: I have the same
- **nepetadosmil**: gents, this is an attack. We've had all firewalls disabled
- **aidanstevens29**: a backdoor was also installed via the exploit
- **ndmgrphc**: entire system is being taken down
- **nebev**: been affected :(
- **venugopalnaidu**: we got the same issue
- **gorgeousJ**: same thing in my servers
- **atastycookie**: we are investigating
- **avasz**: It also stopped and disabled docker services
- **aldenar**: looking through my affected machines, a dropper scriptfile was found
- **foobartender**: it also adds a key to /root/.ssh/authorized_keys
- **bruxy**: same issue here
- **mcpcholkin**: I found it only on one server
- **wavded**: we had one job that was executed that did the following on each server
- **justinimn**: I got hit a few hours ago
- **curu**: Firewall rules stopped and disabled

MAY 3

DigiCert reported that one of its Certificate Transparency logs was affected after attackers used the Salt exploits.

Ghost, a node.js blogging platform, reports an attacker used a CVE in our SaltStack master to gain access to our infrastructure and install a cryptocurrency miner.

Xen-Orchestra *reports coin mining script ran on some of their VMs tied to SaltStack vulnerability.*

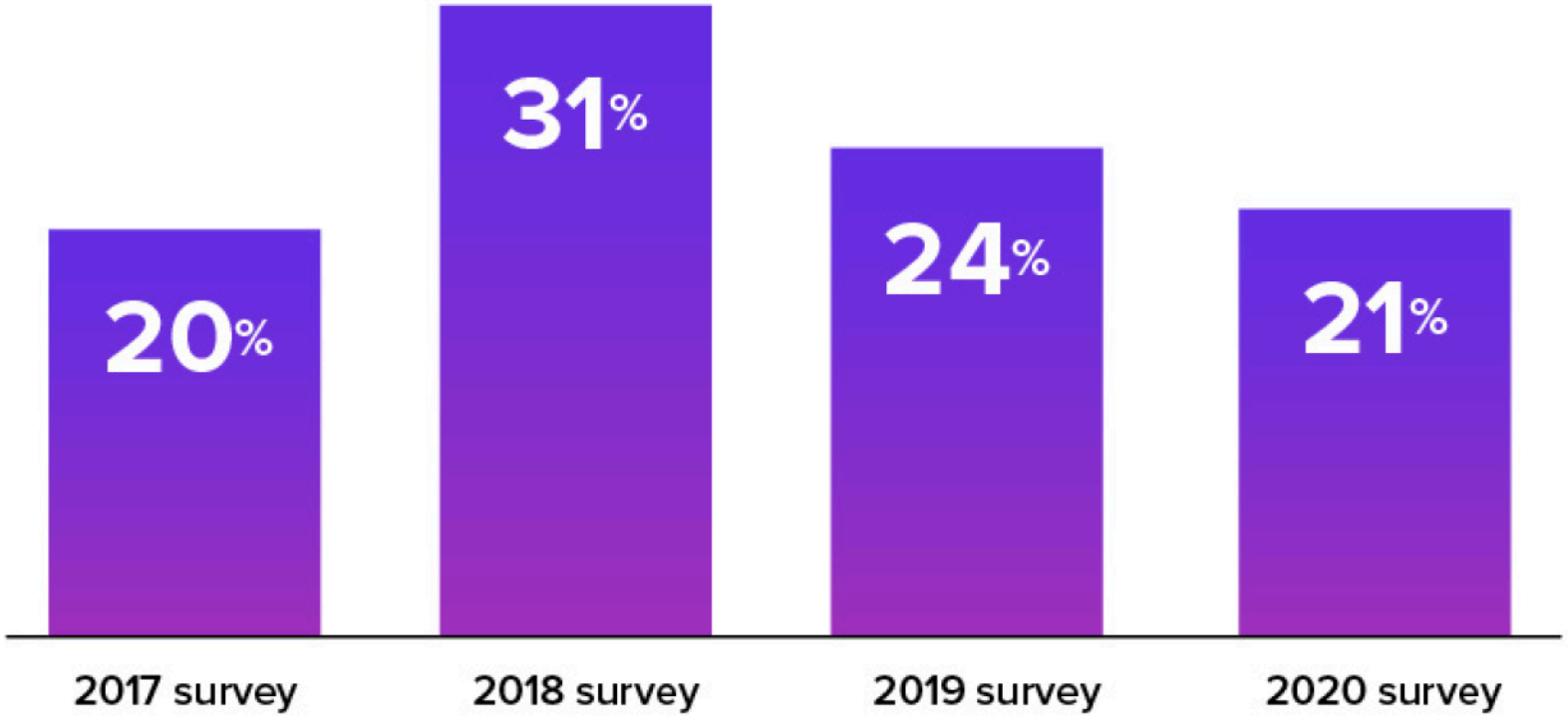
Algolia reports hackers installed a backdoor and a cryptocurrency miner on a small number of its servers.

3 breaches noted on GitHub

- **jblac**: it's the same issue I was plagued with
- **heruan**: minor jobs are still spawning on minions
- **leeyo**: we have the same problem

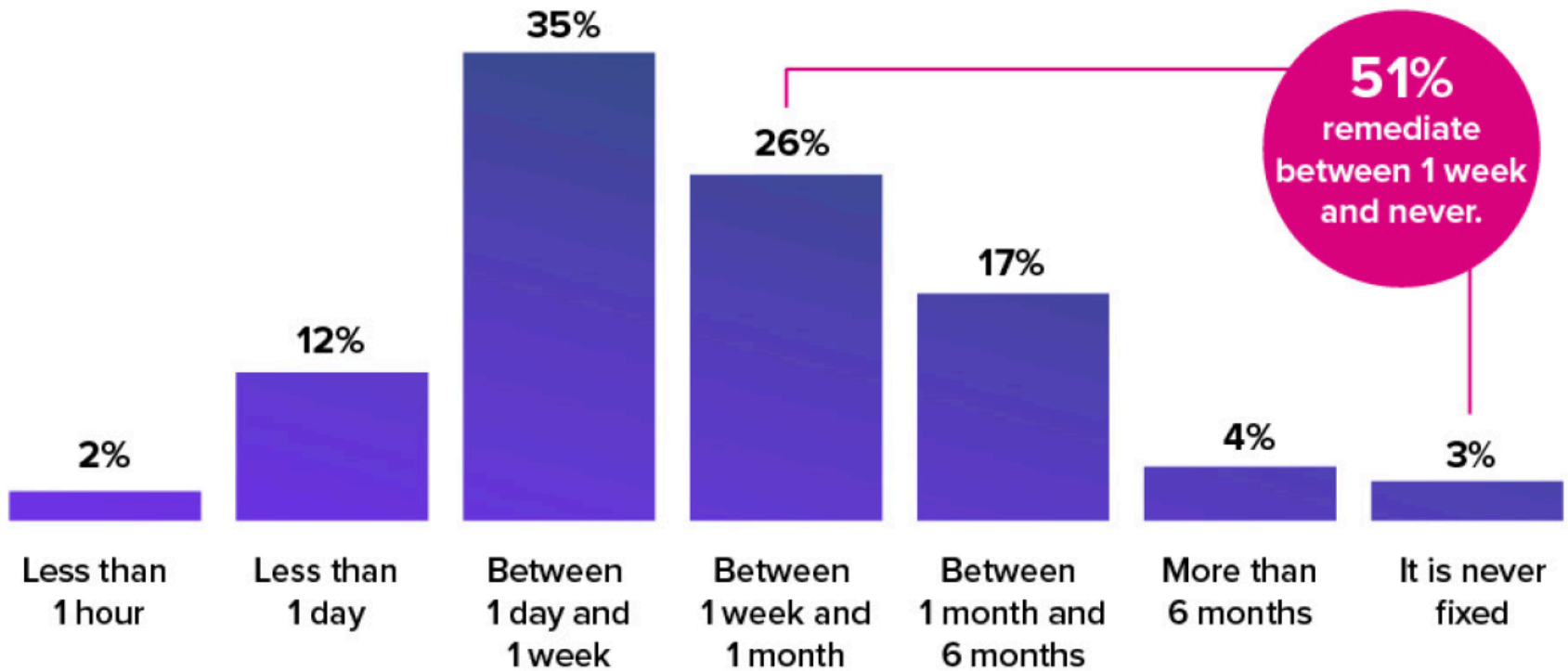
@weekstweets

“Yes, we’ve had an OSS related breach.”

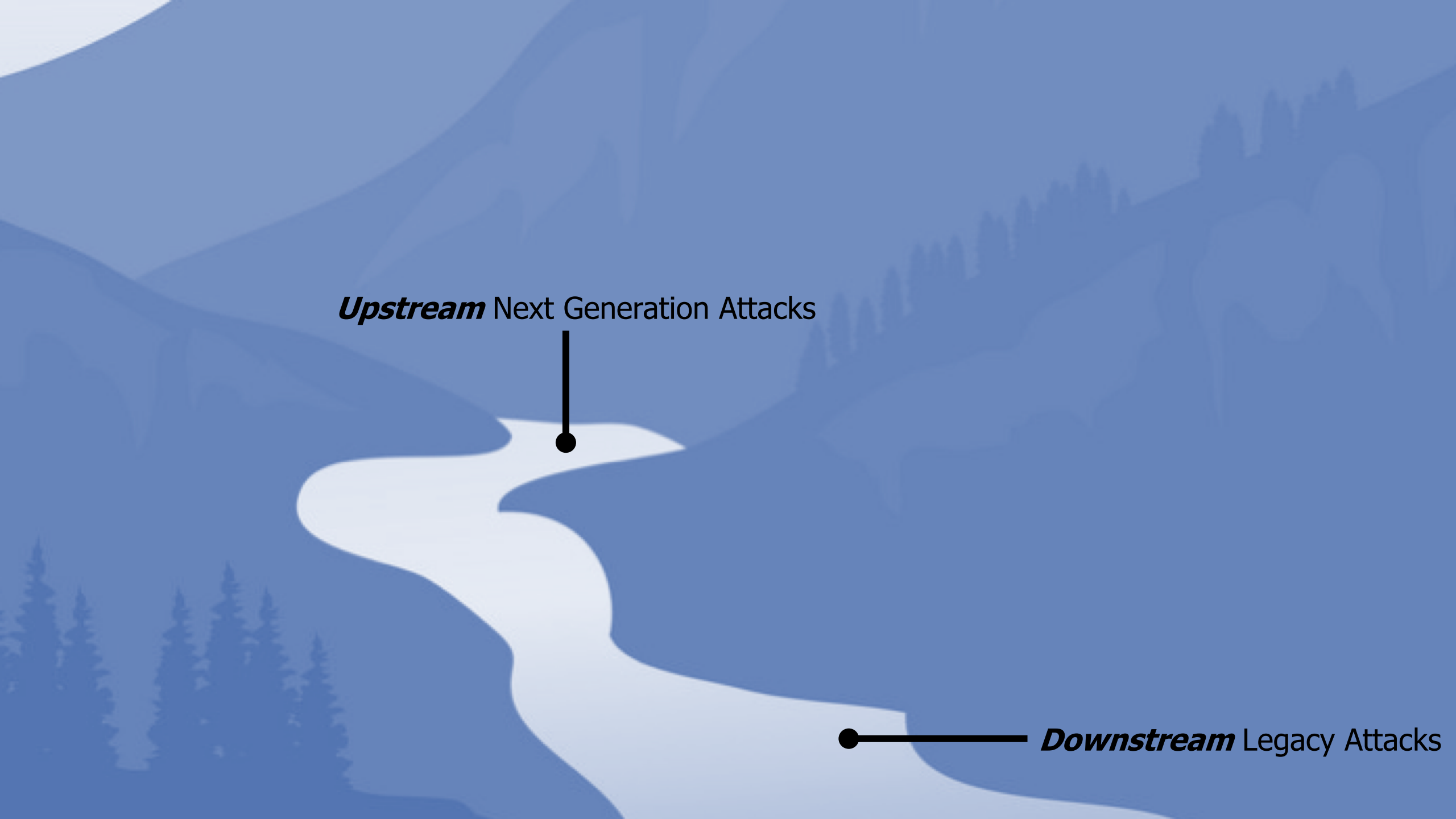


Source: DevSecOps Community Survey 2017 – 2020
Have you had an open source related breach in the past 12 months?

