

AGENDA:

- Software Development Evolution
- What is DevOps?
- History of DevOps
- Why is DevOps?
- DevOps Architecture Features
- DevOps Architecture Features
- How is DevOps different from traditional IT?
- How is DevOps different from Agile?
- Applications of DevOps
- DevOps Advantages and Disadvantages
- DevOps Best Practices
- What is DevSecOps?
- DevOps vs DevSecOps
- Benefits from implementing DevSecOps
- Tools Used in DevSecOps
- Debunk Common myths

Trainer Profile



VENKATA SUBBARAO B

Cloudthat Consultant

UI | UX | JAVA Full Stack | Architect | Cloud | DevSecOps | SRE | DS



Cloud & Other Platforms

AWS, AZURE, GCP, Solution Architect,
DevSecOps, SRE

Front End

ES.Next, JavaScript, React, React Native, Vue,
Angular Ext JS, UXD, UI , HTML, CSS,
Bootstrap, Tailwind CSS, SASS, Wire Framing,
Low & High Fiddle Design, RWD, PWA ...

Back End

JAVA, Spring Boot, Node, ExpressJS, PHP,
Python, SQL(MySQL, PgSQL), NoSQL (Mongo,
Firebase, Elasticsearch, Cassandra,
Couchbase), API

Miscellaneous Tools

Docker, Kubernetes, Jenkins, Kafka, Ansible,
Terraform, SonarQube, Selenium, Atlassian
Stack, Grafana, GraphQL, ELK Stack, Git, SVN,
CVS, Redux, MobX, Thunk, Saga, Figma,
Adobe XD, Karma & Jasmine, Protractor, Jest,
SonarQube, Photoshop, Illustrator, Digital
Marketing, Agile Scrum ...



COGNIZANT

CRIO.DO

JIGSAW ACADEMY

TELSTRA

UNEXT

BAJAJ FINSERVE

MANIPAL PRO LEARN

ETEAMINC

PROBITS

MANTT

AWIGN

MYYESM

ANTHEM

UST-GLOBAL

MAZENET

TATA SED

HONDA USA

COLORS SOFT

TRAINING CURVE

6BENCHES

EDUPPINNACLE

AIMS MBA COLLEGE

VERZE

INVENTATEQ

CODING SUPER STAR

VSN TRADERS

IMPACT FOUNDATION

CRAYONZE

<https://www.linkedin.com/in/bvsrao-venkat-4014133b/>

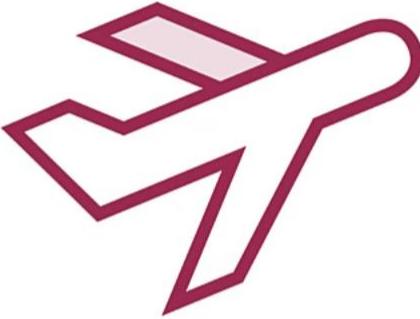
Software Development Evolution

- SDLC Quick Overview
- Agile & Scrum Methodology
- DevOps
- DevSecOps

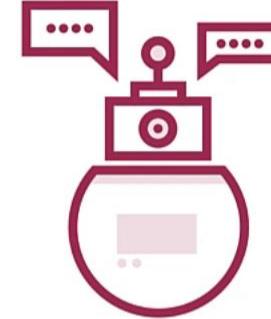
We Now Expect to Find Software Everywhere



Healthcare



Travel experiences



Customer support



Sporting events



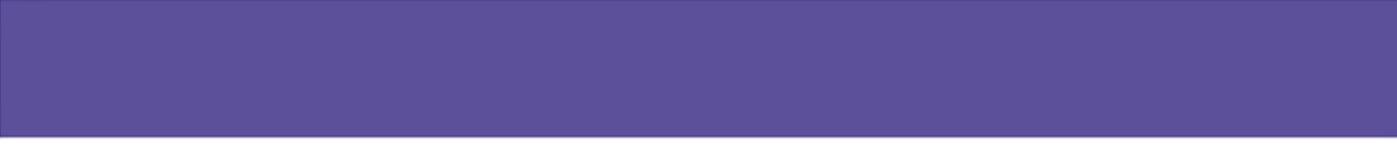
Office work



Shopping

Being “Good at Software” Matters

U.S. consumers changing shopping behavior such as trying new brands or shopping methods



75%

Source: <https://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/the-great-consumer-shift-ten-charts-that-show-how-us-shopping-behavior-is-changing>

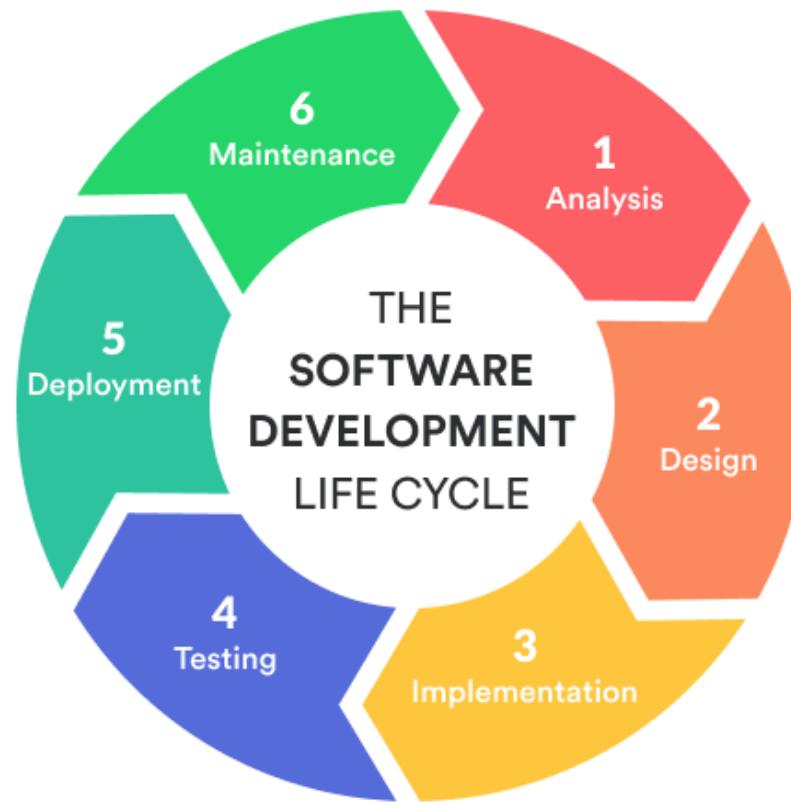
Businesses that think they have less than a year to embrace digital experiences before suffering financially and losing market share



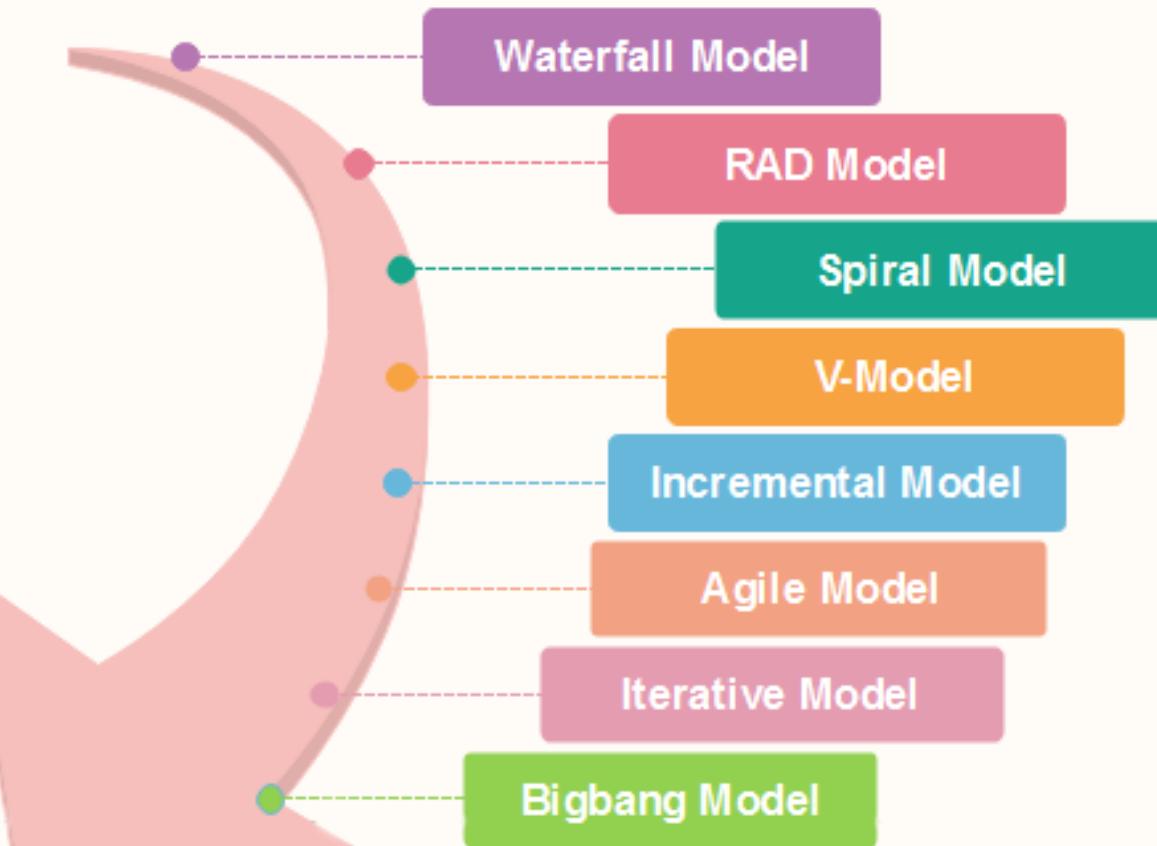
55%

Source: https://www.progress.com/docs/default-source/default-document-library/landing-pages/dach/ebook_digitaltransformation_final.pdf

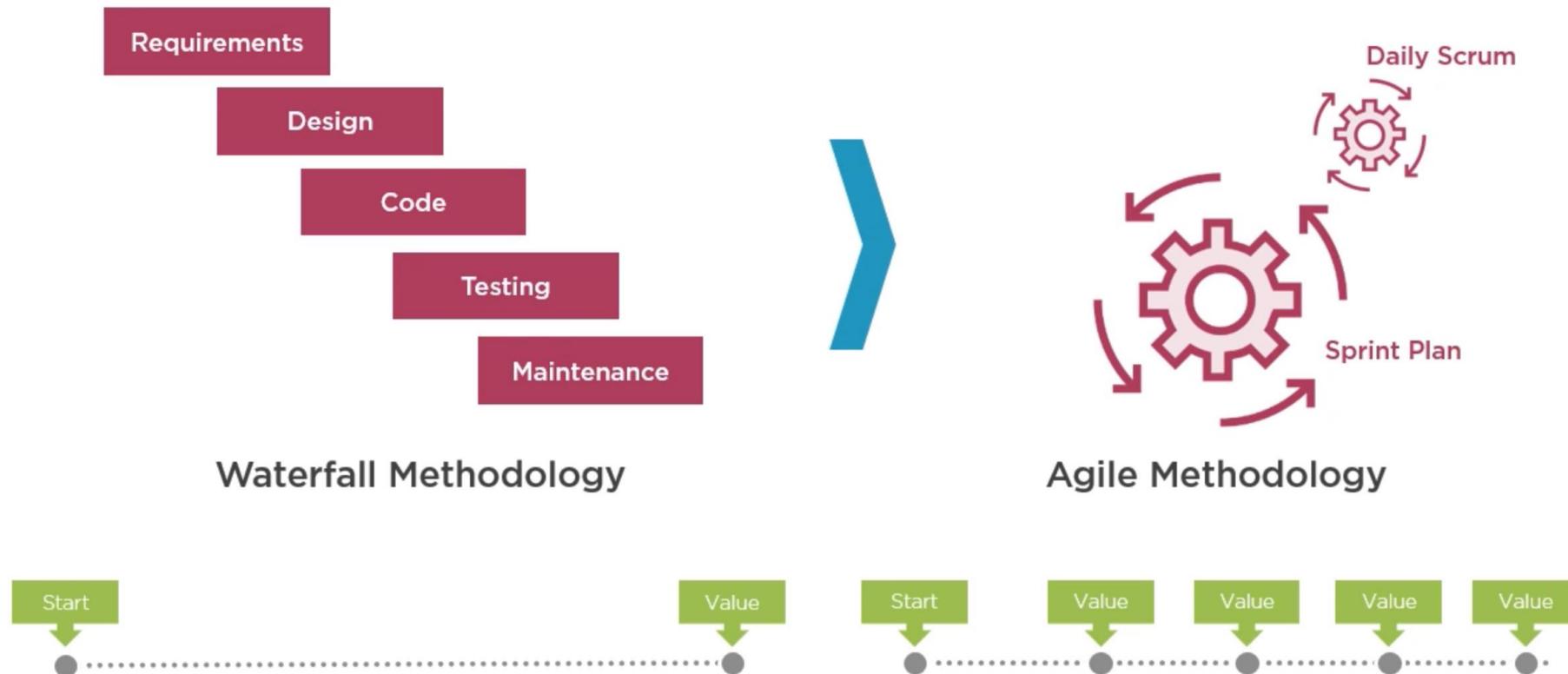
SDLC



SDLC (Models)