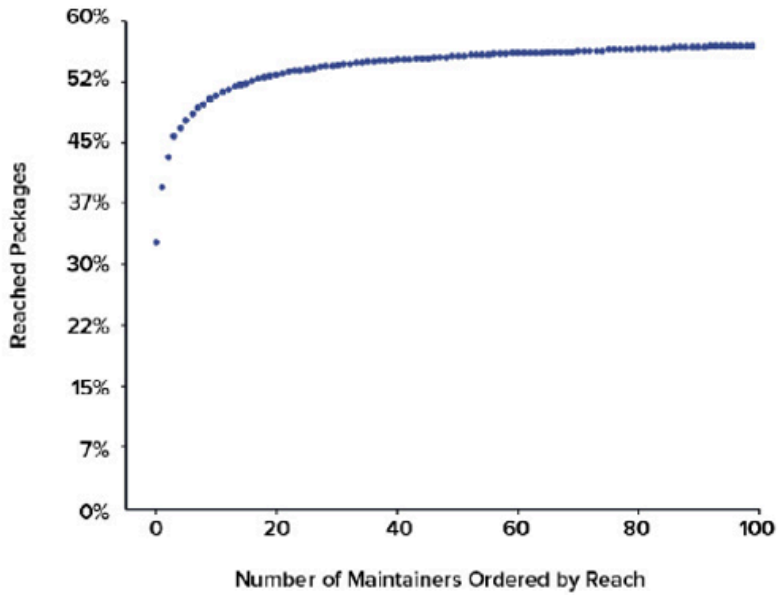
Time to Remediate Known OSS Vulnerabilities After Detection



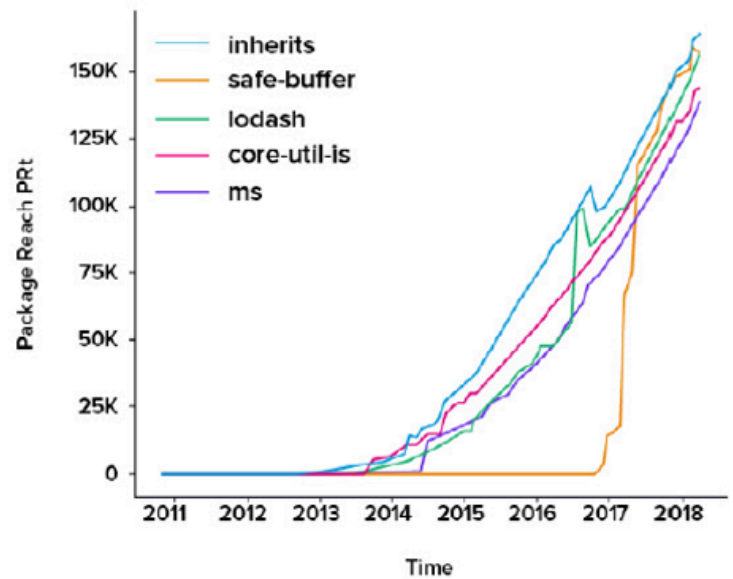
adversaries seek the most efficient path


Upstream Next Generation Attacks
Downstream Legacy Attacks

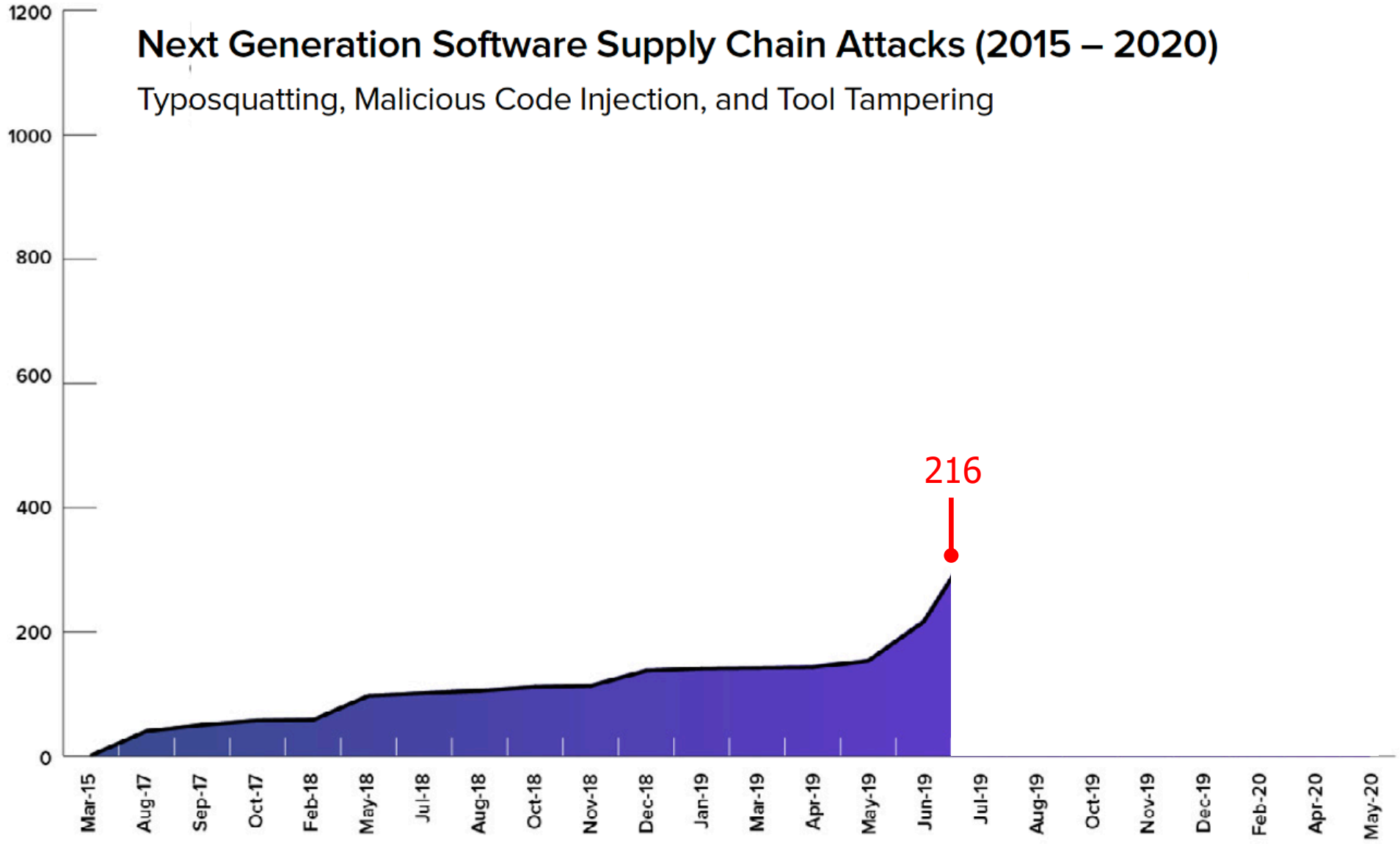
Combined Reach of 100 Influential Maintainers