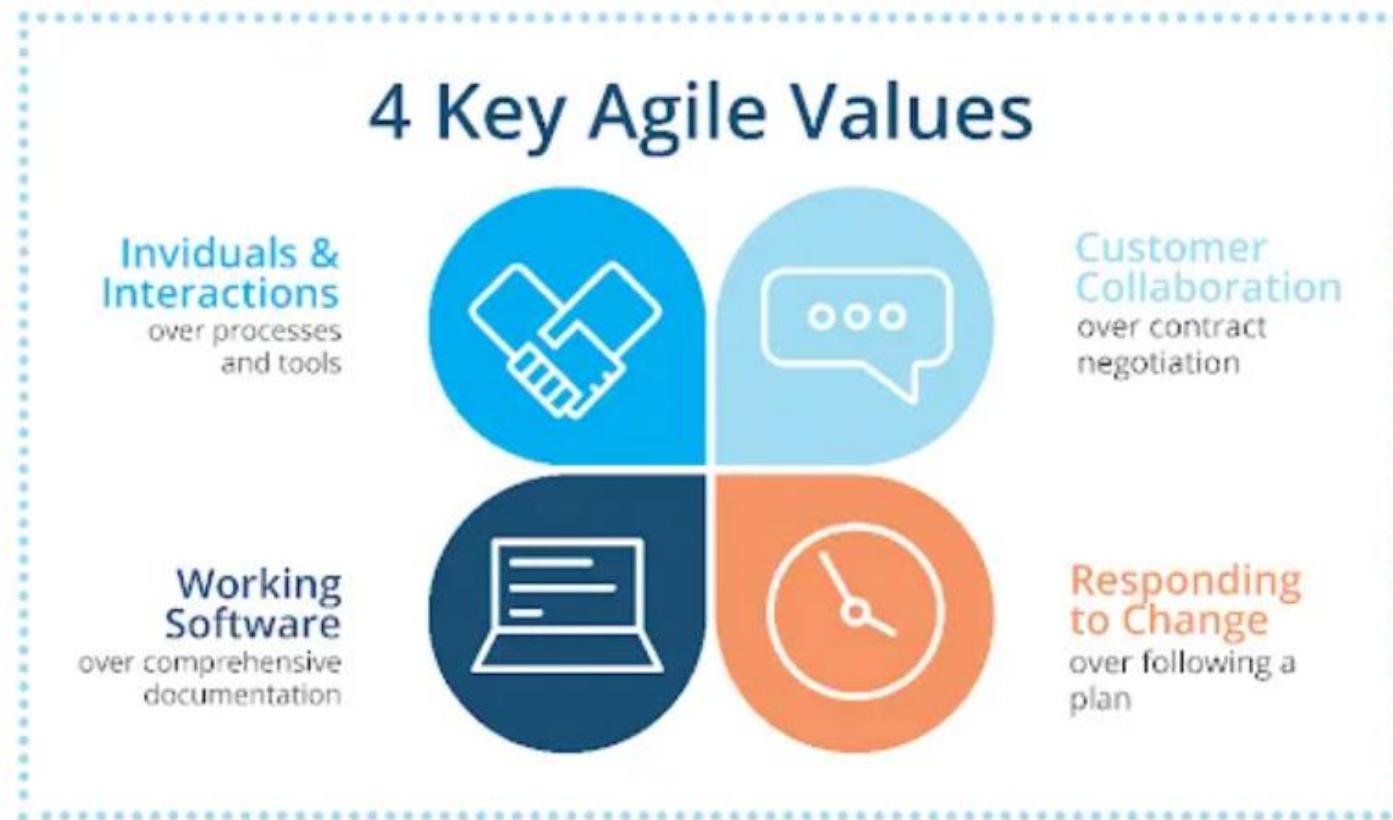
Software Development Evolution



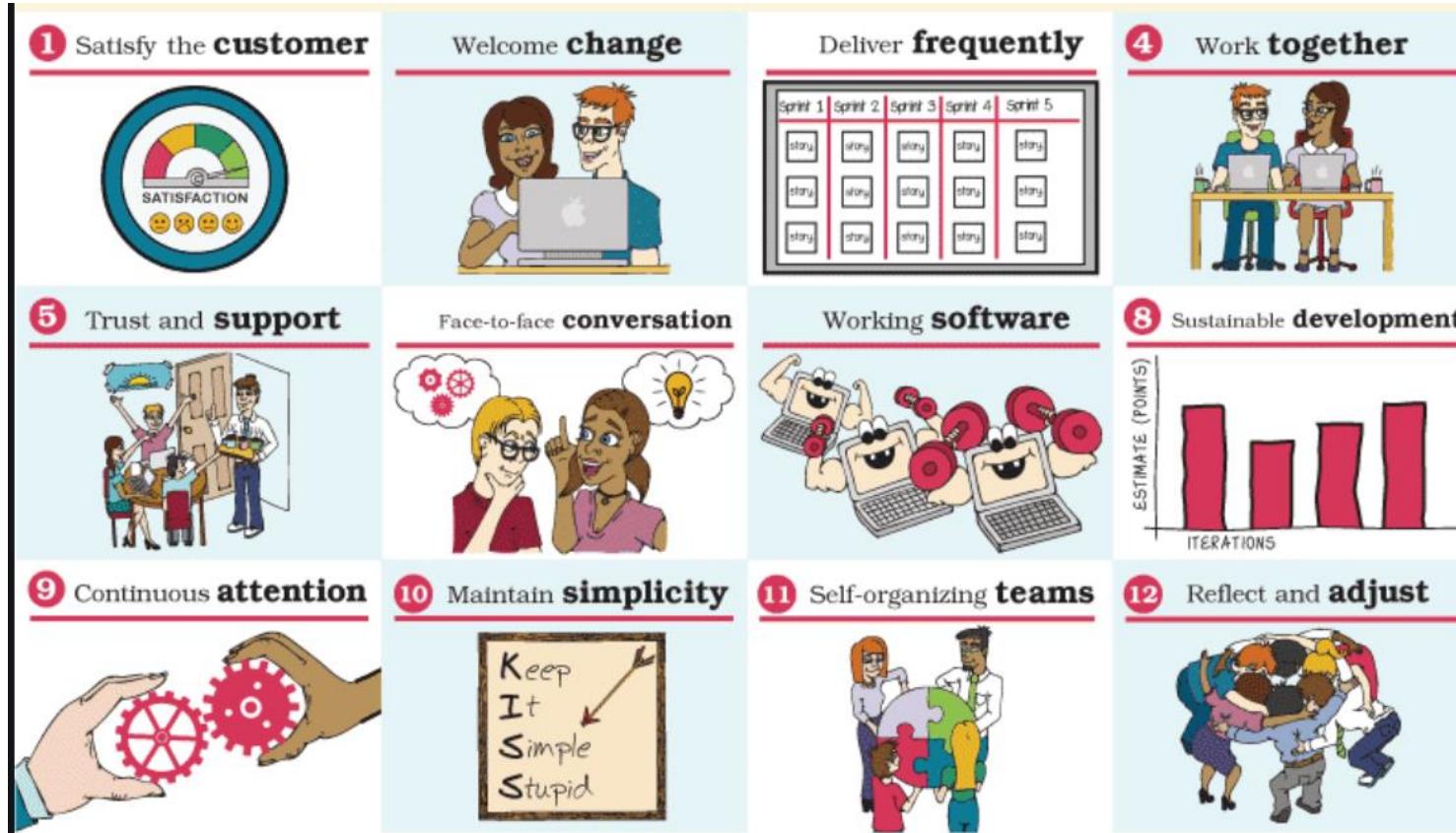
Agile

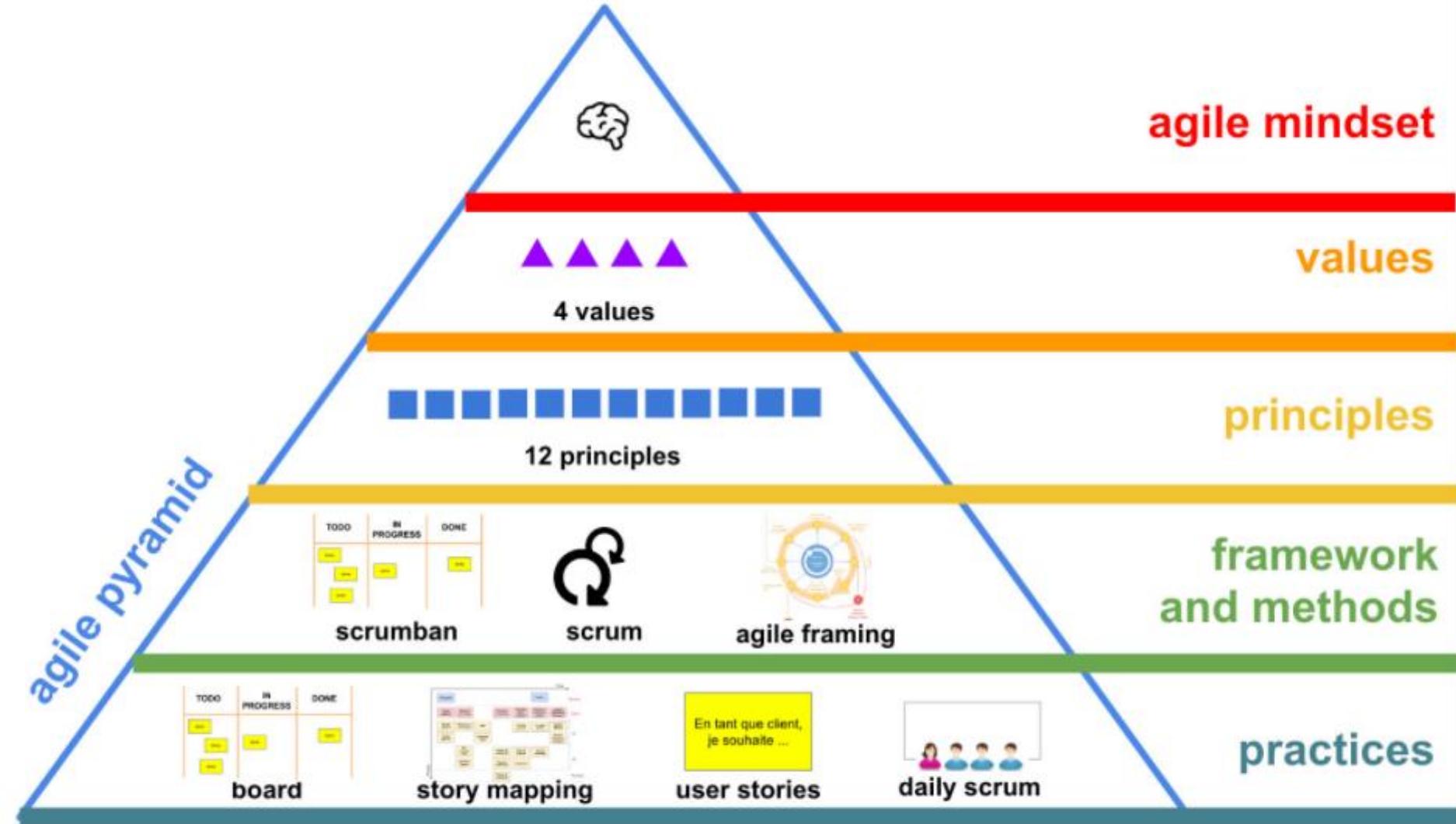


4 KEY VALUES

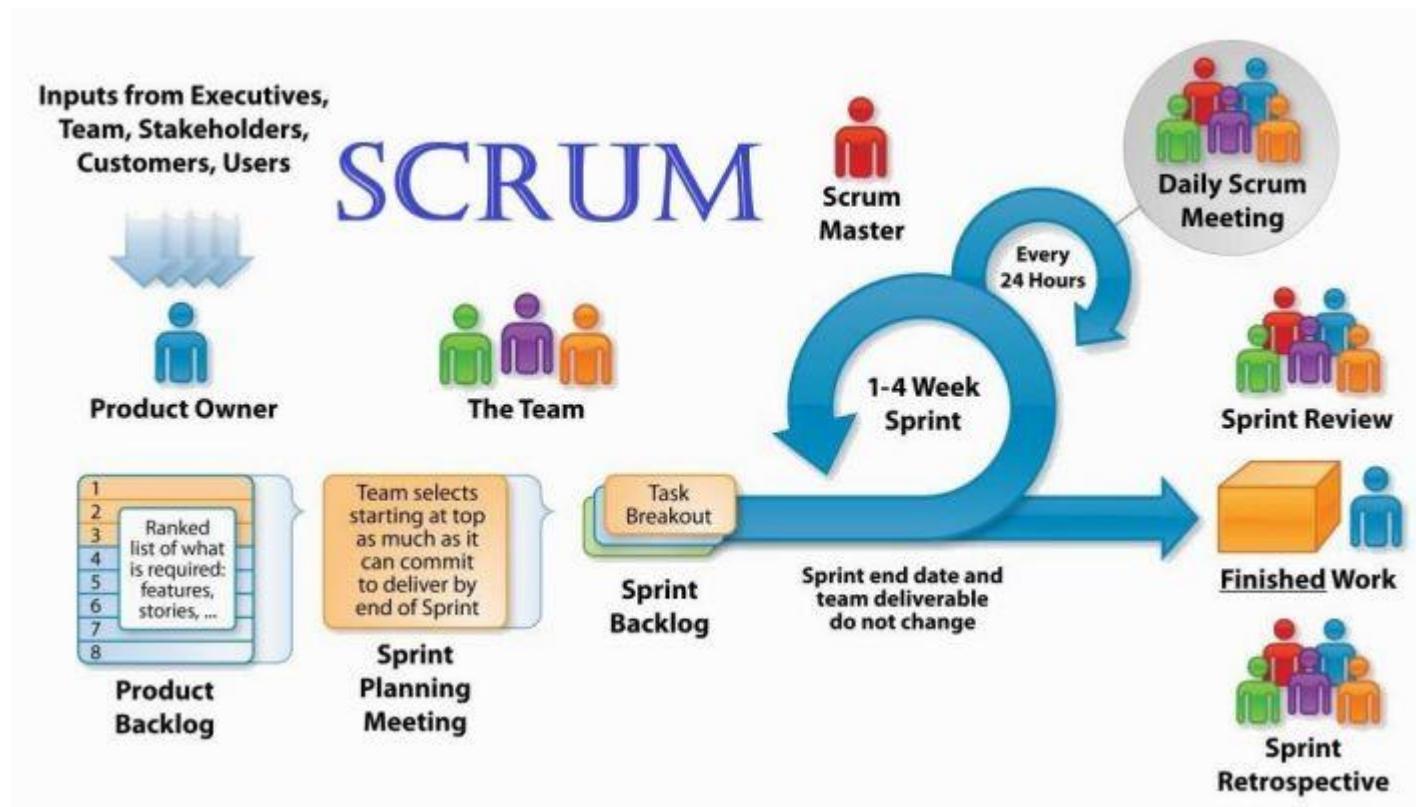


12 PRINCIPLES

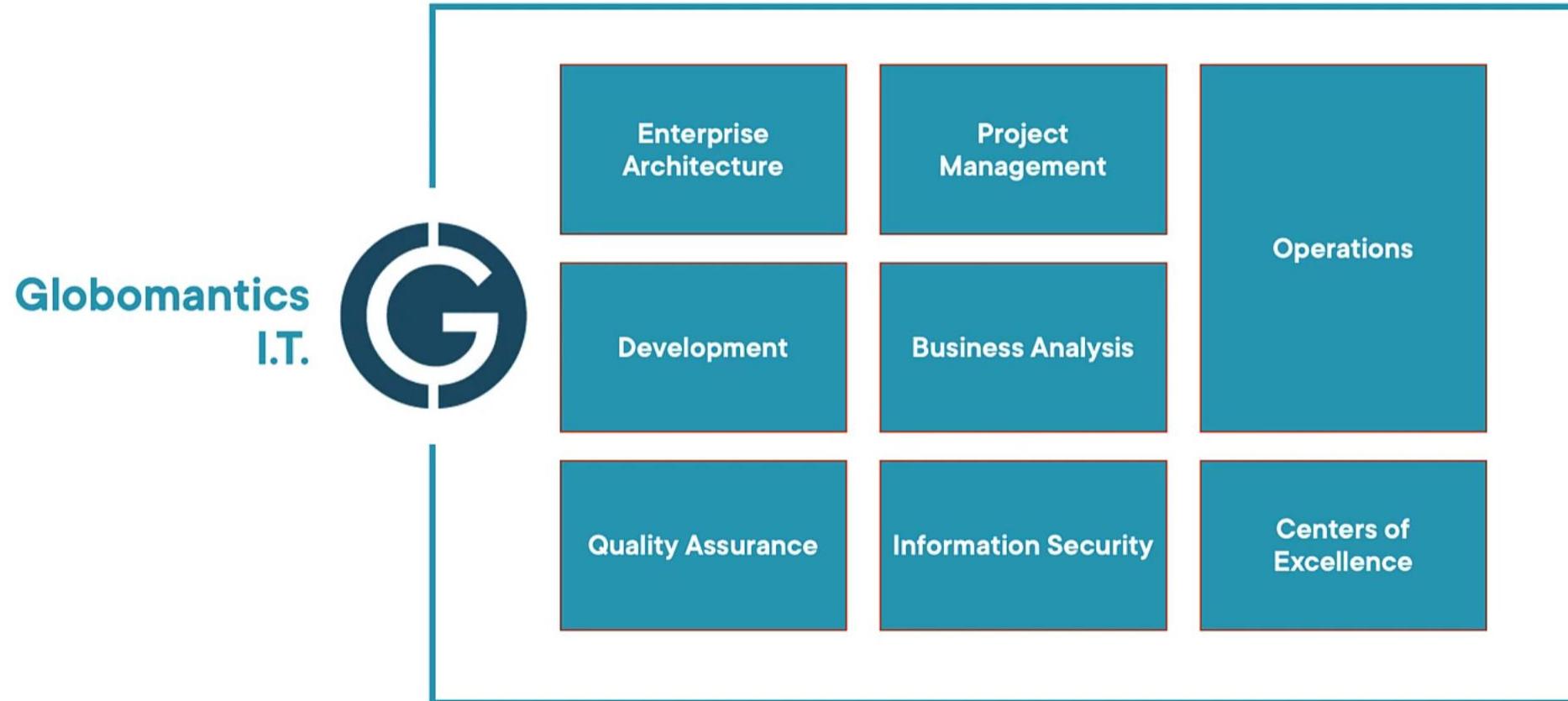




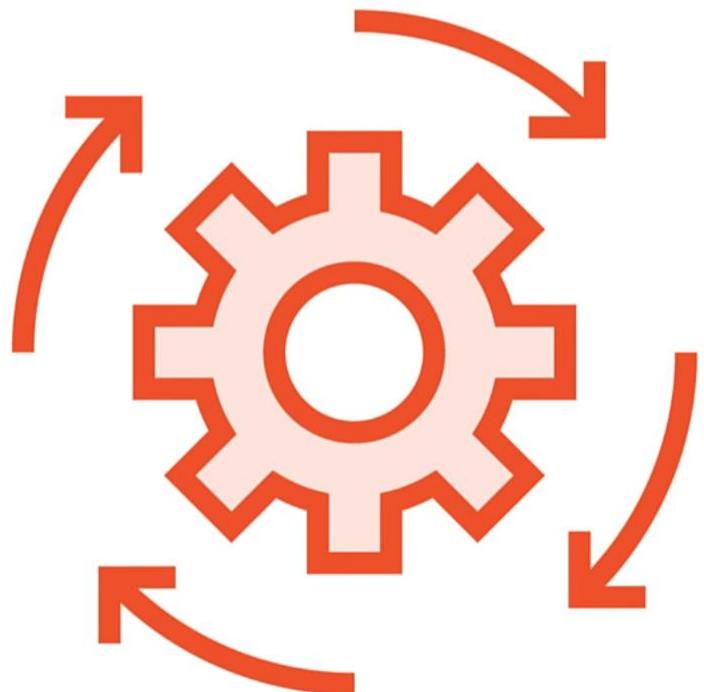
Scrum



How Globomantics Organizes I.T. Today



Pain Point: Limited Automation



- Manual virtual machine builds, even in cloud, causes over-provisioning**
- Ticket-defined networking prevents rapid changes**
- Lack of test coverage in code means manual tests**
- Only parts of environments are stood up with automation, impacting setup time and DR**