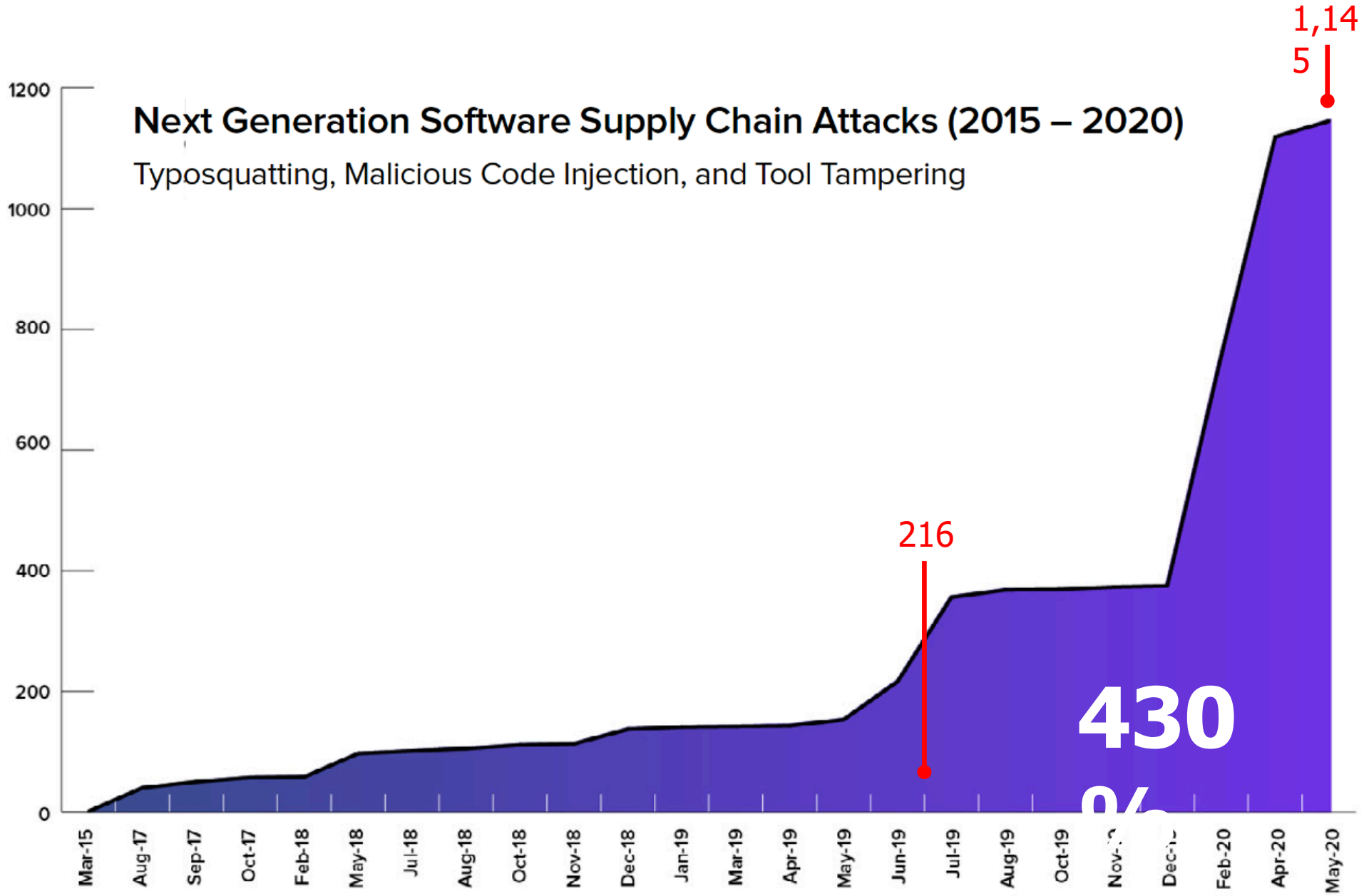


Evolution of Package Reach for the Top 5 npm Packages



SOURCE: MARKUS ZIMMERMANN AND CRISTIAN-ALEXANDRU STAICU, TU DARMSTADT; CAM TENNY, R2C; MICHAEL PRADEL, TU DARMSTADT


Next Generation Software Supply Chain Attacks (2015 – 2020)
Typosquatting, Malicious Code Injection, and Tool Tampering
1200
1000
800
600
400
200
0
216
Mar-15
Aug-17
Sep-17
Oct-17
Feb-18
May-18
Jul-18
Aug-18
Oct-18
Nov-18
Dec-18
Jan-19
Mar-19
Apr-19
May-19
Jun-19
Jul-19
Aug-19
Oct-19
Nov-19
Dec-19
Feb-20
Apr-20
May-20


Next Generation Software Supply Chain Attacks (2015 – 2020)
Typosquatting, Malicious Code Injection, and Tool Tampering
1,145
216
430
1200
1000
800
600
400
200
0
Mar-15
Aug-17
Sep-17
Oct-17
Feb-18
May-18
Jul-18
Aug-18
Oct-18
Nov-18
Dec-18
Jan-19
Mar-19
Apr-19
May-19
Jun-19
Jul-19
Aug-19
Oct-19
Feb-20
Apr-20
May-20

March 2018

steal credentials

npm credentials intentionally compromised.

A malicious version of a package from a core contributor to the conventional-changelog ecosystem is published. The package was installed 28,000 times in 35 hours and executed a Monero crypto miner.

June 2019

steal passwords

23 RubyGems packages pulled from the public repository.

Packages including chrome_taker, color_hacker, aloha_analyser, get-text, ruby_nmap, get-texts, colourize, colourful, TacoBell, unix_crypt, colour-lib, colour_lib, json_colour, unixCrypt, auto-cron, json-colour, Copyip, colour_cat, colour-generator, phantom-proxy, colour_adjuster, colour_parser, and btc-ruby were pulled from the public repository because they contained code for crypto mining or cookie/password stealing.

June 2019

steal money

Code for cryptocurrency theft identified in npm package.

electron-native-notify (version 1.1.6) contains code designed to steal cryptocurrency wallet seeds and other login instruction details specific to cryptocurrency apps. Tipped off by npm researches, makers of the Agama cryptocurrency wallets shifted $13 million worth of currency before adversaries could steal it.

July 2019

backdoored

Libpeshnx Researchers at ReversingLabs identified a PyPI package with back-door vulnerability.

While the package had been reported as containing a known vulnerability, it had not been removed from the Python package repository — as is often the case with intentionally malicious packages.

May 2020

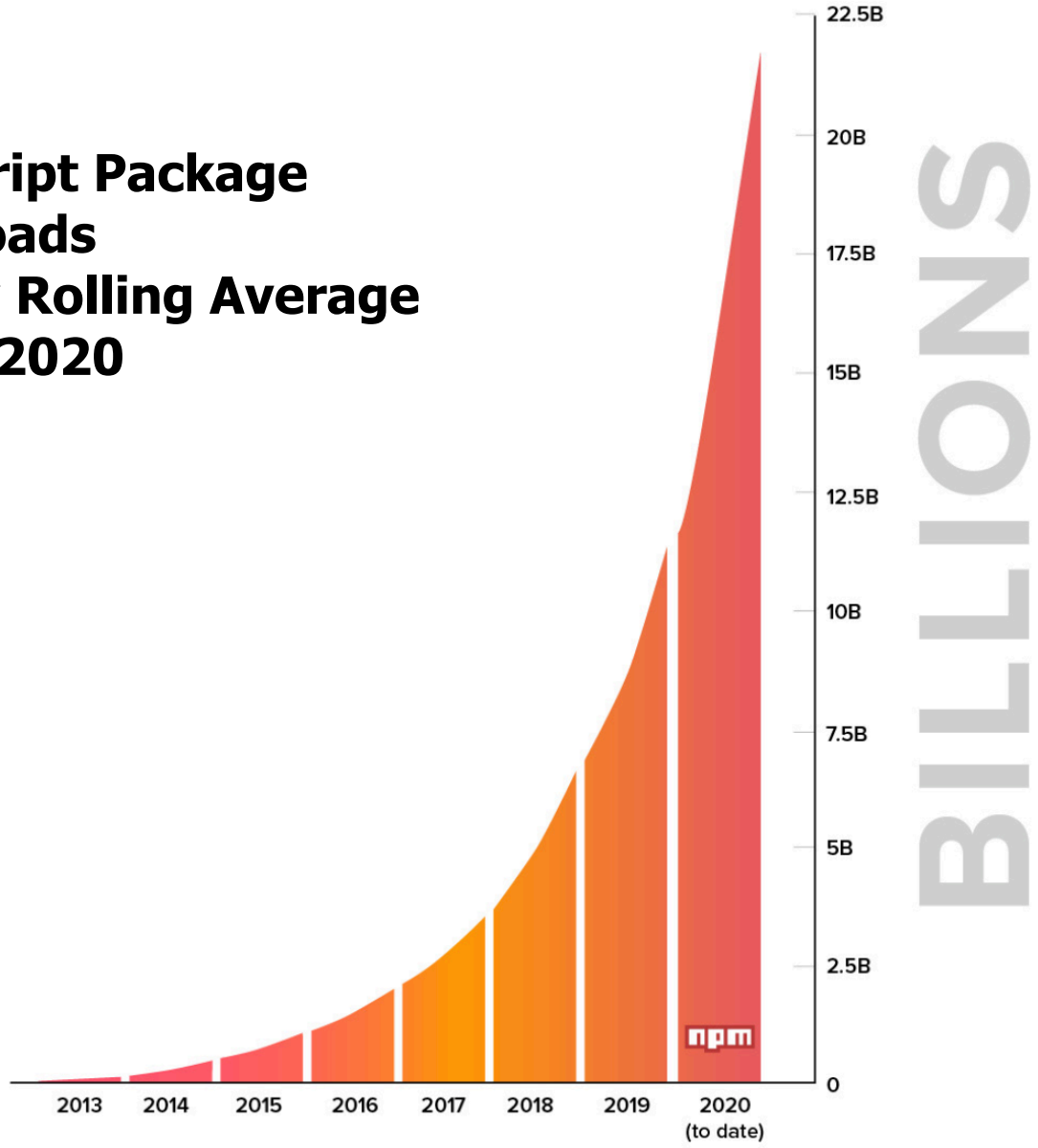
tool tampering

Octopus Scanner

26 open source packages were found to be compromised through malicious code injection. The malware was designed to enumerate and backdoor NetBeans projects through the NetBeans IDE.

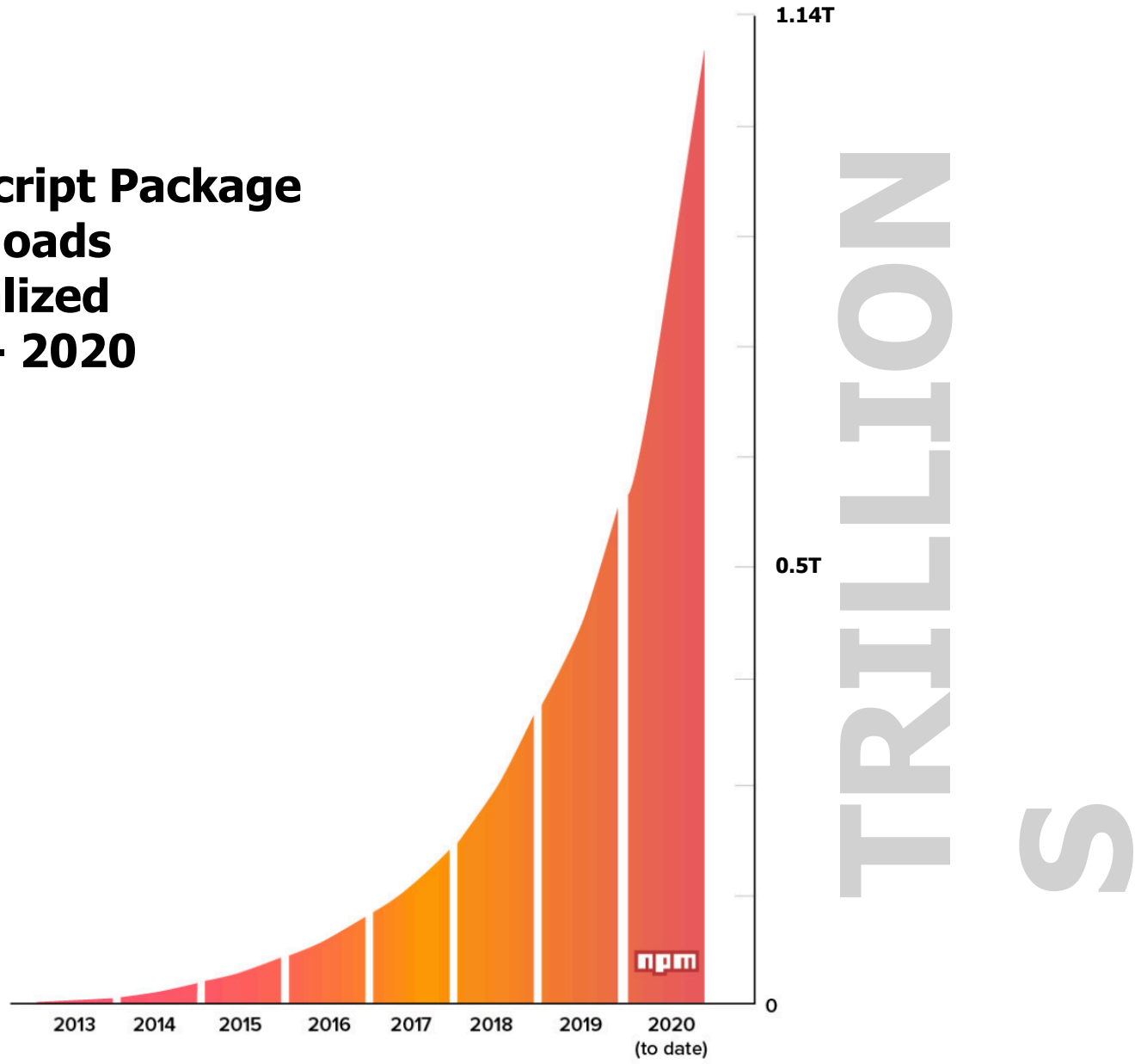
developers are getting faster

JavaScript Package Downloads
Weekly Rolling Average
2013 - 2020



SOURCE: Microsoft, npm inc.