Pain Point: Slow Delivery



- Idea-to-production takes months, even years**
- Work is piled up in front of multiple teams**
- No self-service platforms for app teams**
- Multiple change freezes throughout the year**
- Big-bang releases are the norm, causing downtime**
- Limited experimentation because of slow feedback loops**

Pain Point: Poor Uptime



Poor change failure rates, with the same mistakes occurring repeatedly

Long mean-time-to-recovery

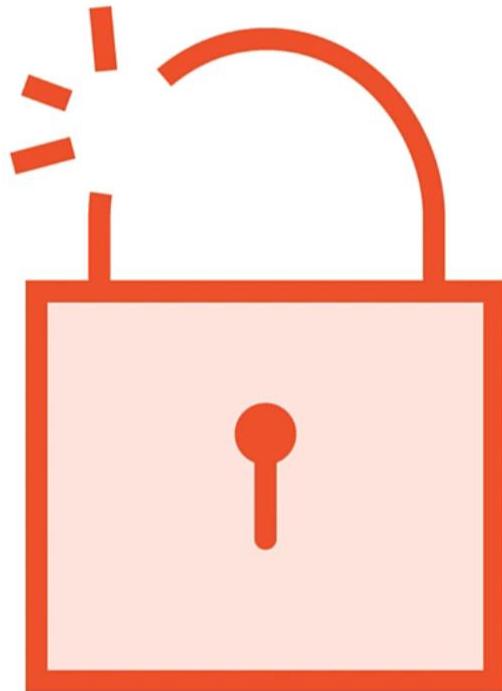
Panic when things go wrong

Lack of complete, centralized telemetry delays resolution time

Over-alerting has led to ignored alarms

Teams focused on preventing failures

Pain Point: Security Gaps



- Security concerns offloaded to InfoSec**
- No common approach to credential and secrets management**
- Inconsistent hardening of compute environments, including container images**
- Challenges understanding vulnerabilities in app dependencies**
- Different policies applies to cloud and on-prem environments**
- Security often sacrificed to attempt speed**

Pain Point: Lacking a Customer Focus



Teams think about their silo, not the total product

Every department has unique measure of success

No shared definition of “done”

Quality issues deprioritized to ship on time

No A/B testing or feature-based usage monitoring

Customer support is handled centrally

The Result

Poor customer satisfaction

Revenue is declining

Best employees are leaving

No innovations in years

LOBs lack faith in I.T.

Us vs. them mentality taking over

Looking to Manufacturing for Inspiration

**Automotive companies like Toyota
reinvented manufacturing**

**Fresh thinking on flow, continuous
improvement, and empowerment**

**Focus on consistency, sustainable work,
and eliminating waste**



Identifying Waste



Knowledge waste



Waiting waste



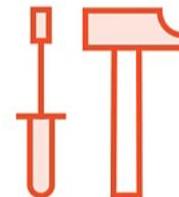
Over-processing waste



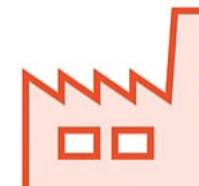
Motion waste



Transportation waste



Correction waste

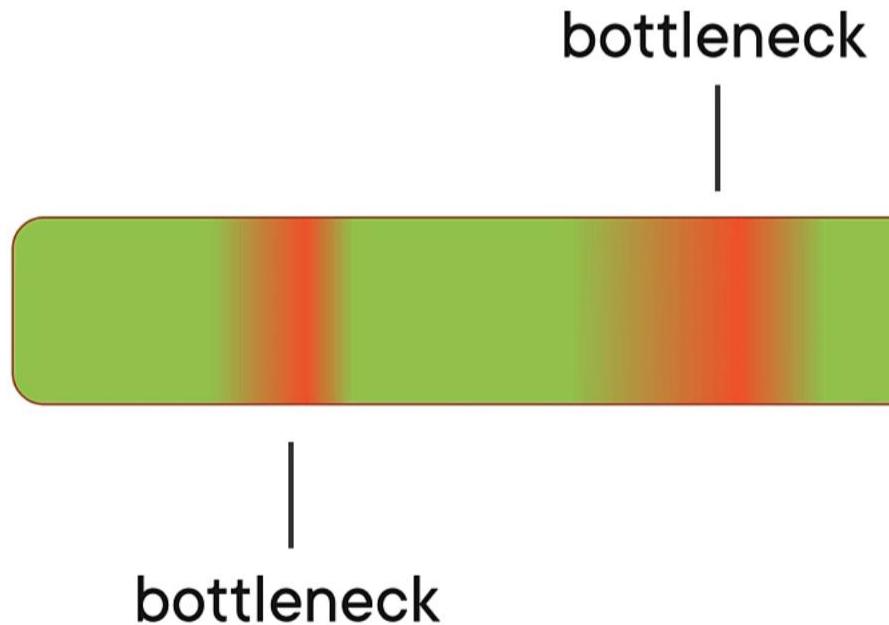


Inventory waste



Over-production waste

Exploring the Theory of Constraints



Throughput is limited by the constraint(s)

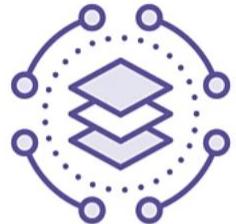


Optimizing **before** it results in excess inventory



Optimizing **after** it results in starvation

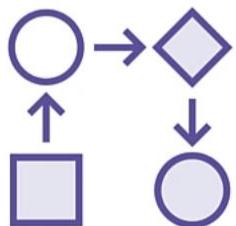
How Lean Relates to Software Delivery



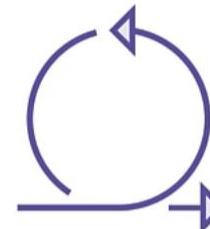
Reorient towards product thinking



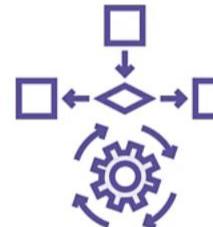
Remove wasteful processes



Continuously deliver software in small batches



Focus on continuous improvement

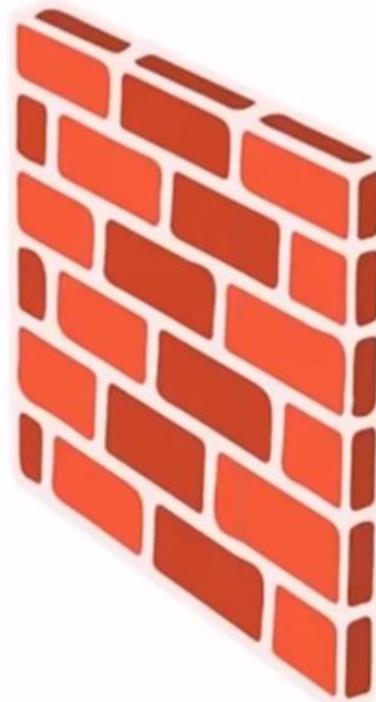


Automate to remove overburdening of staff



Attack bottlenecks that decrease flow

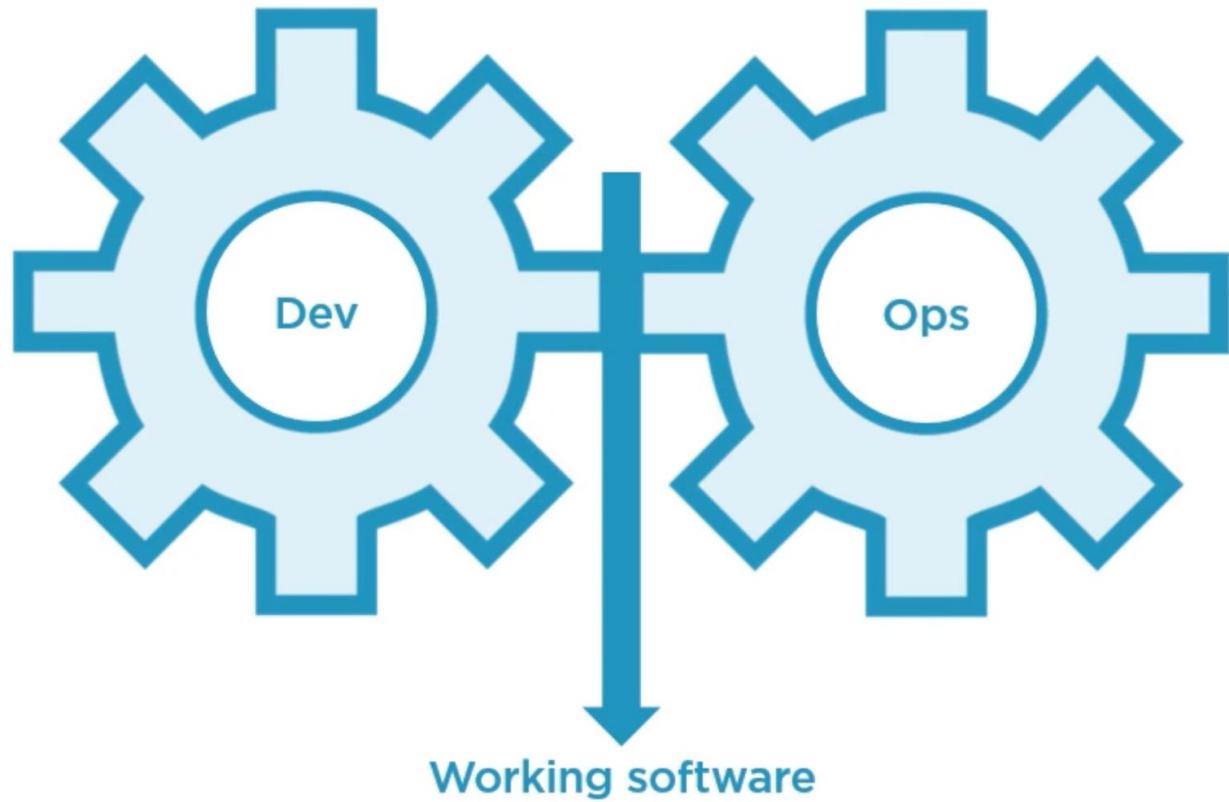
Agile Limitations



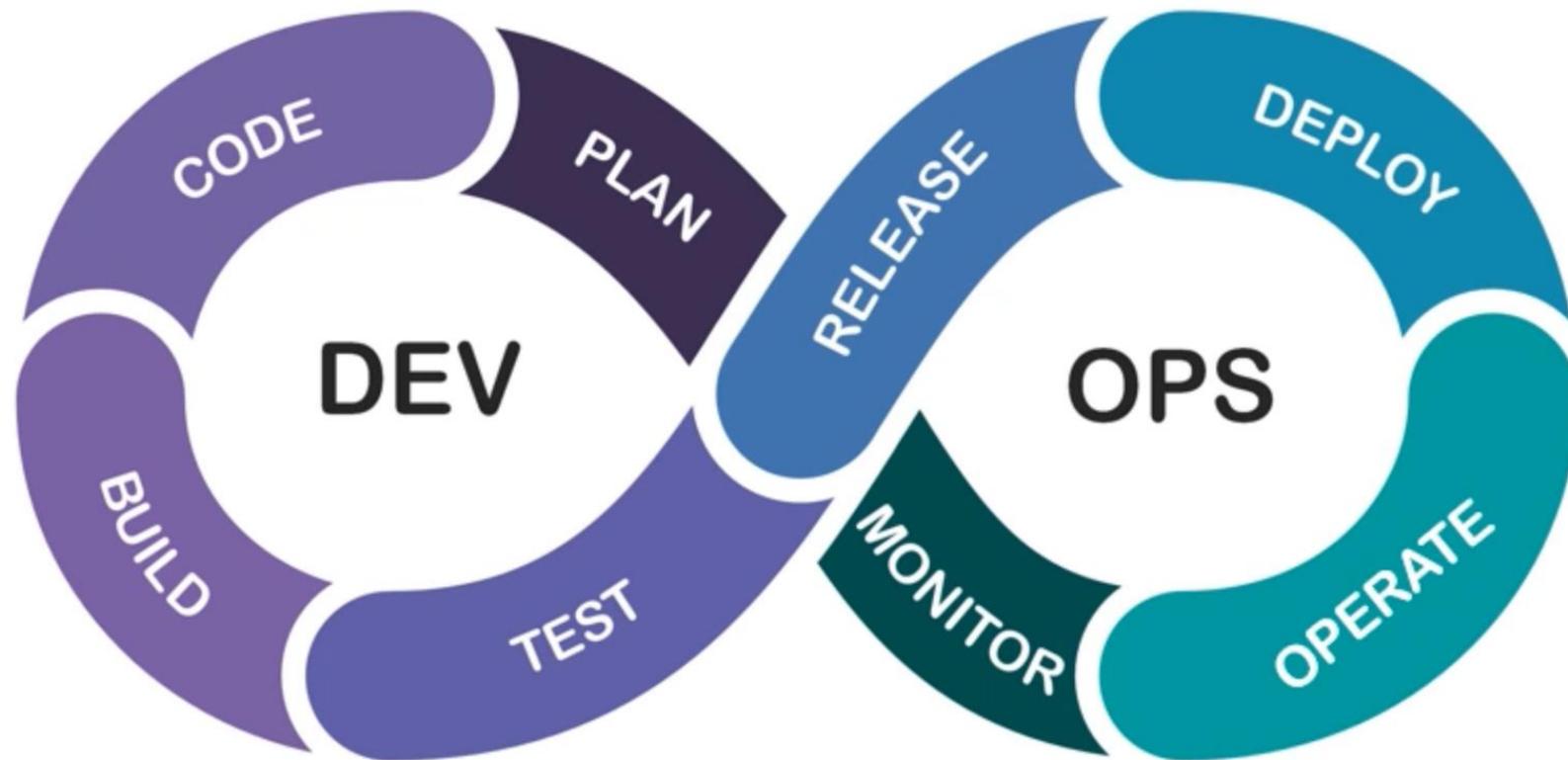
Information Security Friction



Solution is DevOps



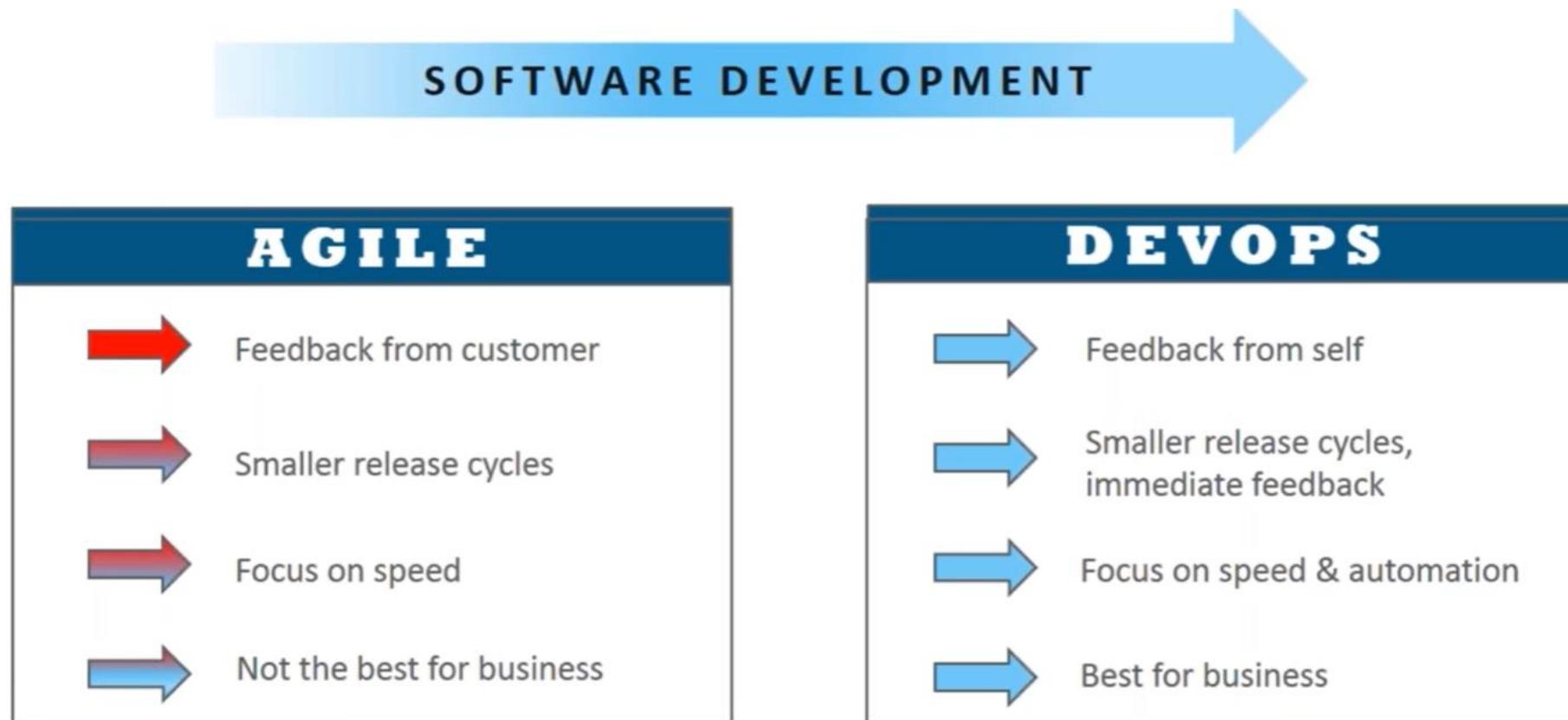
Merge Dev and Ops to Remove Friction



Agile vs DevOps

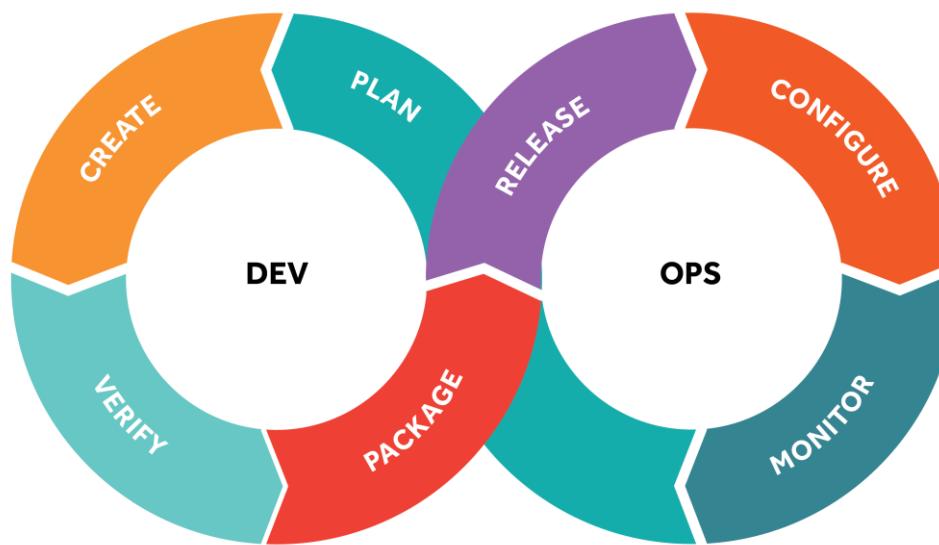


Agile vs DevOps



DevOps

DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology.



Defining DevOps

“DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach.”

How DevOps Works?

Under a DevOps model, development and operations teams are no longer “siloed.” Sometimes, these two teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function.

In some DevOps models, quality assurance and security teams may also become more tightly integrated with development and operations and throughout the application lifecycle. When security is the focus of everyone on a DevOps team, this is sometimes referred to as DevSecOps.

These teams use practices to automate processes that historically have been manual and slow. They use a technology stack and tooling which help them operate and evolve applications quickly and reliably. These tools also help engineers independently accomplish tasks (for example, deploying code or provisioning infrastructure) that normally would have required help from other teams, and this further increases a team’s velocity.

Having a Product Mindset

Project-based mindset

Funding by project milestones

Success based on time, budget targets

Teams assembled from staff allocated for the term of the project

Durable knowledge lives in documentation

Has a defined end date, with little attention on post-release lifecycle

Product-based mindset

Funding based on output

Success based on business value

Teams comprised of multi-disciplinary staff to one product at a time

Durable knowledge within the team and select documentation

Look at multi-year lifecycle and consider ongoing activities

What DevOps is NOT



DevOps is not a team or software methodology



DevOps is not something you buy



DevOps is not an IT-only effort



DevOps is not just automating infrastructure



DevOps is not easy!

The Data Proves This Approach Works

973x

More frequent code
deployments

3x

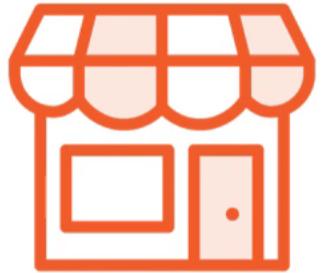
Lower change failure
rate

1.6x

More likely to meet or
exceed organizational
goals when integrating
security best practices

Source: 2021 Accelerate State of DevOps Report

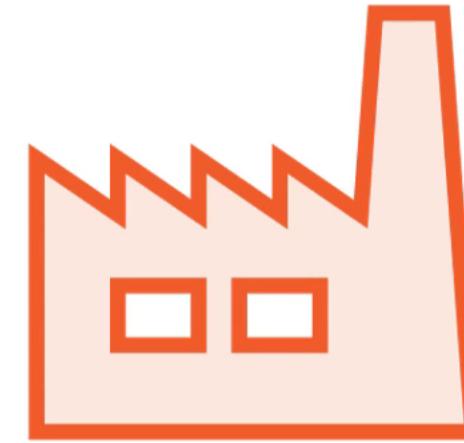
Value Through IT



Value Through IT

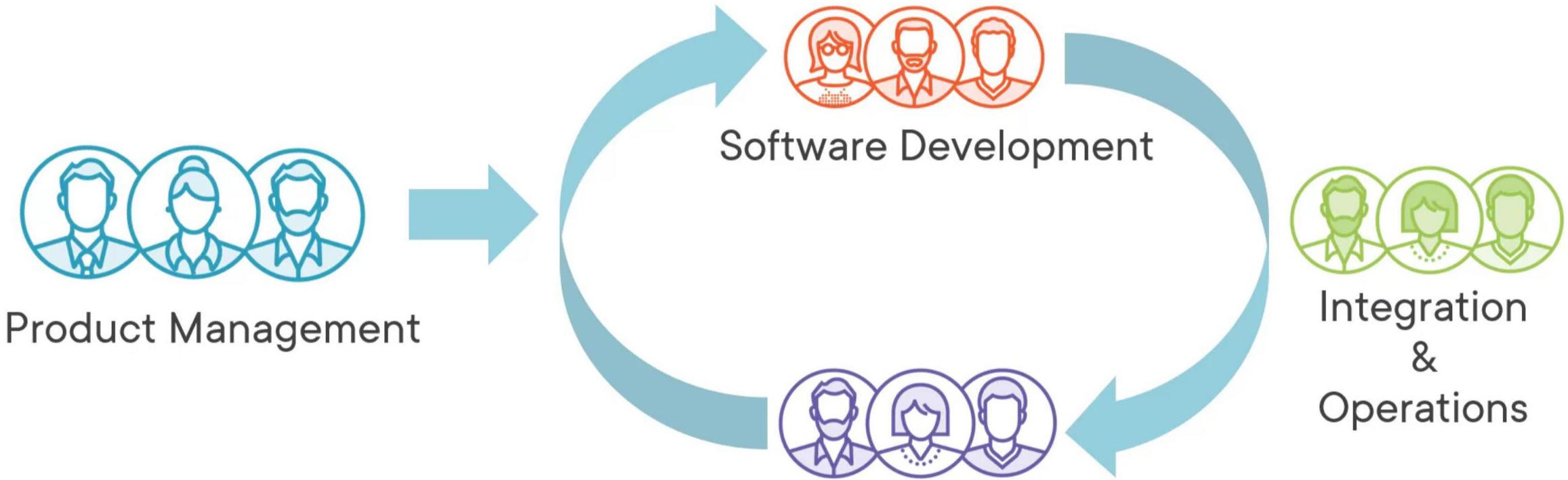


- Manage warehouses
- Manage inventory
- Manage schedules



Logistics Company

How Value Through IT Creates Software



- Iterations of 3 weeks
- 3 weeks of work

How Value Through IT Creates Software



- Additional stabilization iteration

The Software Teams of Value Through IT



Warehouse Module



Schedule Module



Inventory Module



Integration



Operations

The Software Development Process



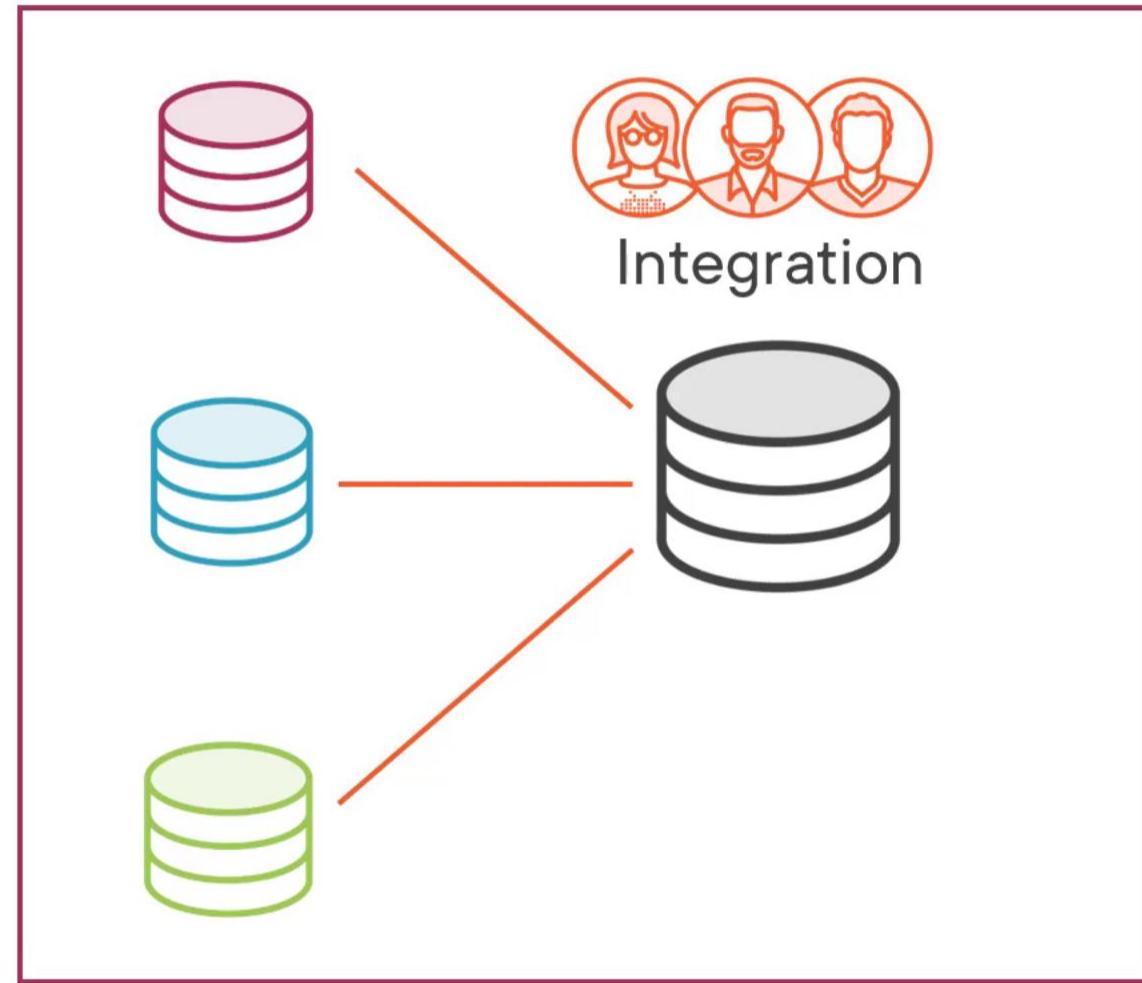
Warehouse Module



Schedule Module



Inventory Module

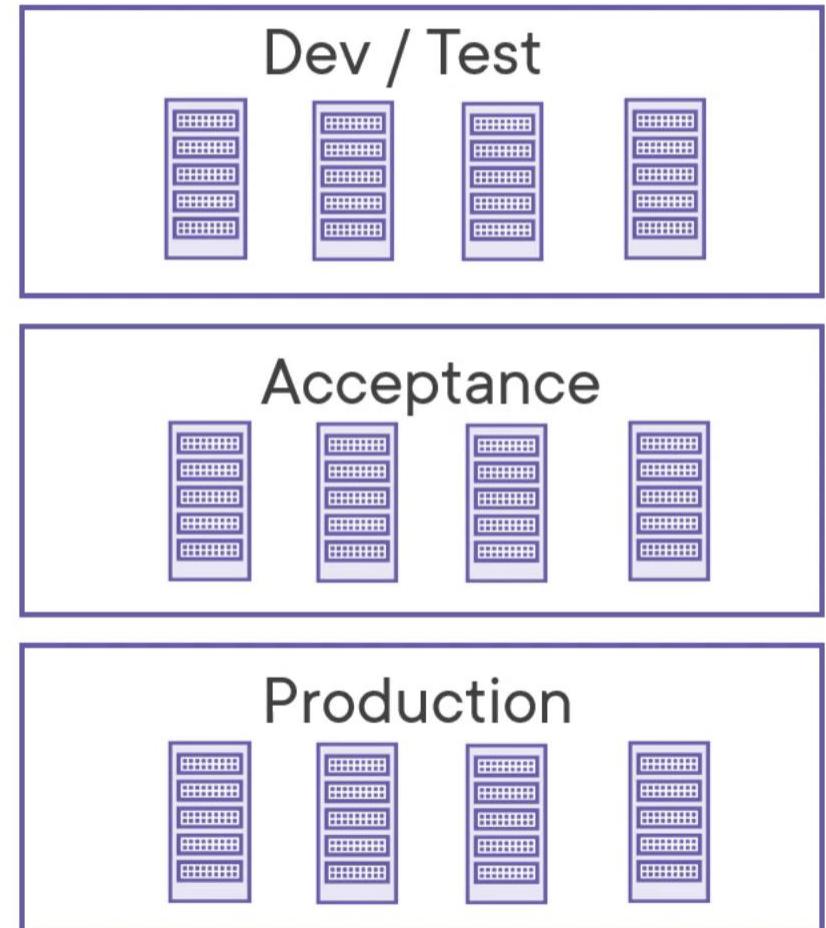


The Software Development Process

At the end of each iteration:

- Merge the module branches into the main source control code
- Build the integrated code locally
- Pass it on to operations for deployment

The Deployment Process



Software Development and Deployment



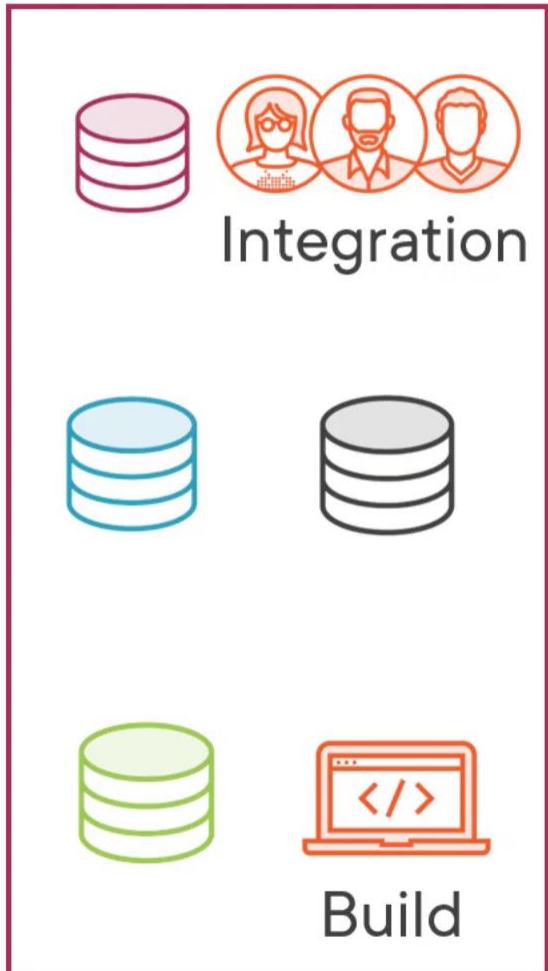
Warehouse
Module



Schedule
Module



Inventory
Module



Source control



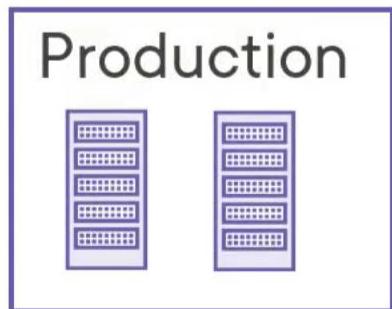
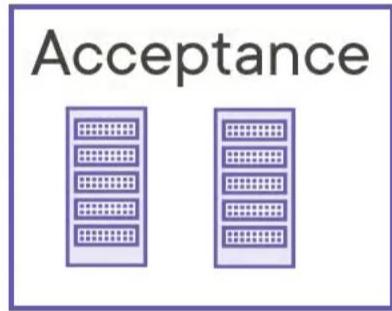
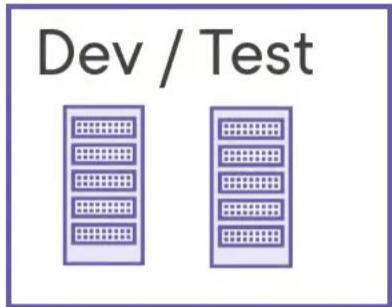
Release