JavaScript Package Downloads Annualized 2013 - 2020



SOURCE: Microsoft, npm inc.

OSS download volumes are a proxy for build automation.

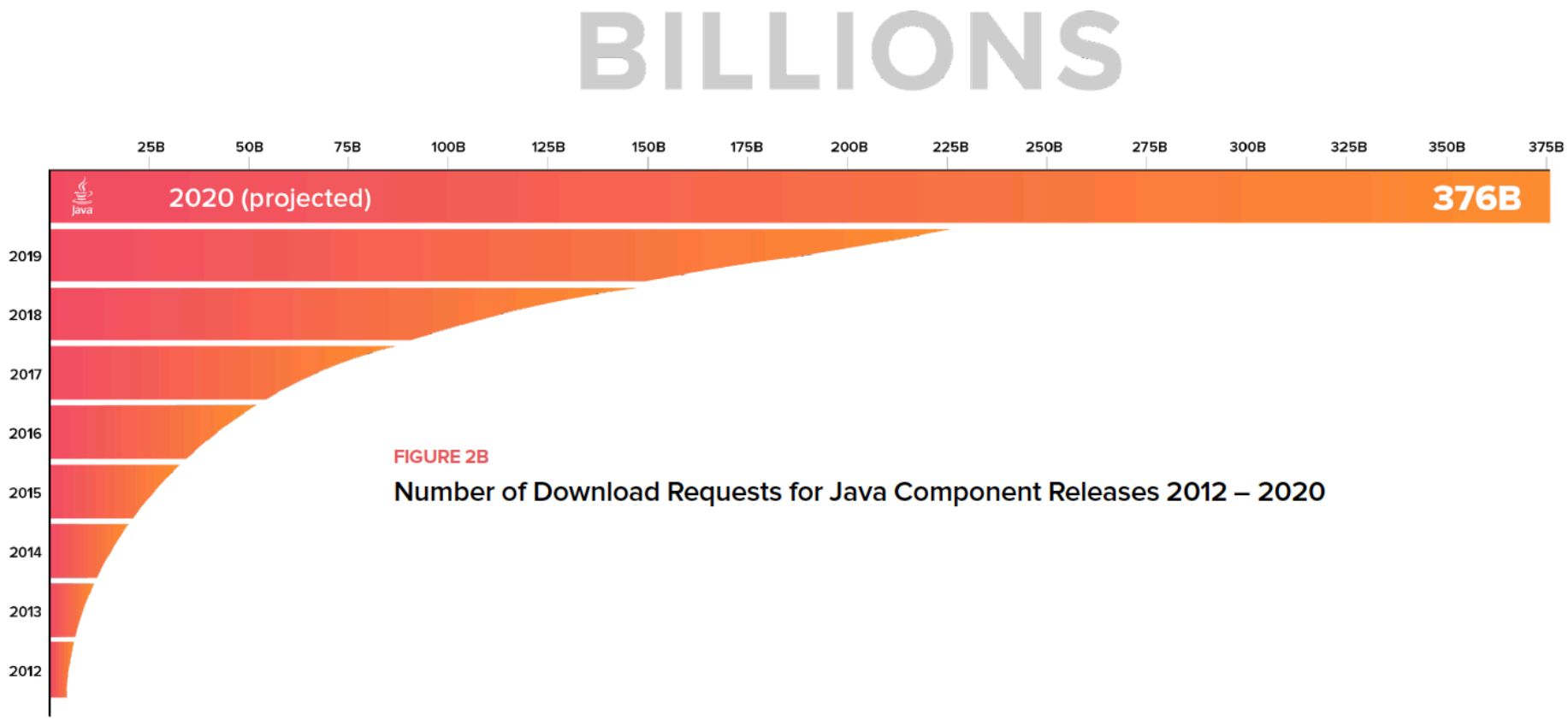


FIGURE 2B

Number of Download Requests for Java Component Releases 2012 – 2020

sonatype

@weekstweets

2015: John Willis

is faster is better for open source?

What does High Performance mean?

Enterprise	Open Source
Deployment Frequency	Release Frequency
Organizational Performance	Popularity
Mean Time to Restore	Time to Remediate Vulnerabilities

Our "Interview Process" for 24,000 OSS Projects

Attributes	Measure
Popularity	Avg. daily Central Repository downloads
Size of Team	Avg. unique monthly contributors
Development Speed	Avg. commits per month
Release Speed	Avg. period between releases
Presence of CI	Presence of popular cloud CI systems
Foundation Support	Associated with an open source foundation
Security	More complicated
Update Speed	More complicated

HYPOTHESIS 1

Projects that release frequently have better outcomes.

are 2.5x more popular.

1.4x larger development teams

have 12% greater foundation support rates

(VALIDATED)

HYPOTHESIS 2

Projects that update dependencies more frequently

are generally more secure.

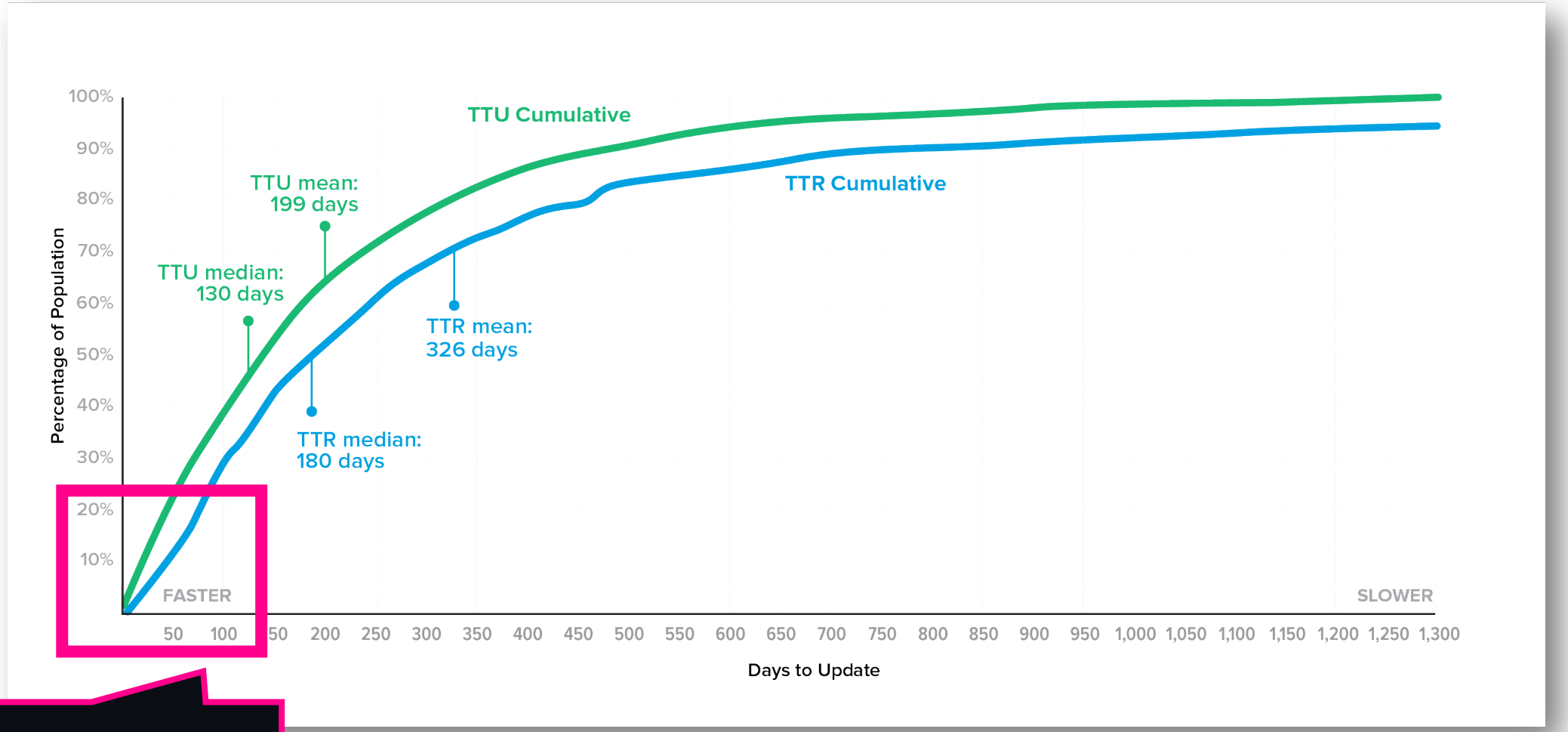
1.5x more frequent releases

530x faster median time to update

173x less likely to have out of data dependencies

(VALIDATED)

Time to Remediate (TRR) vs. Time to Update (TTU)



Most projects stay secure by staying up to date.

@weekstweets

HYPOTHESIS 3

Projects with fewer dependencies will stay more up to date.