Instructions



Operations



Software Development and Deployment



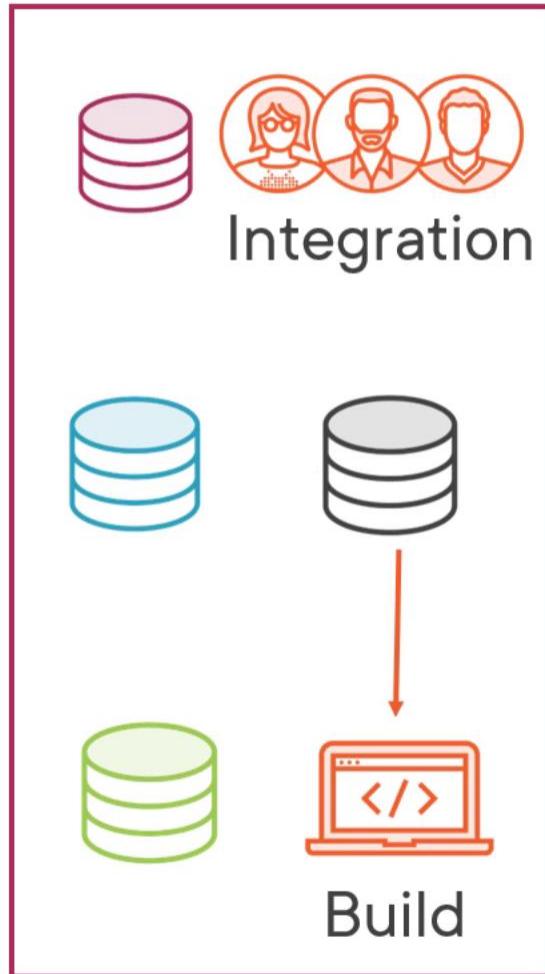
Warehouse
Module



Schedule
Module



Inventory
Module



Source control



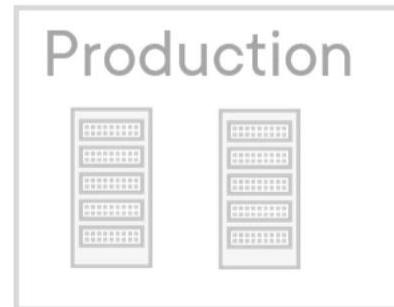
Release



Instructions



Operations



Software Development and Deployment



Warehouse
Module



Schedule
Module



Inventory
Module



Source control



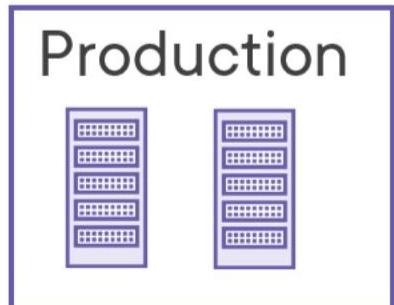
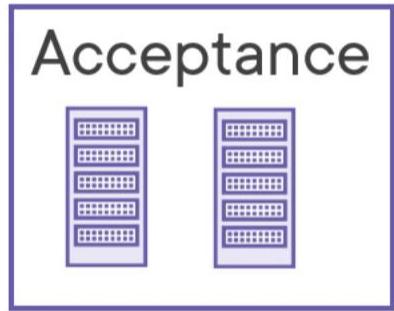
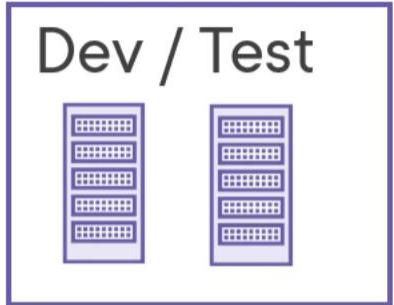
Release



Instructions



Operations



What Can We Improve?

Integration



Integration



Build

Deployment



Release



Instructions



Operations

What Can We Accomplish?



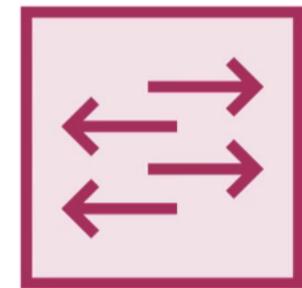
Higher Quality



Faster Delivery



Lower Costs



More Flexibility

What Can We Accomplish?



Happy Users

Customer Satisfaction

How Do We Get There?

CI

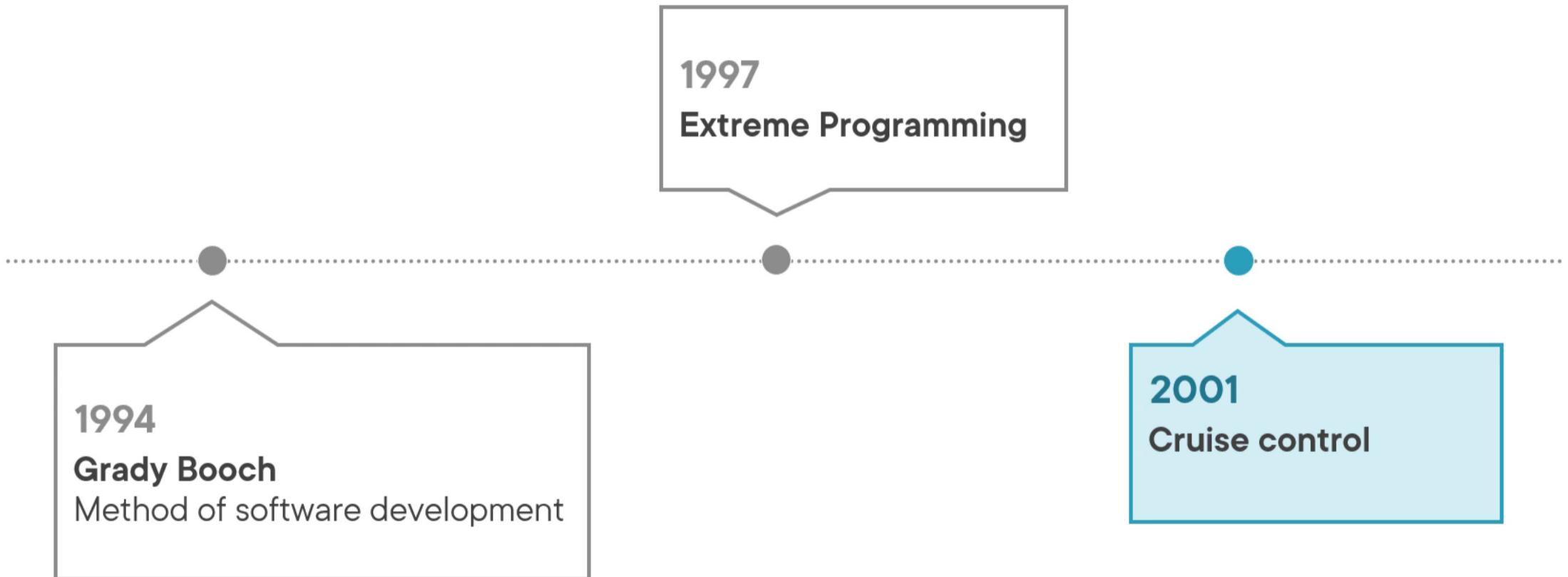
Continuous Integration

CD

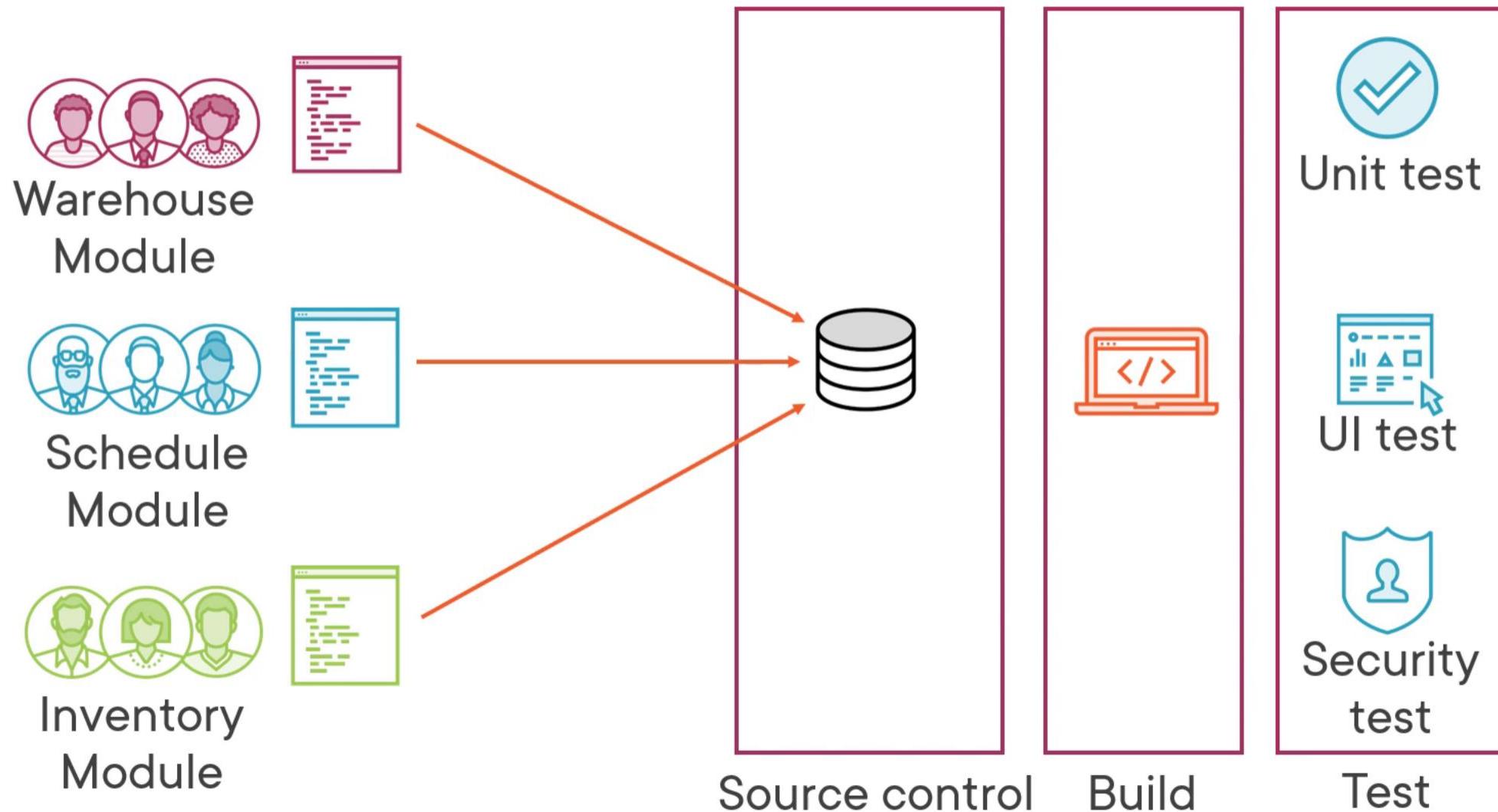
Continuous Delivery

Continuous Integration (CI) is a
development practice that requires
developers to integrate code into a
shared repository several times a day

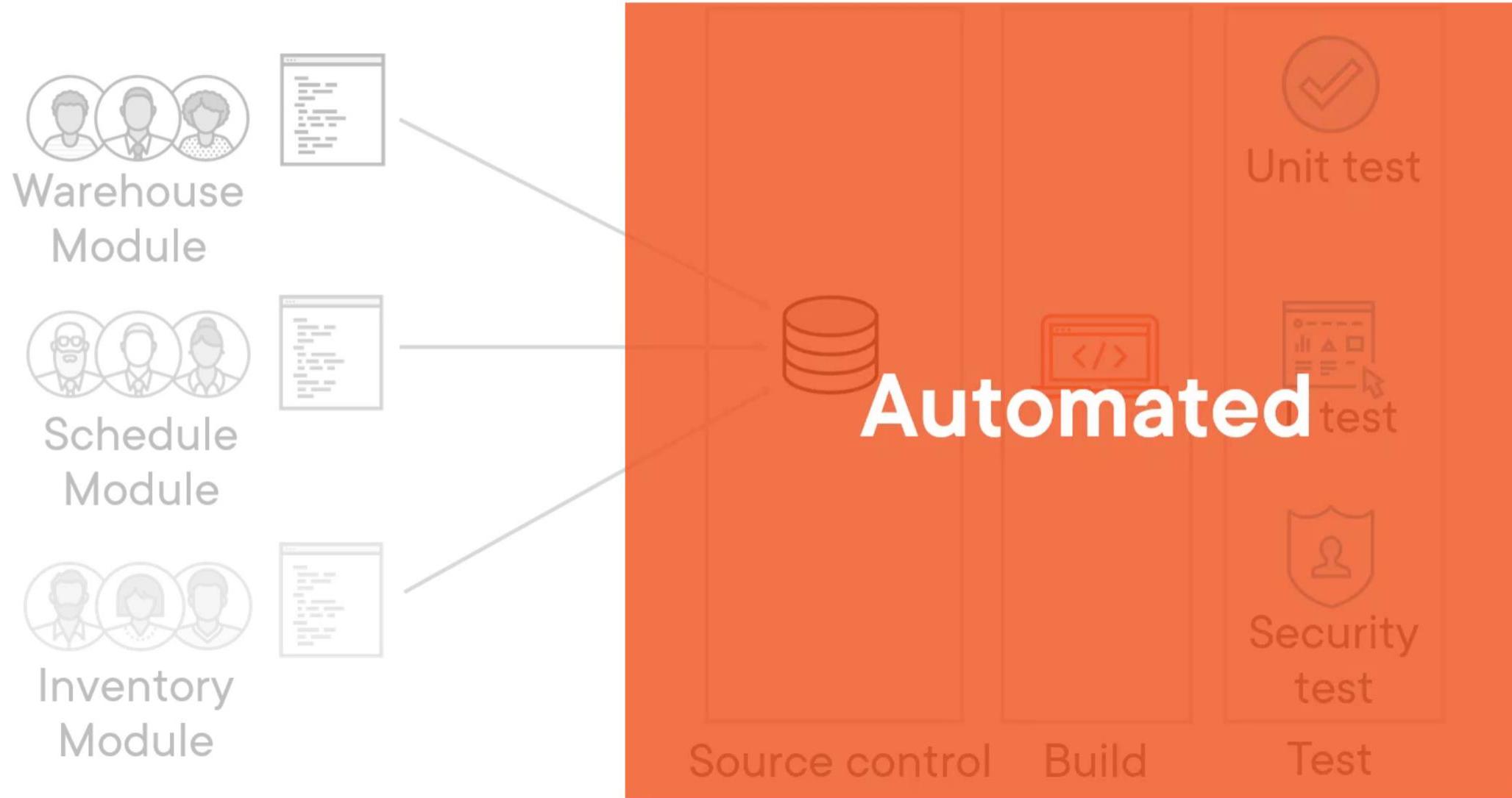
Where Did Continuous Integration Come From?