(REJECTED)

*Components with more dependencies actually have **better** MTTU.*

HYPOTHESIS 4

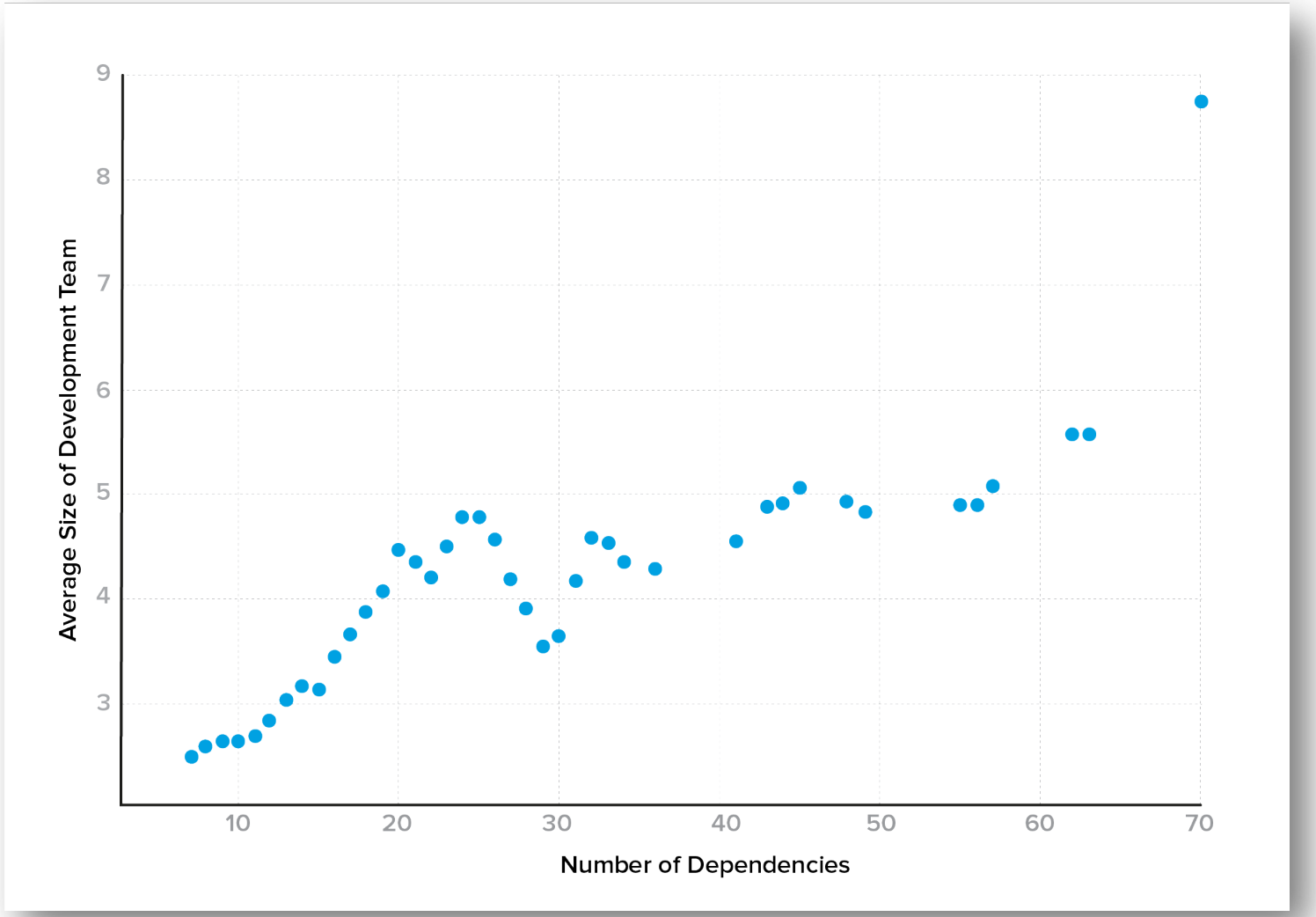
More popular projects will be better about staying up to date.

(REJECTED)

There are plenty of popular components with poor MTTU. Popularity does not correlate with MTTU.

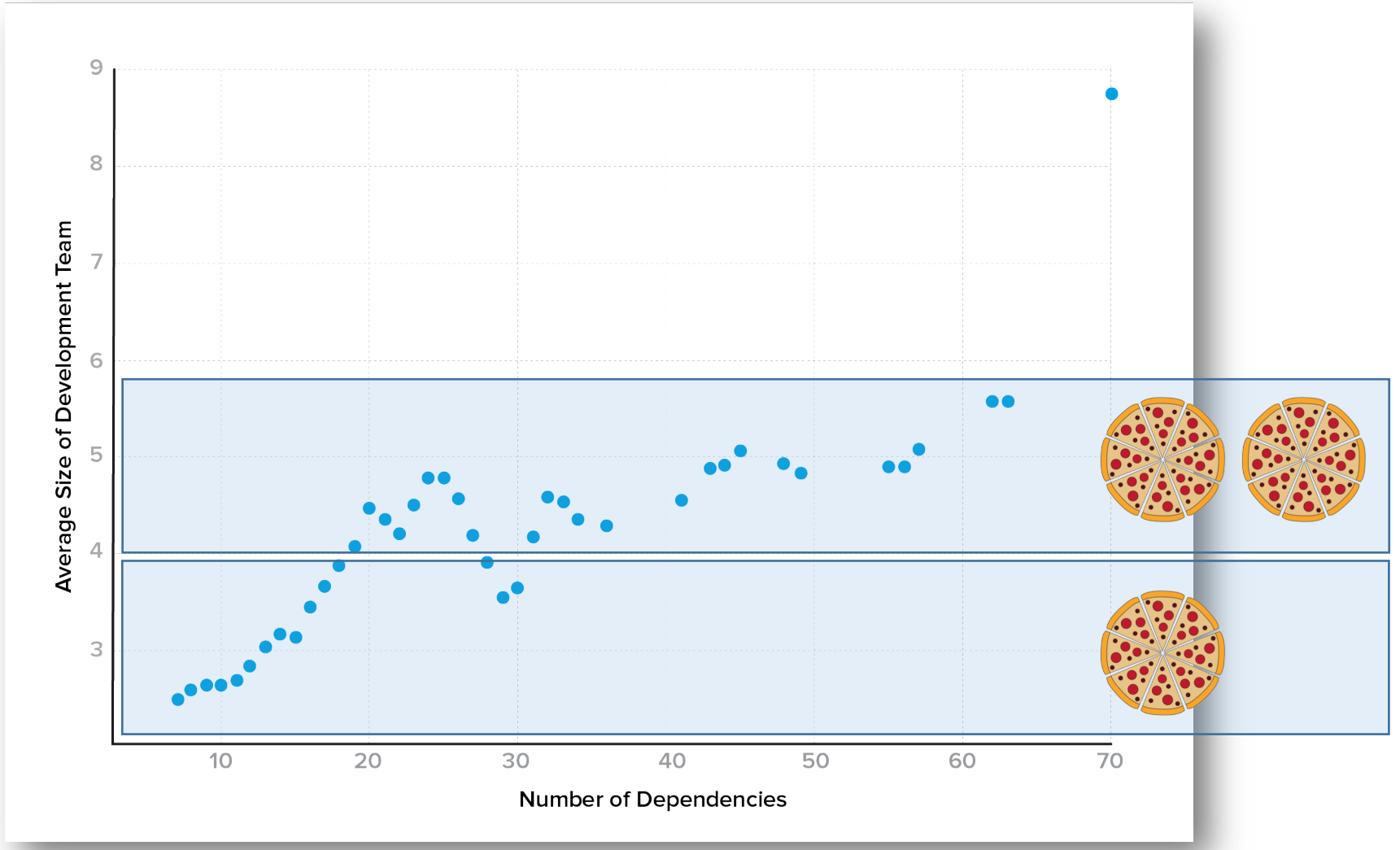
More dependencies correlate with larger development teams.

Larger development teams have 50% faster MTTU and release 2.6x more frequently.



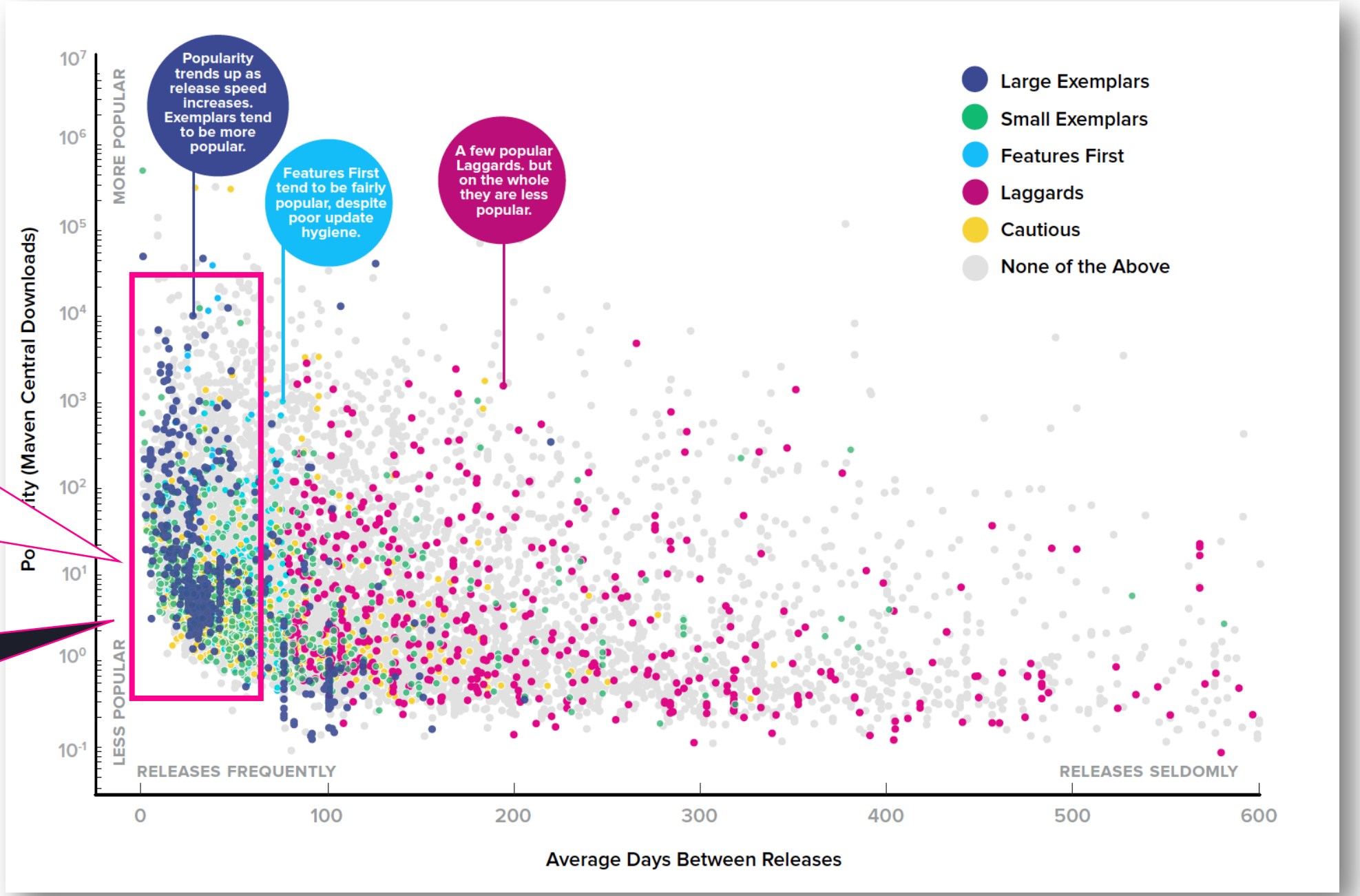
More dependencies correlate with larger development teams.

Larger development teams have 50% faster MTTU and release 2.6x more frequently.



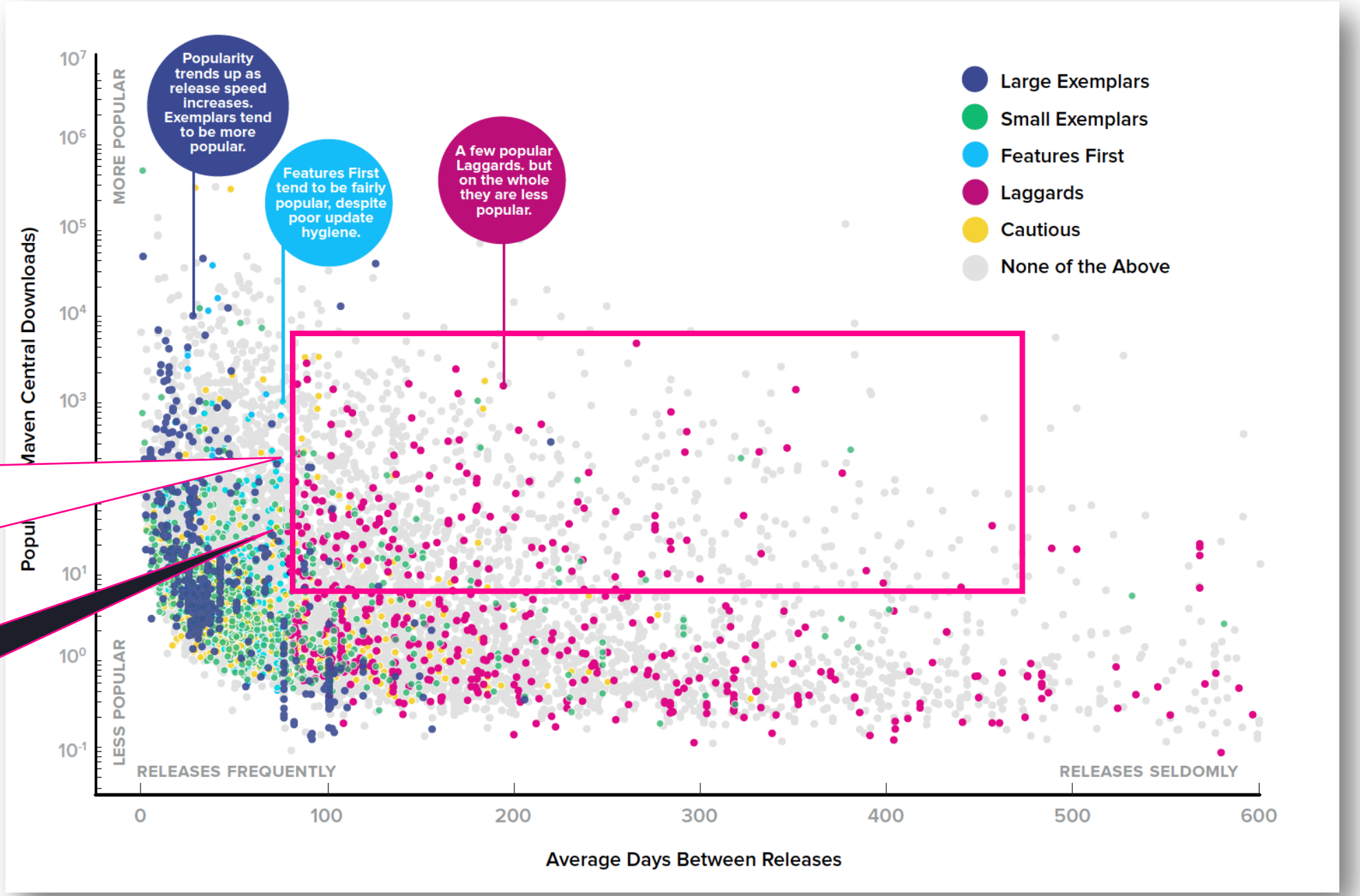
Exemplars release fast and tend to be more popular.

Pick suppliers from here.



Not all popular projects are exemplary and release fast

Avoid suppliers from here.



Guidance for OSS Projects

focus on accelerating and maintaining rapid MTTU (for users too)

projects commit resources to dependency management

when adding a new dependency look for a metric to guide that choice

aim for a minimum of four releases annually

aim to upgrade at least 80% of dependencies with every release

Guidance for Enterprise Development

choosing OSS projects should be a strategic decision

implement selection criteria

MTTU should be an important metric

formalize a procurement process that works at the speed of dev

minimize variability by relying on the fewest and best suppliers

…faster is better in the enterprise

1.75x more likely to make extensive use of OSS components

1.5x more likely to be expanding use of OSS components

Source: Accelerate: State of DevOps 2019

DORA
DEVOPS RESEARCH & ASSESSMENT

373,000
java component downloads annually

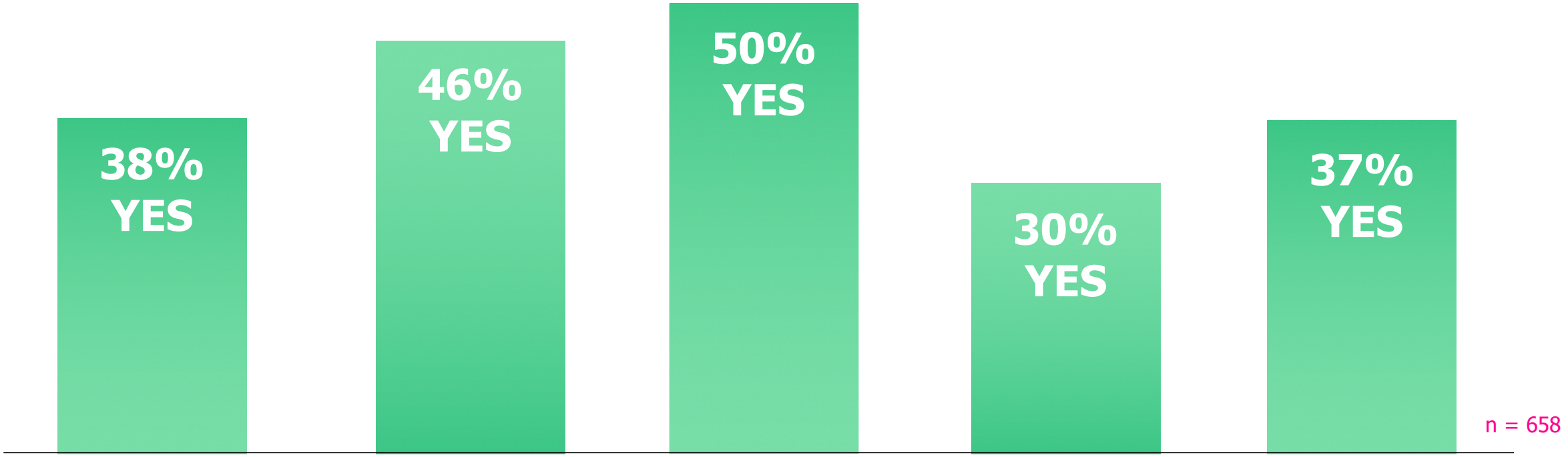
3,552
Component suppliers

11,294
Component release

@weekstweets

…are faster and more secure achievable in the enterprise?

Enterprise Devs Manage Dependencies



We schedule updating dependencies as part of our daily work

We strive to use the latest version (or latest-N) of all our dependencies

We use some process to add a new dependency (e.g., evaluate, approve, standardize, etc.)

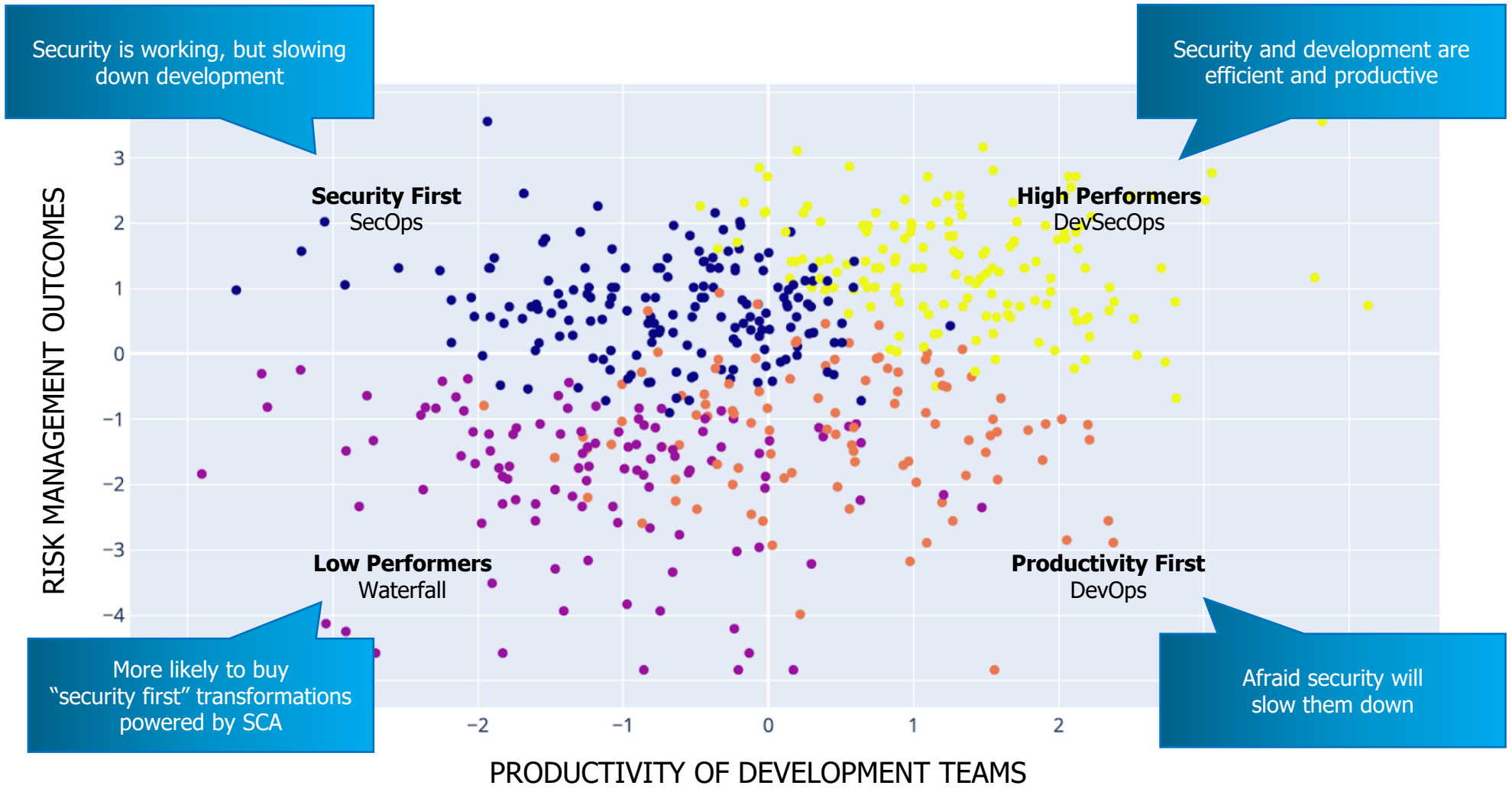
We have a process to proactively remove problematic or unused dependencies