Continuous Integration



Continuous Integration





CI principles

- Have a single place where all the code lives**
- Everyone commits to the mainline every day**
- Automate the build process**
 - Fix broken build immediately
 - Make and keep the build fast
- Every commit triggers a build**
- Automate the testing process**
- Everyone has access to the latest result**
- Everyone can see everything**



CI benefits

Integration takes less effort

Issues will come up more early

Automation means less issues

The process is more visible

Improved team communication

Short integration iterations means more flexibility

The code is ready to be delivered more often

What Can CI Accomplish?



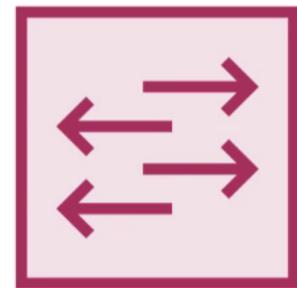
Higher Quality



Faster Delivery



Lower Costs



More Flexibility

Continuous Delivery (CD) is a
software development discipline
where software **can be released to**
production at any time

Difference Between Delivery and Deployment

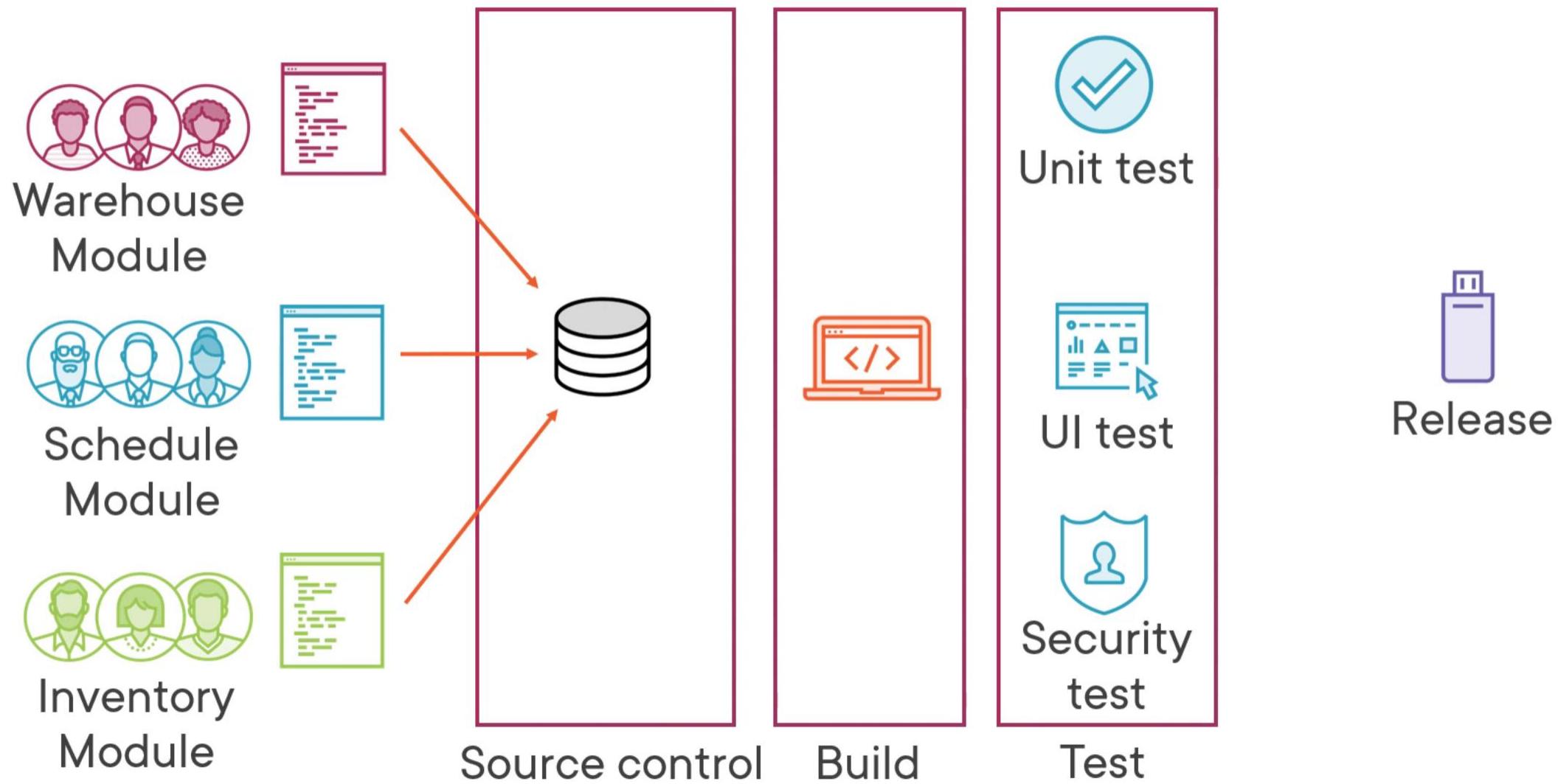
Continuous Delivery

Software **can** be deployed to production at any time

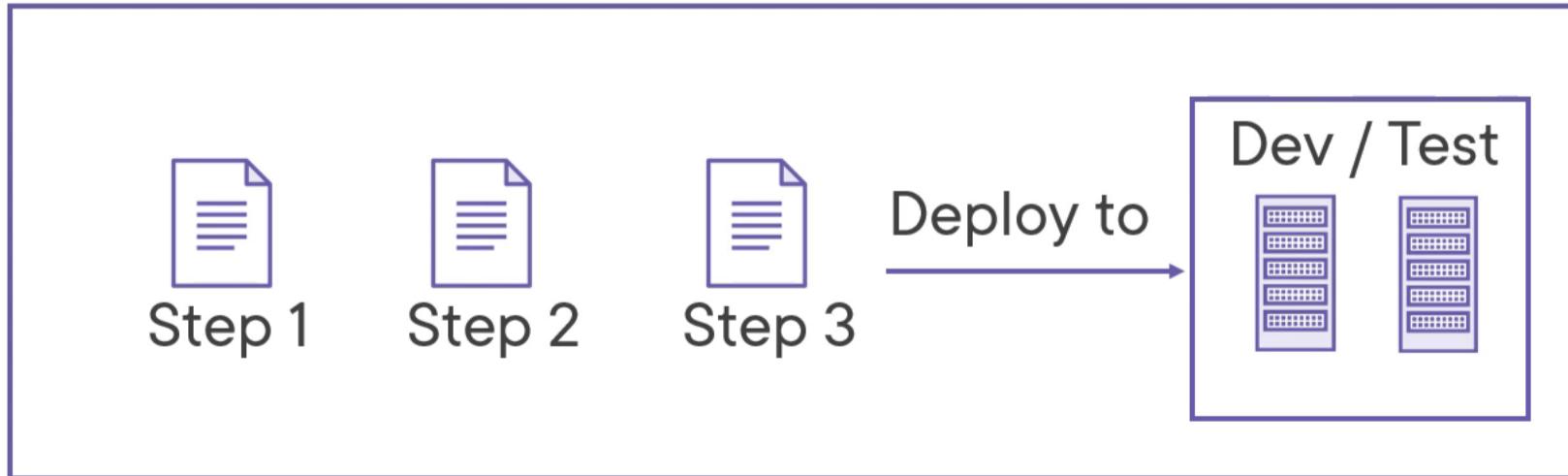
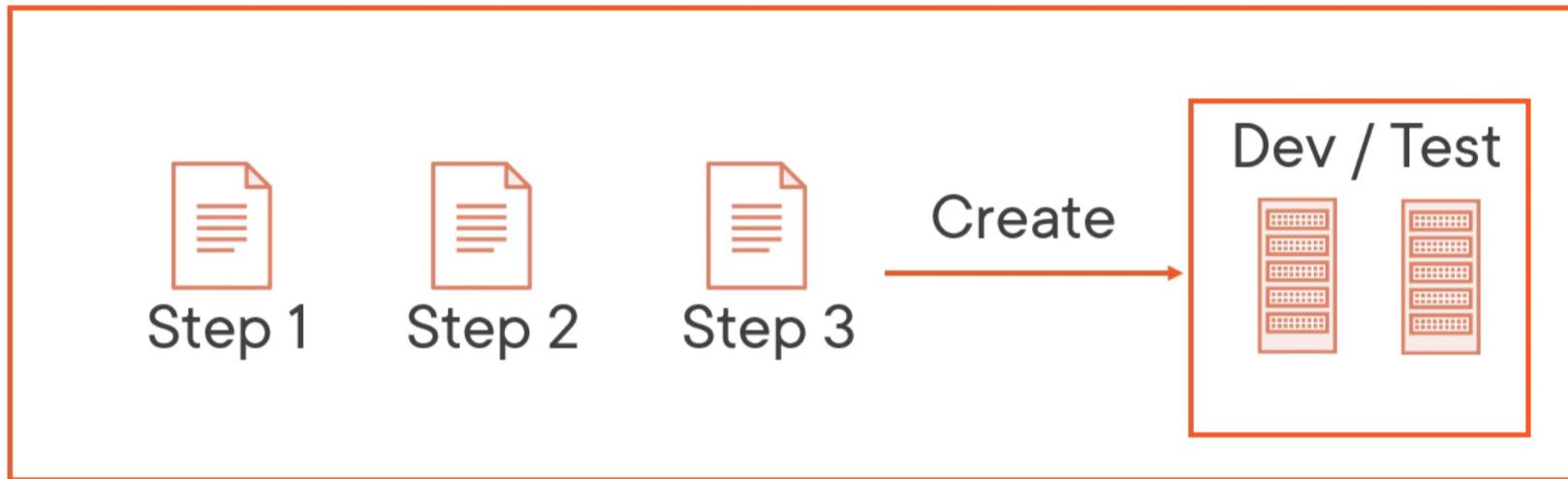
Continuous Deployment

Software **is automatically** deployed to production all the time

Continuous Delivery

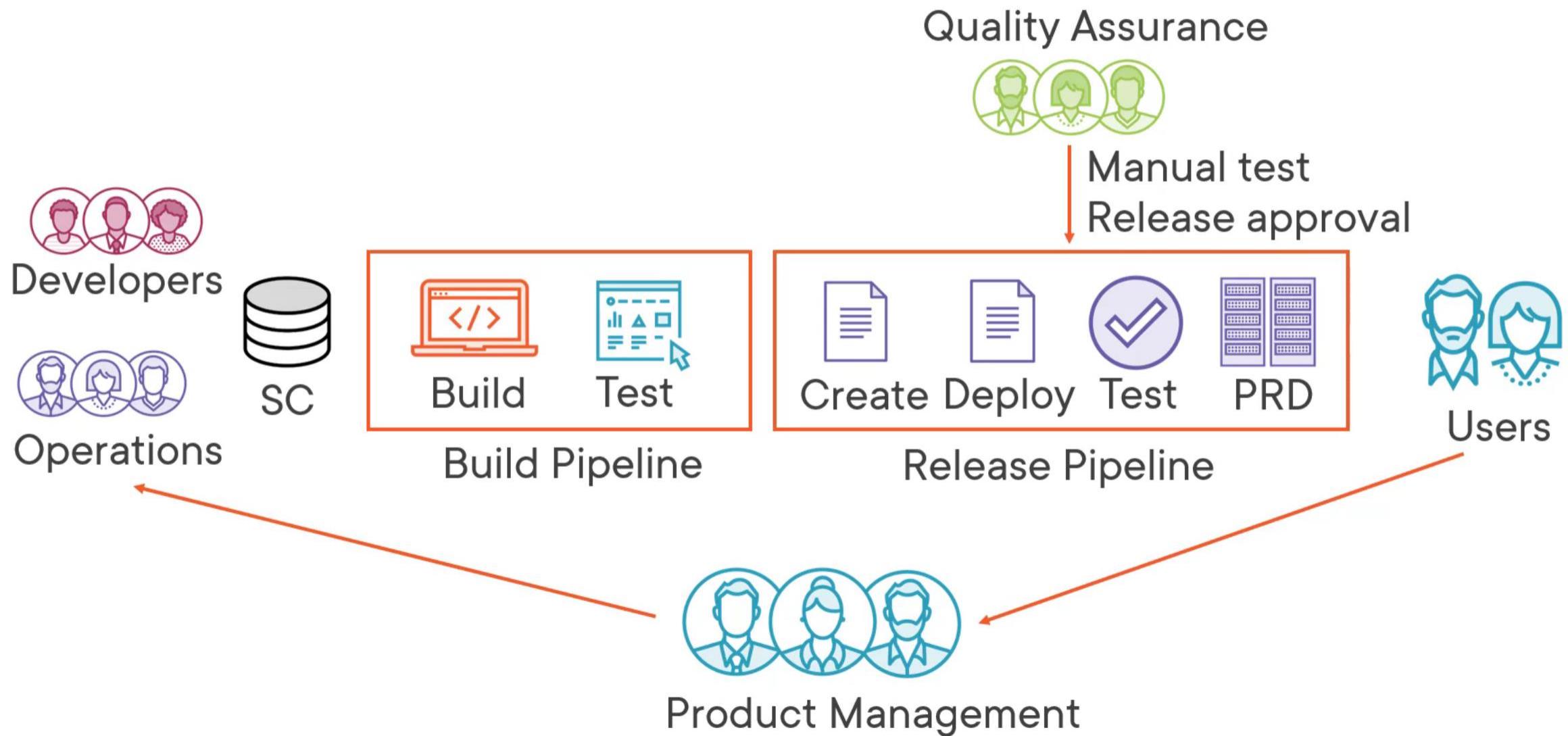


Continuous Delivery

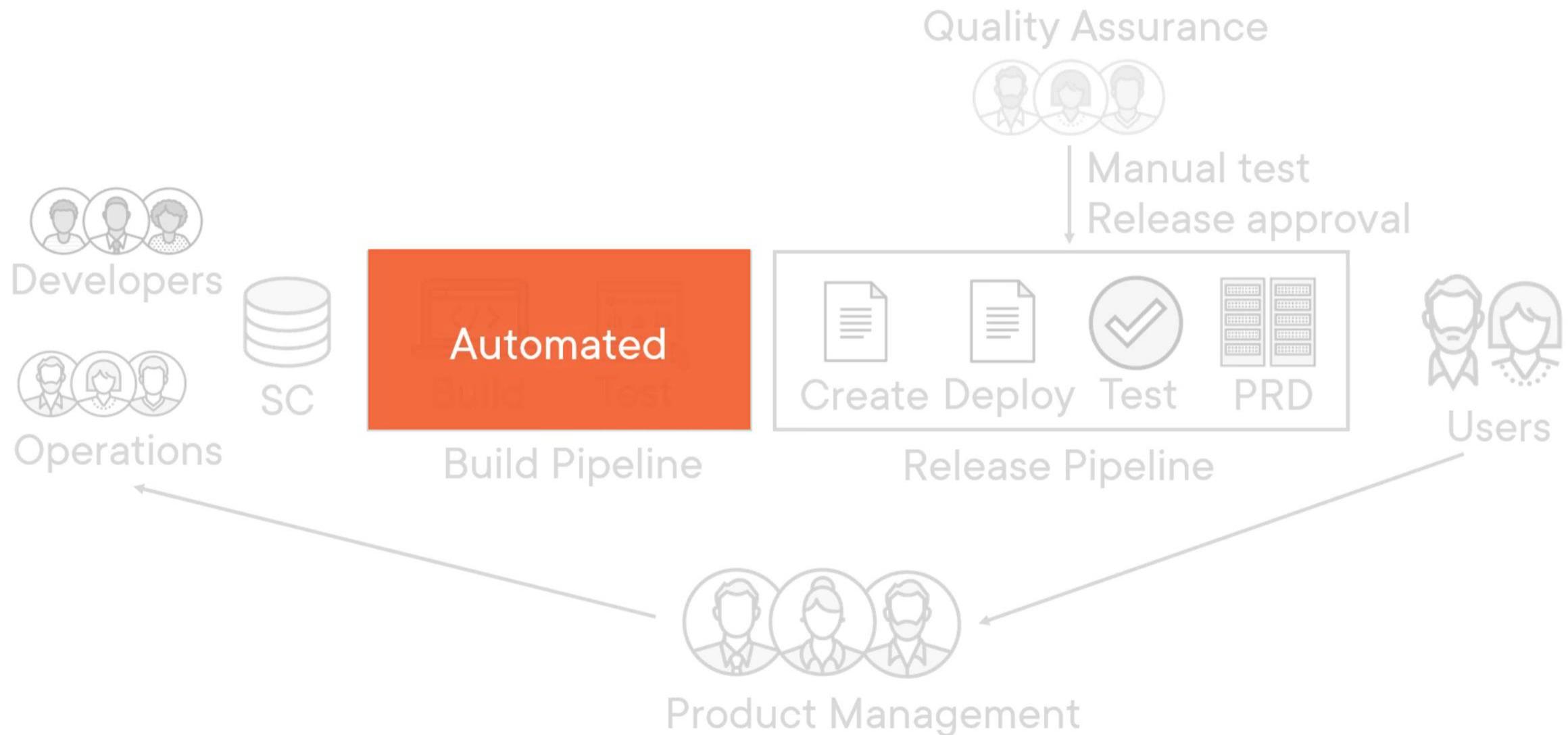


Pipeline

Continuous Delivery



Continuous Delivery





CD principles

- Have continuous integration in place**
- Development and Operations should work well together**
- Treat infrastructure as a code artifact**
- Automate the environment creation process**
- Automate the release process**
 - Automate acceptance tests
- Include release to definition of done**
- Releasing should be on-demand**
- Everyone has access to the latest result**
- Everyone can see everything**



CD benefits

Releasing takes less effort

Releasing is more

- Reliable
- Repeatable

Put control of release in the hands of business

Release more often

Get feedback earlier

What Can CD Accomplish?