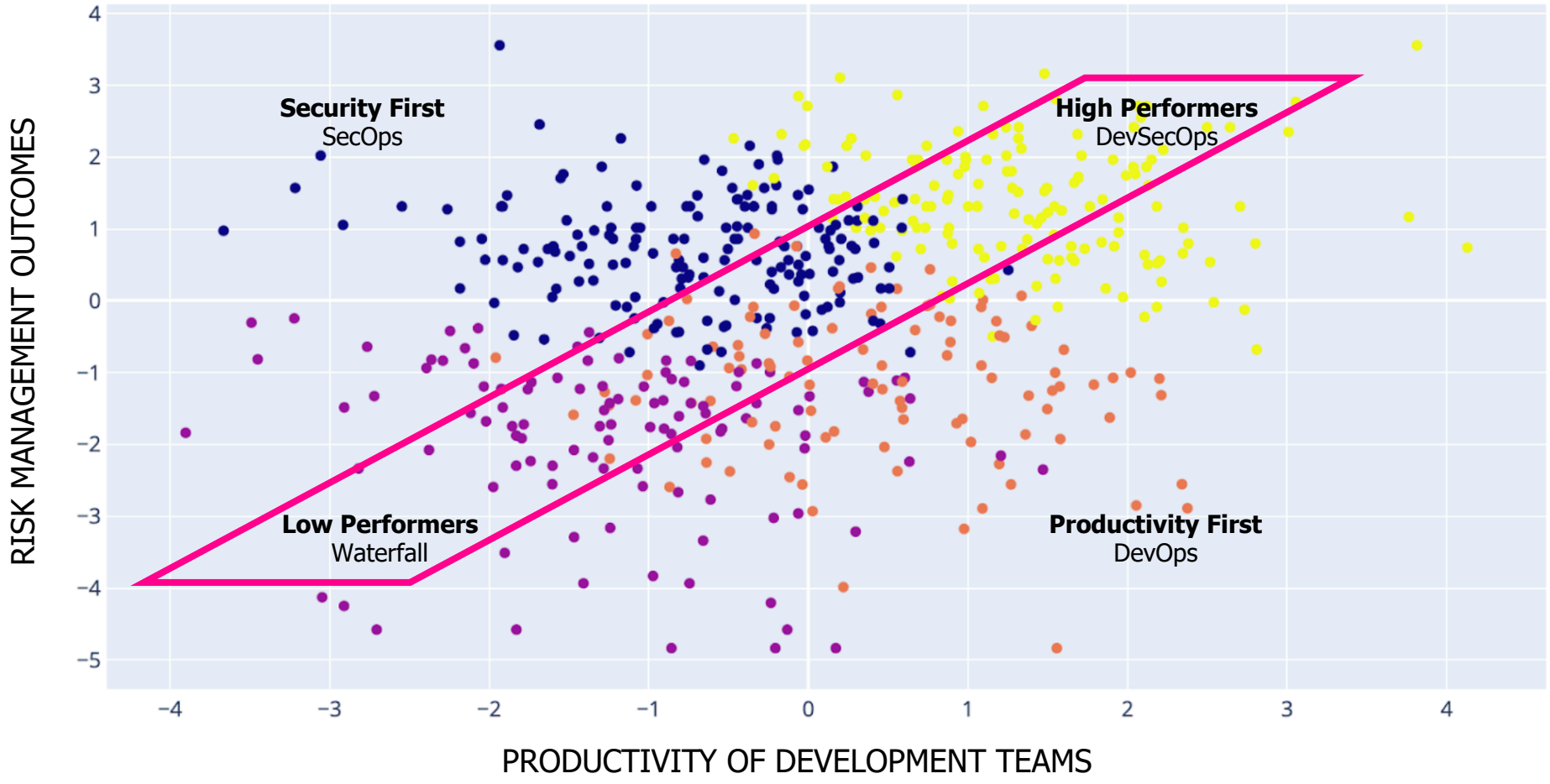
We have automated tools to track, manage, and/or ensure policy compliance of our dependencies

@weekstweets

679 enterprises

Practices	Factors
Development	Development philosophy
	Deployment automation and frequency
Build, Test, Release	Confidence in automated testing
	Scheduled dependency updates
	Scheduled patching
	Static analysis tools
	Artifact repository centralization
OSS Suppliers	OSS selection criteria
OSS Philosophy	Process to add OSS components
	Process to remove OSS components
	OSS enlightenment
Organization and Policy	Centralization of asset management
	Centralized OSS governance
	OSS enforcement via automated CI
	OSS governance enforcement


Security is working, but slowing down development
Security and development are efficient and productive
Security First
SecOps
High Performers
DevSecOps
Low Performers
Waterfall
Productivity First
DevOps
More likely to buy "security first" transformations powered by SCA
Afraid security will slow them down
RISK MANAGEMENT OUTCOMES
PRODUCTIVITY OF DEVELOPMENT TEAMS
3
2
1
0
−1
−2
−3
−4
−2
−1
0
1
2


RISK MANAGEMENT OUTCOMES
PRODUCTIVITY OF DEVELOPMENT TEAMS
Security First
SecOps
High Performers
DevSecOps
Low Performers
Waterfall
Productivity First
DevOps
4
3
2
1
0
−1
−2
−3
−4
−5
−4
−3
−2
−1
0
1
2
3
4

Comparing high performers against low performers

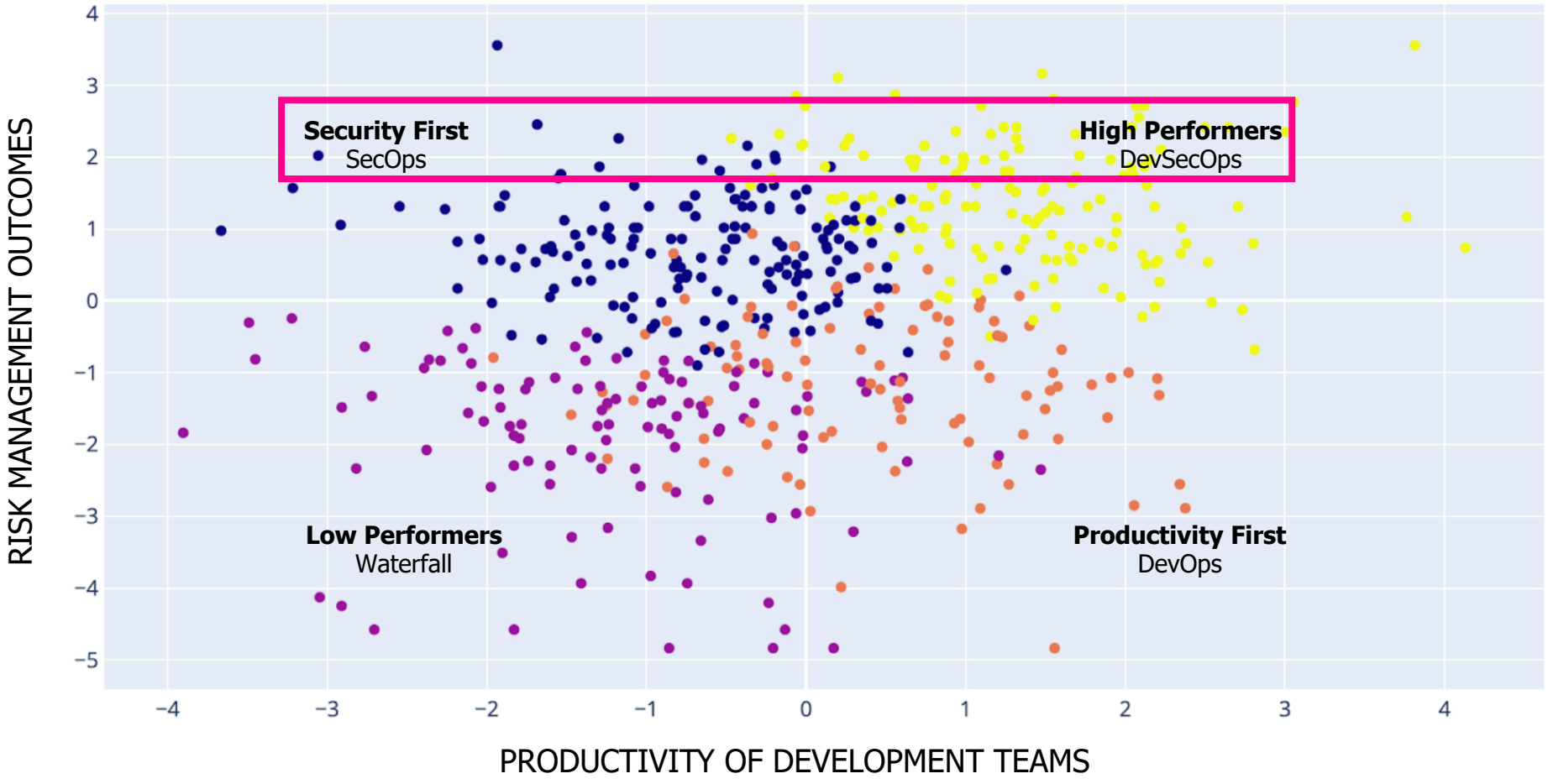
15x
more frequent deployments

26x
faster DETECTION of vulnerable OSS components

26x
faster REMEDIATION of vulnerable OSS components

5.7x
less time required for developers to be productive when SWITCHING teams

26x
less time to APPROVE a new OSS dependency for use


RISK MANAGEMENT OUTCOMES
PRODUCTIVITY OF DEVELOPMENT TEAMS
Security First
SecOps
High Performers
DevSecOps
Low Performers
Waterfall
Productivity First
DevOps
4
3
2
1
0
−1
−2
−3
−4
−5
−4
−3
−2
−1
0
1
2
3
4