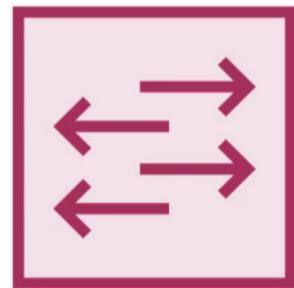
Higher Quality



Faster Delivery



Lower Costs



More Flexibility

DevOps Benefits

Speed

Rapid Delivery

Reliability

Scale

Improved Collaboration

Security

DevOps Stages

Version Control

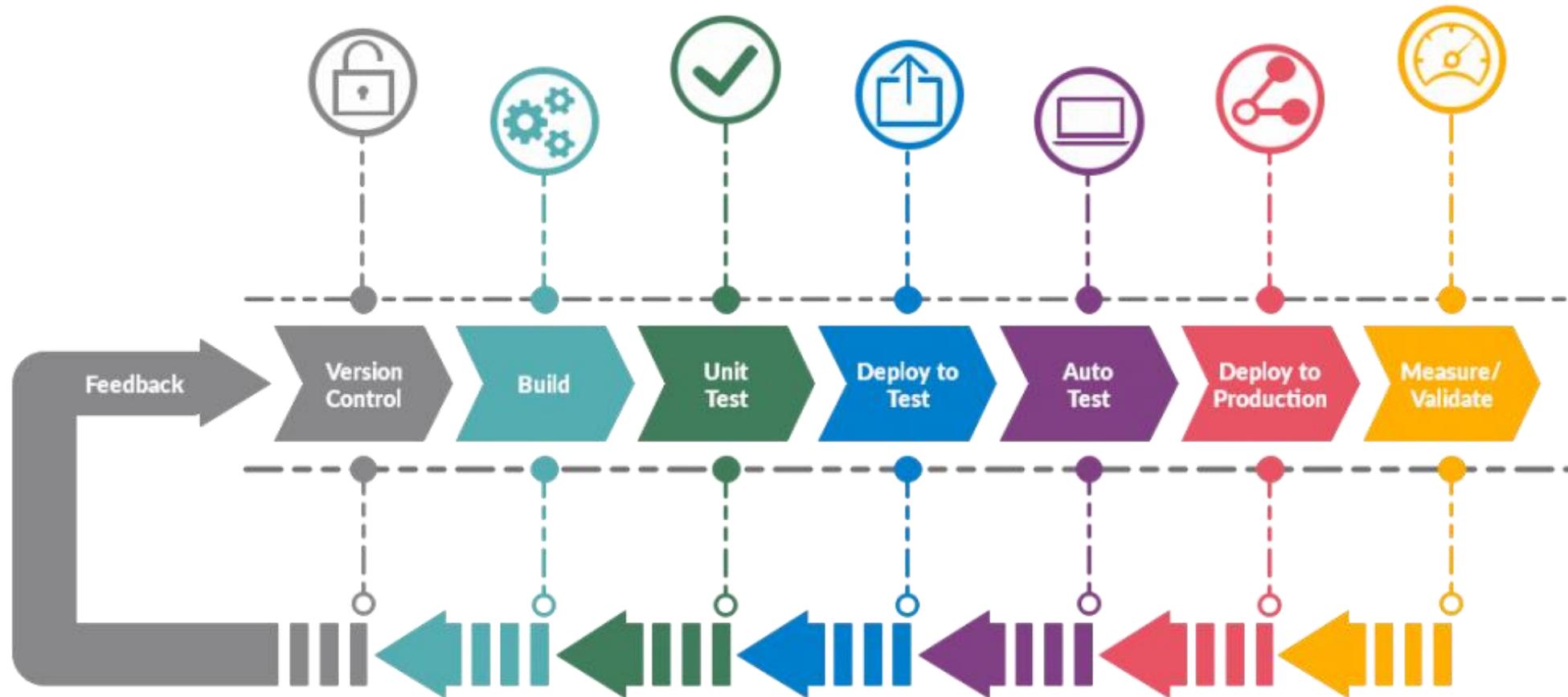
Continuous Integration

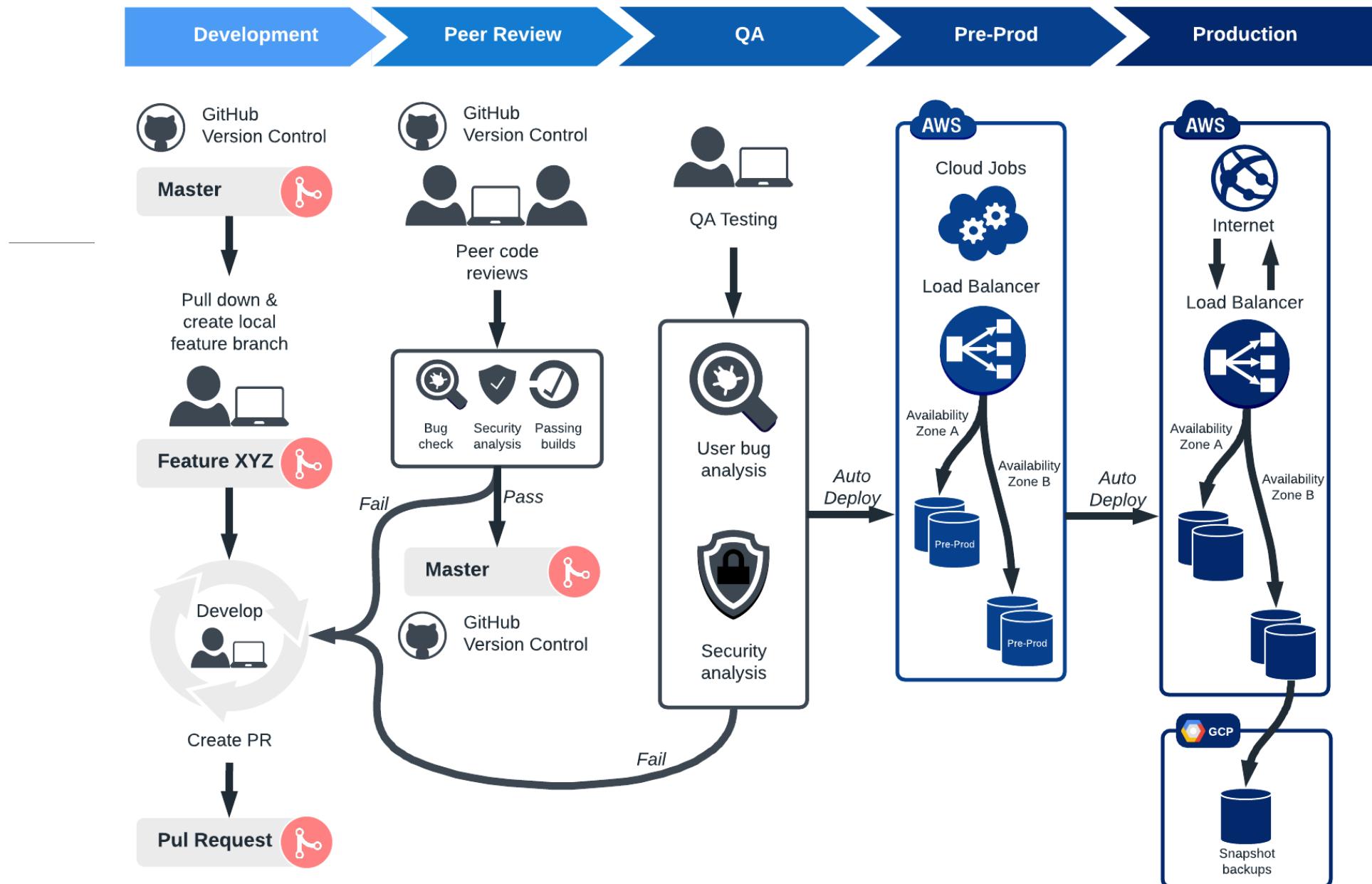
Continuous Delivery

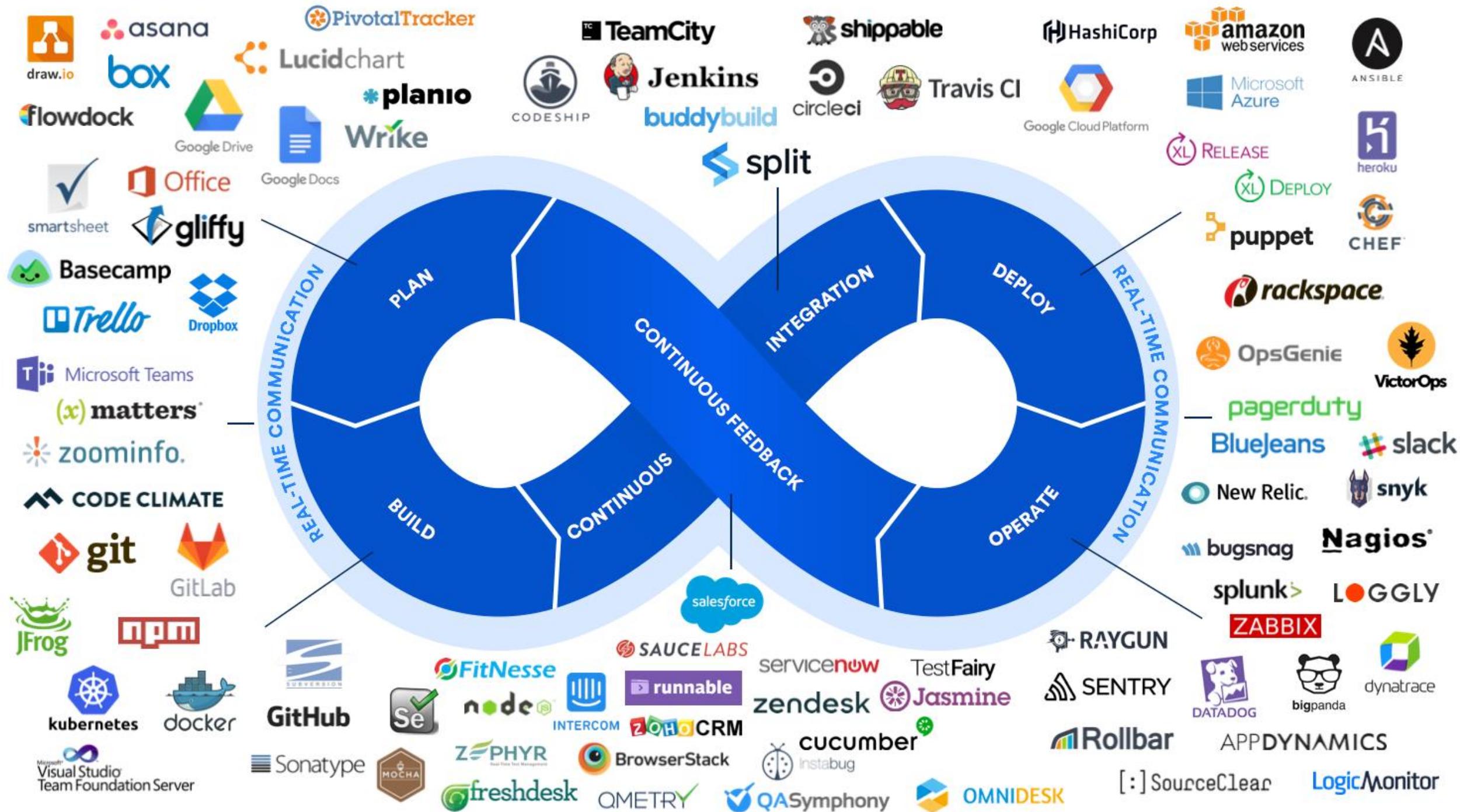
Continuous Deployment

Continuous Monitoring

DevOps Pipeline

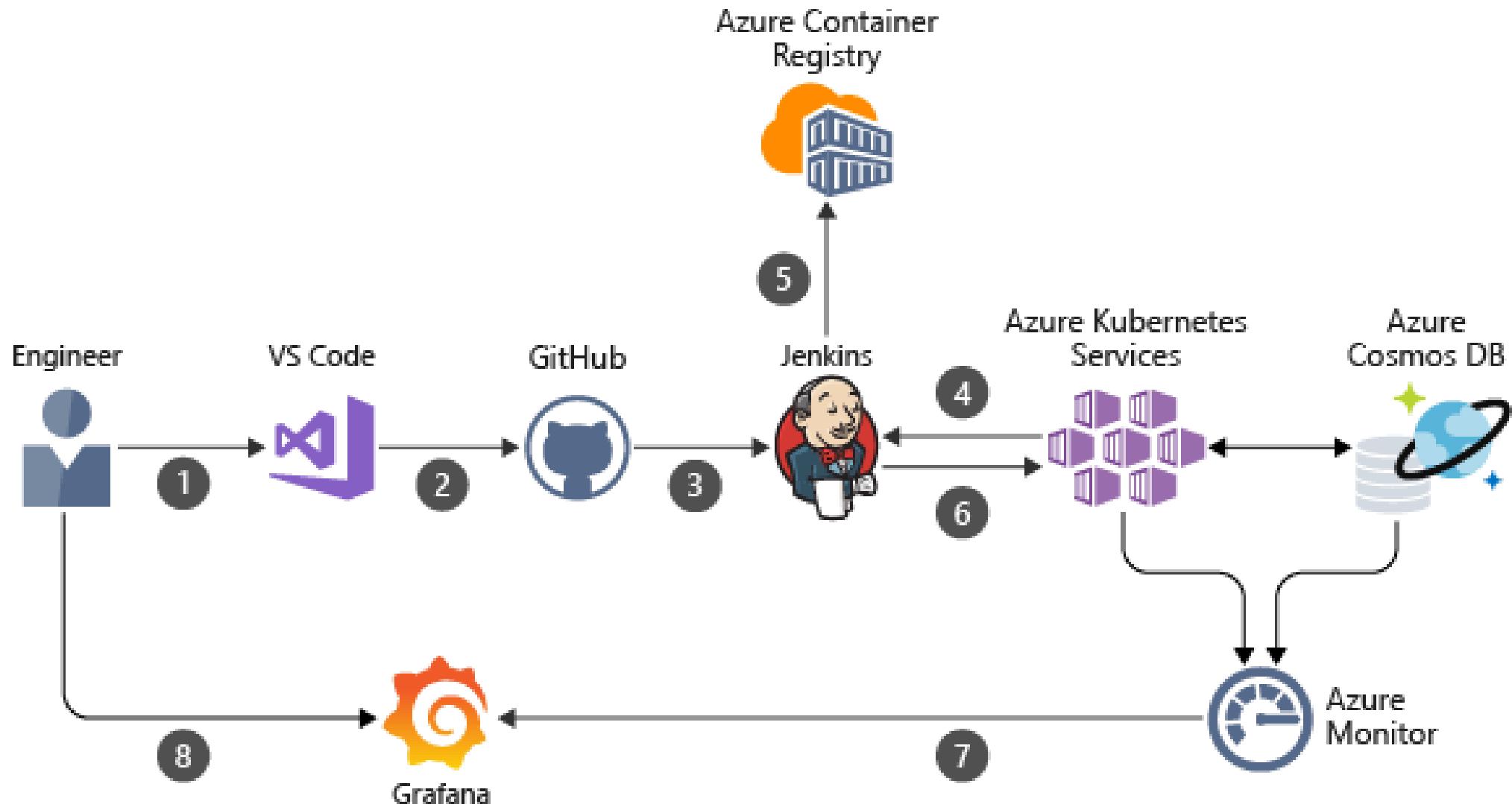






DevOps Tools





DevOps Best Practices

Incident
Command
System

Developers On
call

Public Status
Image

Blameless
Post-mortems

Embedded
Teams

Cloud

Andon Cords

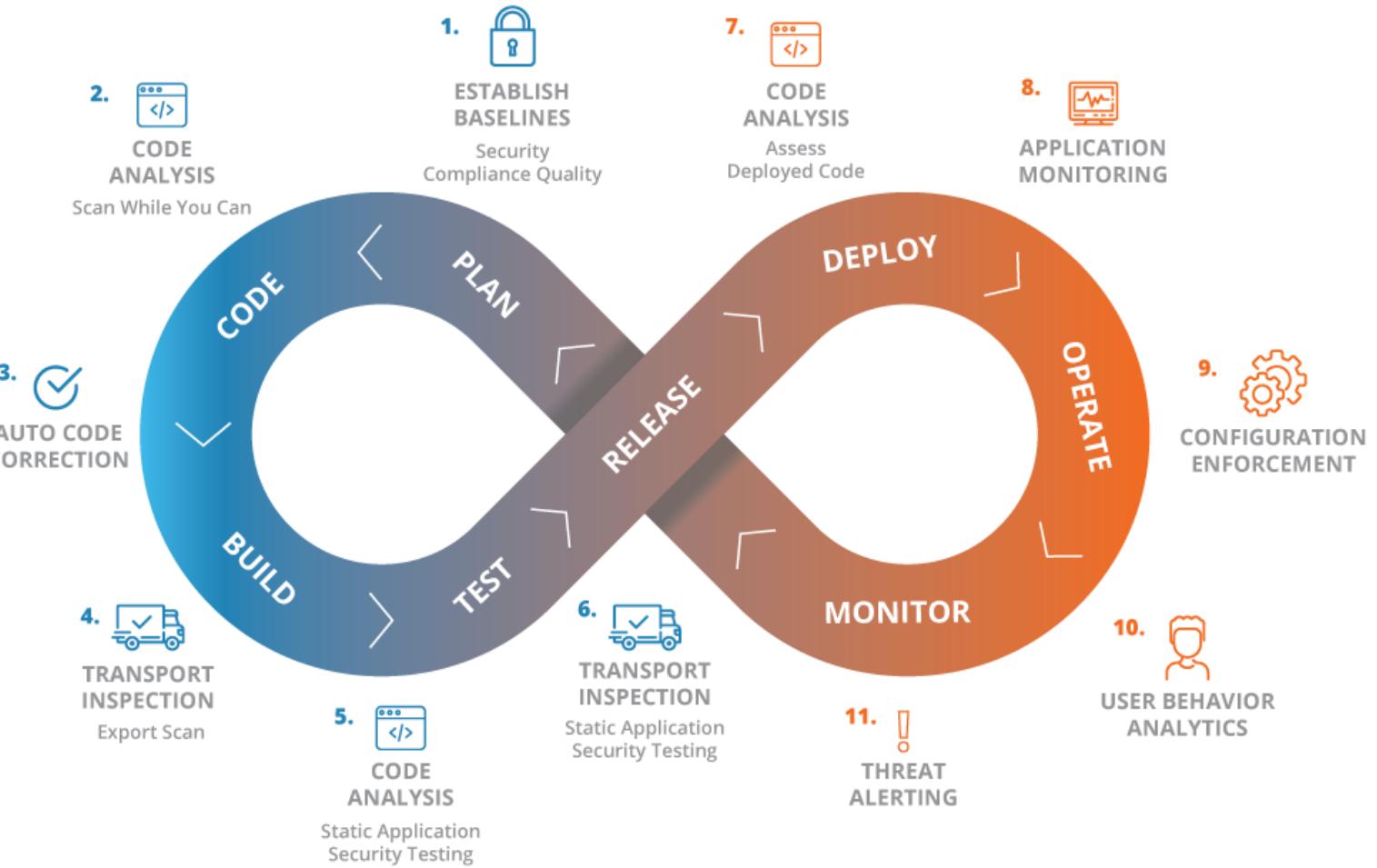
Dependency
Injection

Blue/Green
Deployment

Chaos Monkey

DEVSECOPS

DevOps and security teams have historically operated in silos, leaving huge gaps in application security and making apps more prone to attacks. DevSecOps presents a modern approach that embeds security throughout the application lifecycle to reduce risks.

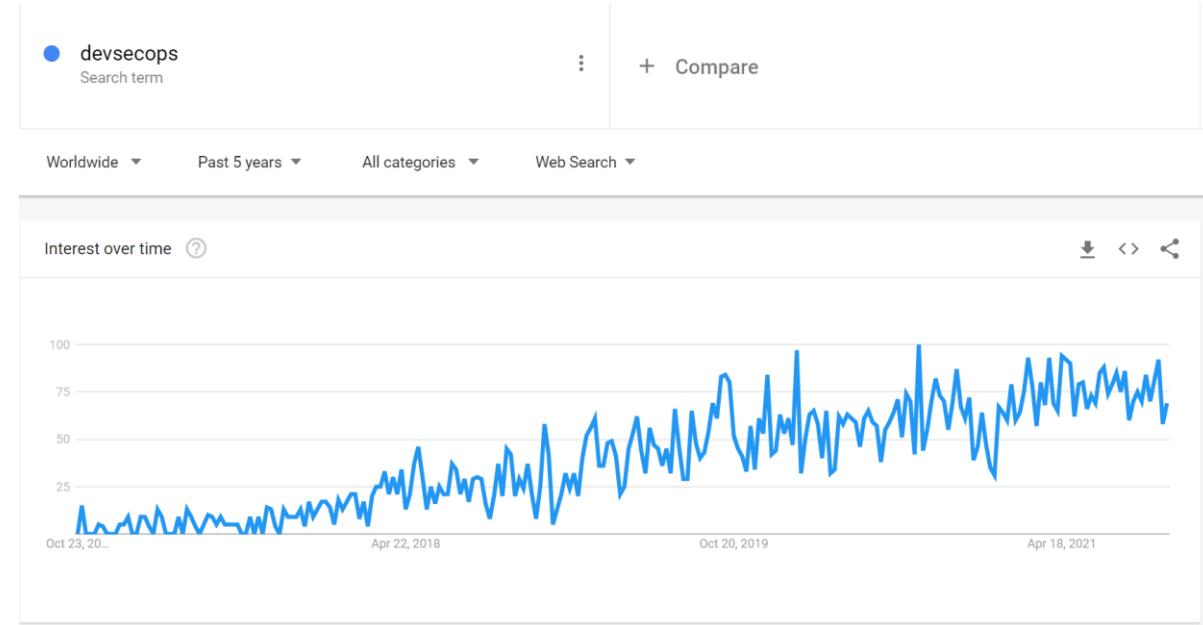




DevSecOps Concept

How to incorporate security within agile and DevOps Practices

The Purpose and intent of DevSecOps is to build on the mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required



<https://trends.google.com/trends/explore?date=today%205-y&q=devsecops>

DevSecOps Manifesto

Leaning in over Always Saying “No”

Data & Security Science over Fear, Uncertainty and Doubt

Open Contribution & Collaboration over Security-Only Requirements

Consumable Security Services with APIs over Mandated Security Controls & Paperwork

Business Driven Security Scores over Rubber Stamp Security

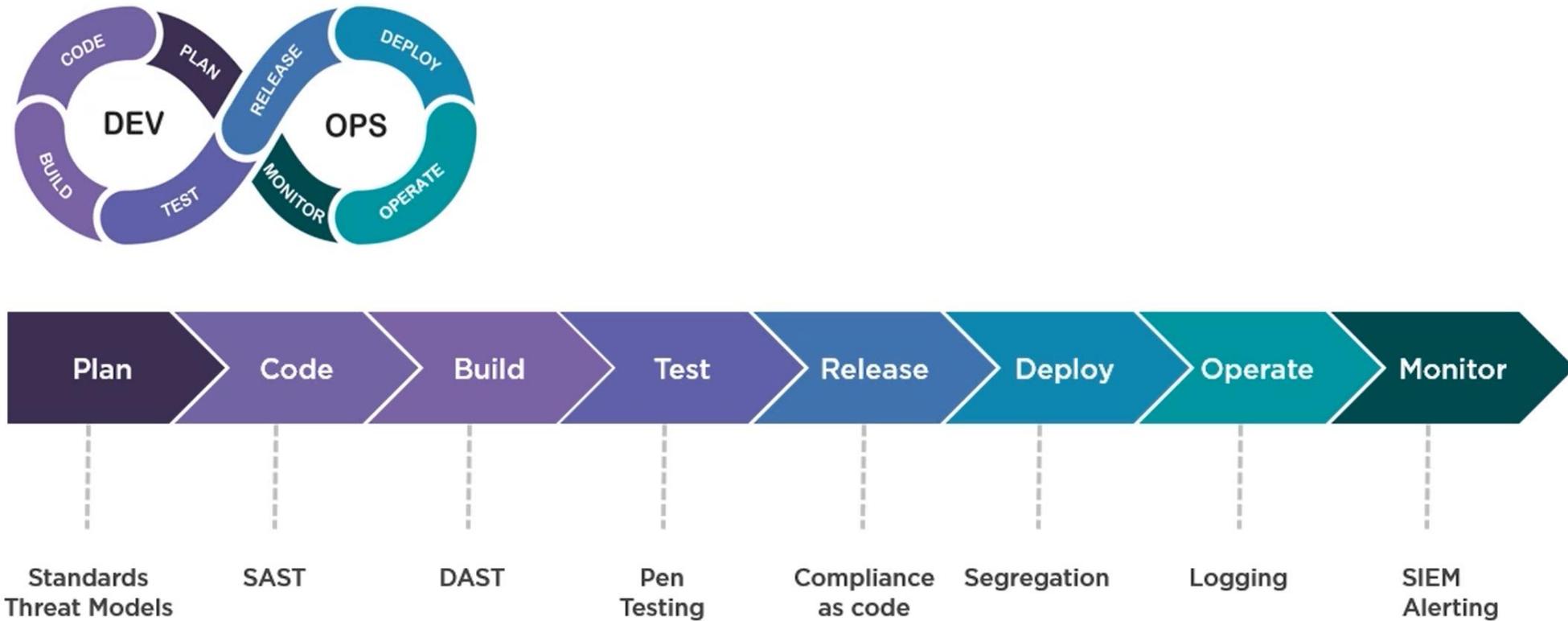
Red & Blue Team Exploit Testing over Relying on Scans & Theoretical Vulnerabilities

24x7 Proactive Security Monitoring over Reacting after being Informed of an Incident

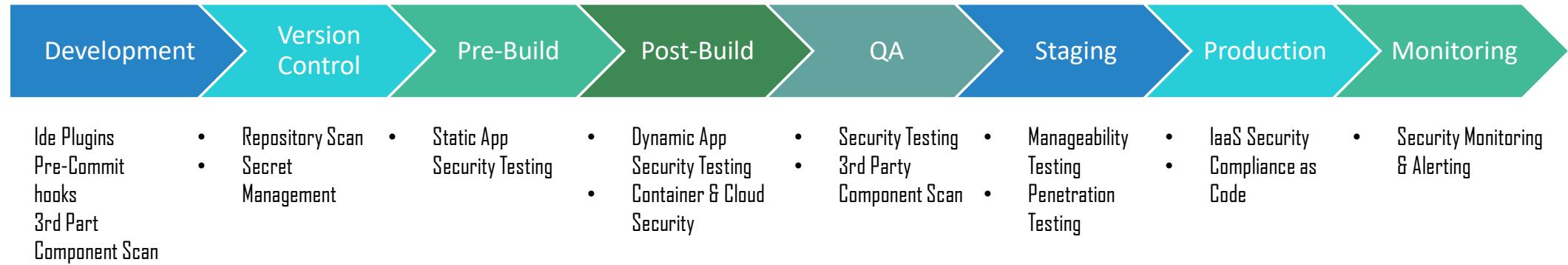
Shared Threat Intelligence over Keeping Info to Ourselves

Compliance Operations over Clipboards & Checklists

CI / CD Pipeline



Implementation of DevSecOps





cloudthat

Os
ci

1	En	Aja	Atlassian Jira Align	AiOps/Analytics	Continuous Integration	Security	2	Os
3	En	Daa	Digital.ai Agility	Artifact/Package Management	Database Management	Serverless/PaaS	Gi	Git
4	En	Tp	Targetprocess	Cloud	Deployment	Source Control Management		