Comparing high performers against security first

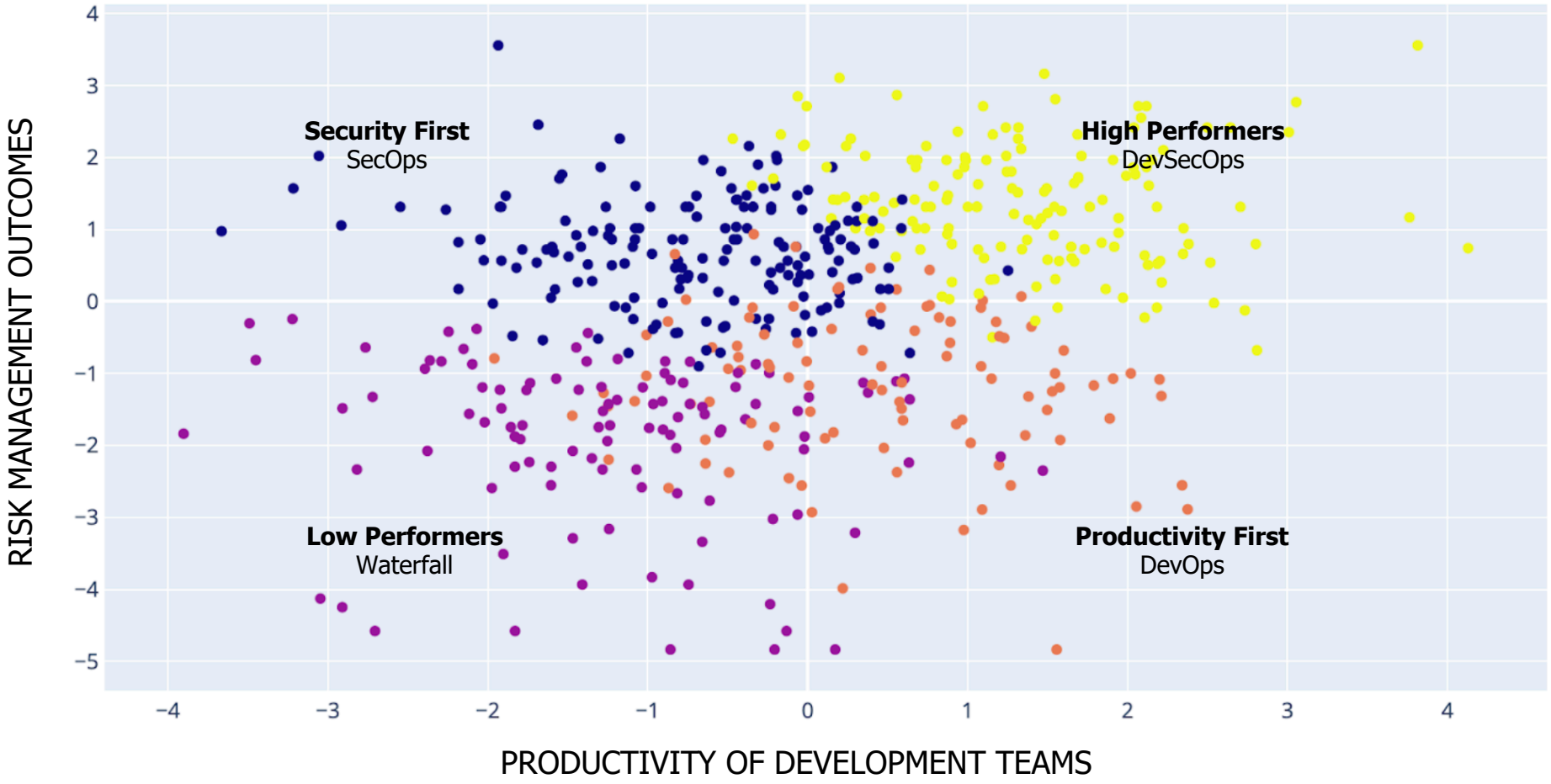
59%
more likely to be using software composition analysis (SCA)

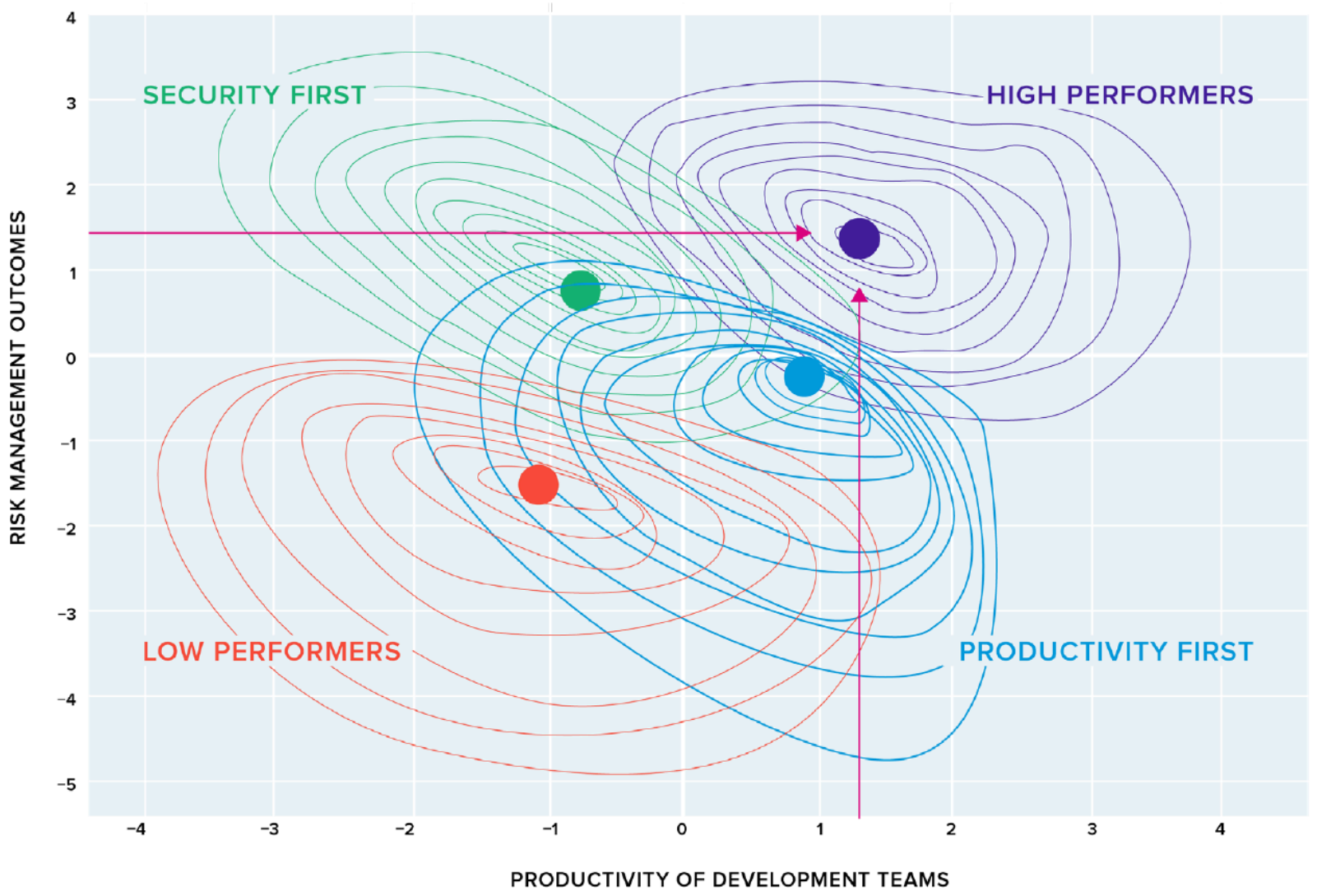
77%
more likely to automate approval, management, and analysis of

28%
more likely to enforce governance policies in CI

51%
more likely to maintain SBOMs

96%
more likely to centrally scan all deployed artifacts for security and license compliance


RISK MANAGEMENT OUTCOMES
PRODUCTIVITY OF DEVELOPMENT TEAMS
Security First
SecOps
High Performers
DevSecOps
Low Performers
Waterfall
Productivity First
DevOps
4
3
2
1
0
−1
−2
−3
−4
−5
−4
−3
−2
−1
0
1
2
3
4



HIGH PERFORMERS:

Better risk management outcomes

Higher developer productivity

Improved job satisfaction

Guidance for Enterprise Development

prioritize software supply chain and OSS management

identify your gaps and constraints

aim for quick wins

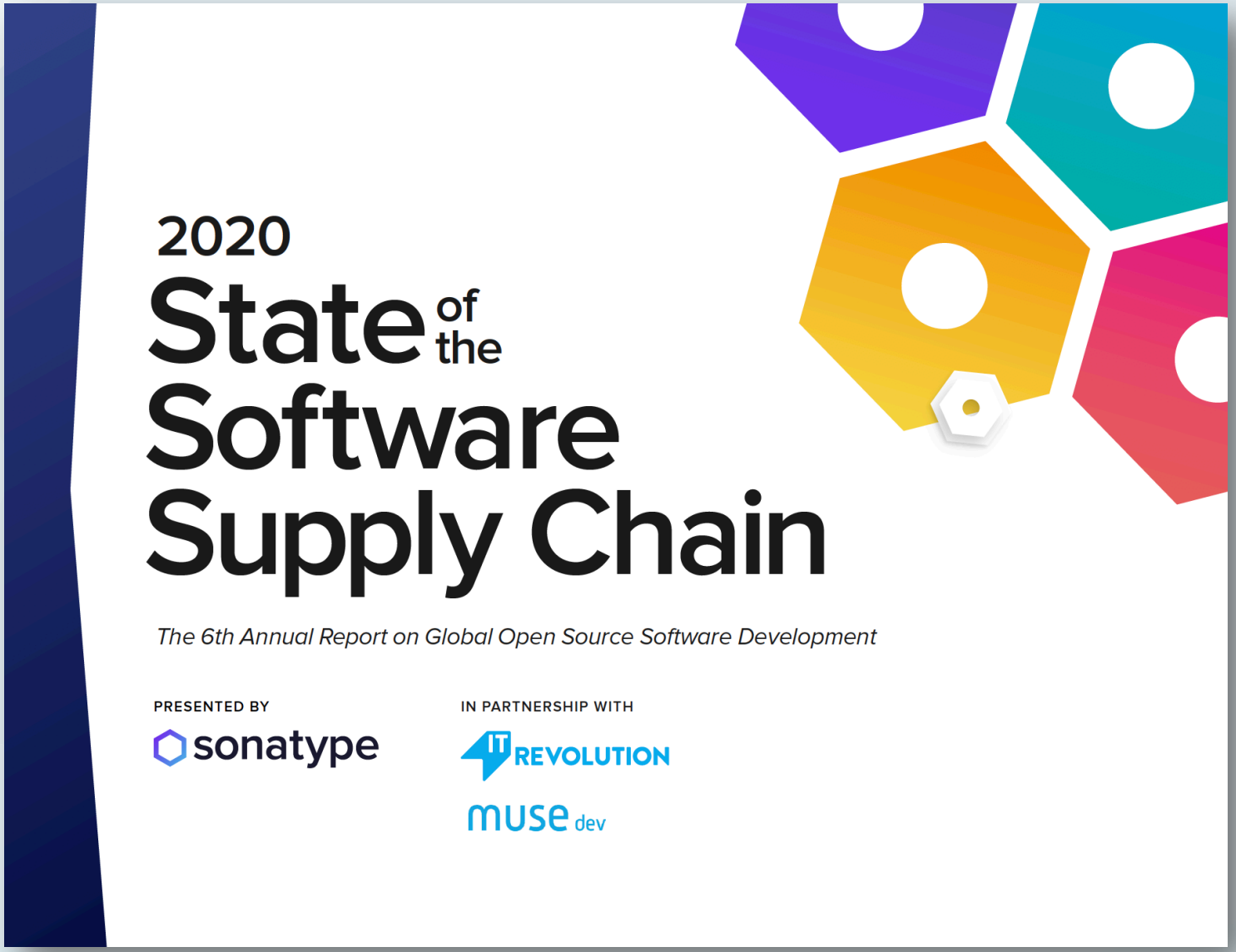
pursue speed and security improvements

happier employees

faster is better

faster is more secure

happier too



weeks@sonatype.com

Thank You!

Derek E. Weeks

@weekstweets