5	En	Br	Broadcom Rally	Collaboration	Enterprise Agile Planning	Testing	Azp	Ow
11	En	Pv	Planview	Configuration Automation	Issue Tracking/ITSM	Value Stream Management	Dap	Dar
12	En			Containers	Release Management		Acp	Gh
19	Pd	In	Instana	Dd	Datadog	JFrog Artifactory	Ap	Br
20	En	Ja	Aws	Sl	Slack	AWS	Gl	Gls
21	En			Mt	Rha	Microsoft Teams	Ac	Ab
22	En			Ht	Dk	Red Hat Ansible	Ur	Al
23	En			Rho	Rho	HashiCorp Terraform	Ck	Abb
24	En			Lb	Dp	Red Hat OpenShift	Hv	
25	Os			Dp	Ud	Liquibase	Ur	
26	Os			Ud	Ha	Delphix	Al	
27	Os			Ha	Vc	UrbanCode Deploy	Ab	
28	En			Vc	Id	CyberArk Conjur		
29	Os			Id	De	HashiCorp Vault		
30	Fm			De	Ha	UrbanCode Release		
31	En			Ha	Vc	AWS Lambda		
32	En			Vc	Id	Atlassian Bitbucket		
33	Os			Id	Ha			
34	En			Ha	Vc			
35	En			Vc	Id			
36	Fm			Id	Ha			
37	En	Sp	Splunk	Ad	Snx	Sonatype Nexus	Ff	Azf
38	En			Ad	Az	Azure	Sr	Ci
39	Os			Snx	Gc	Google Cloud	Ff	
40	En			Az	Ac	Atlassian Confluence	Azf	
41	En			Gc	Ch	Chef	Ci	
42	En			Ac	Acf	AWS Cloud Formation		
43	Os			Ch	Ku	Kubernetes		
44	En			Acf	Ak	Amazon EKS		
45	Os			Ku	De	Docker Enterprise		
46	En			Ak	Id	Idera		
47	En			De	Ha	Harness		
48	En			Id	Vc	Veracode		
49	En			Ha	Id	SonarQube		
50	En			Vc	Ha	Micro Focus Fortify SCA		
51	Os			Id	Ha	Azure Functions		
52	En			Ha	Vc	Compuware ISPW		
53	En			Vc	Id			
54	En			Id	Ha			
55	En	Dt	Dynatrace	Nr	Dh	Docker Hub	Fm	Ci
56	En			Nr	Np	npm		
57	Fm			Dh	Ic	IBM Cloud		
58	En			Np	So	Stack Overflow		
59	En			Ic	Pu	Puppet		
60	En			So	Hc	HashiCorp Consul		
61	Fm			Pu	Ae	Amazon ECS		
62	Os			Hc	Azk	Azure AKS		
63	En			Ae	Ra	Rancher		
64	En			Azk	Qt	Quest Toad		
65	Os			Ra	Sk	Spinnaker		
66	Fm			Qt	Od	Octopus Deploy		
67	Os			Sk	Od	Synopsys Black Duck		
68	En			Od	Sb	Checkmarx SAST		
69	En			Sb	Cx	Heroku		
70	En			Cx	He	Subversion		
71	Fm			He	Sv			
72	Os			Sv				
73	Os	Gr	Grafana	EI	Yn	Yarn		
74	Os			EI	Nu	NuGet		
75	Os			Yn	Os	OpenStack		
76	Os			Nu	Mm	Mattermost		
77	Os			Os	Sa	Salt		
78	Os			Mm	Hg	HashiCorp Vagrant		
79	Os			Sa	Hp	HashiCorp Packer		
80	Os			Hg	Gk	Google GKE		
81	Os			Hp	Hm	helm		
82	En			Gk	Db	DBMaestro		
83	Os			Hm	Cfd	Cloudbees CD		
84	En			Db	Acd	AWS CodeDeploy		
85	En			Cfd	Ac			
86	En			Ac	Sn	Snort		
87	Os			Sn	Pbs	PortSwigger Burp Suite		
88	Fm			Pbs	Gf	Google Firebase		
89	En			Gf	Cf	Cloud Foundry		
90	Os			Cf				

Os Open-source

Fr Free

Freemium

Paid

Enterprise

