<epam>

DevOps education program

What is DevOps?

Lecture 1.1

Module 1. DevOps Introduction

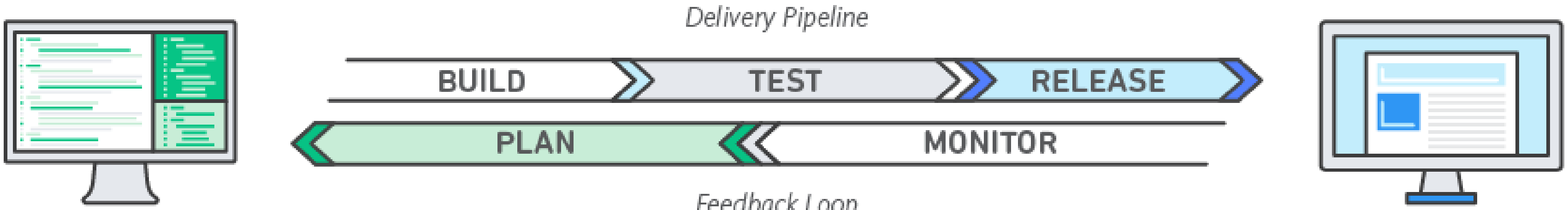
Andrii Kostromytskyi

Agenda

- What is DevOps?
- DevOps tools
- Q&A

WHAT IS DEVOPS?

Customer interaction

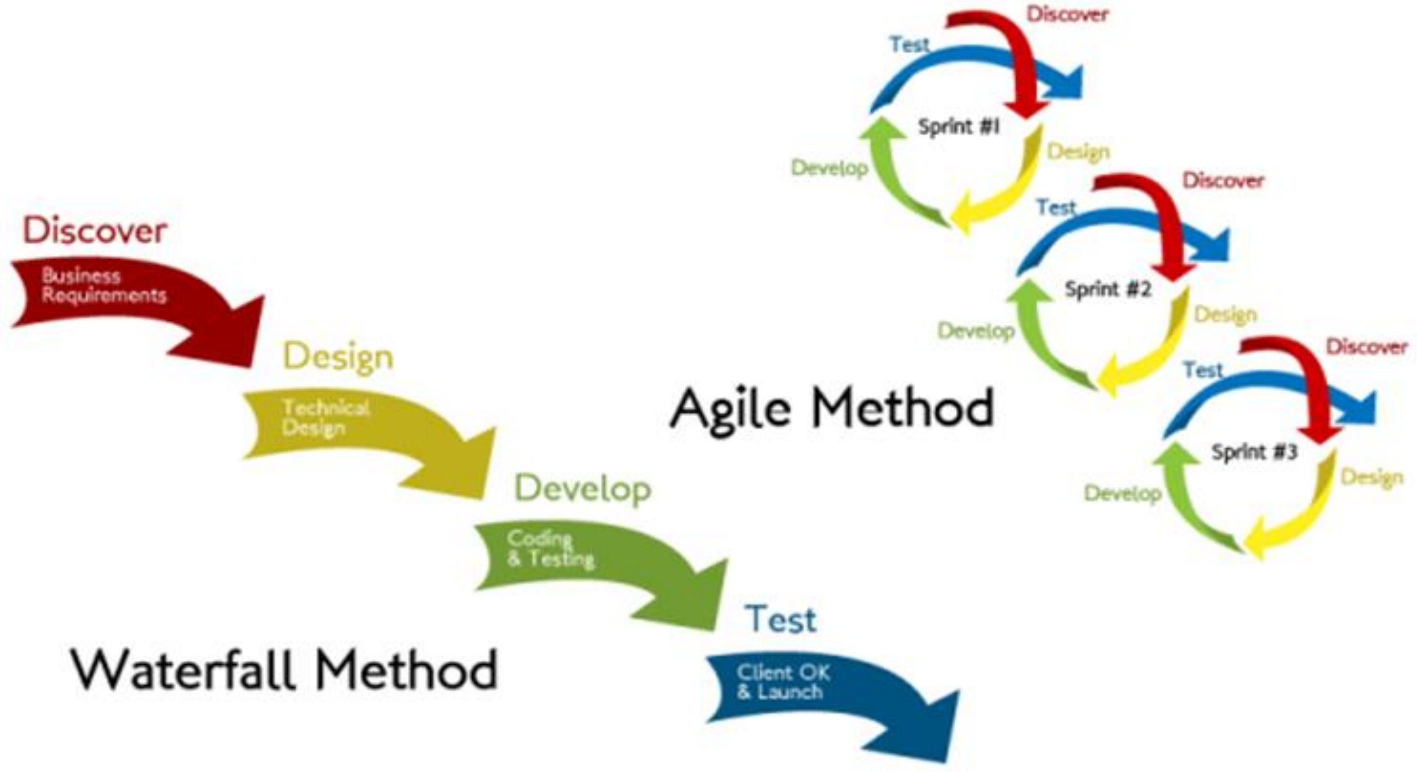


Software Development

Customer

- New feature requests
- Issues & enhancement

Development methodologies



Waterfall vs Agile

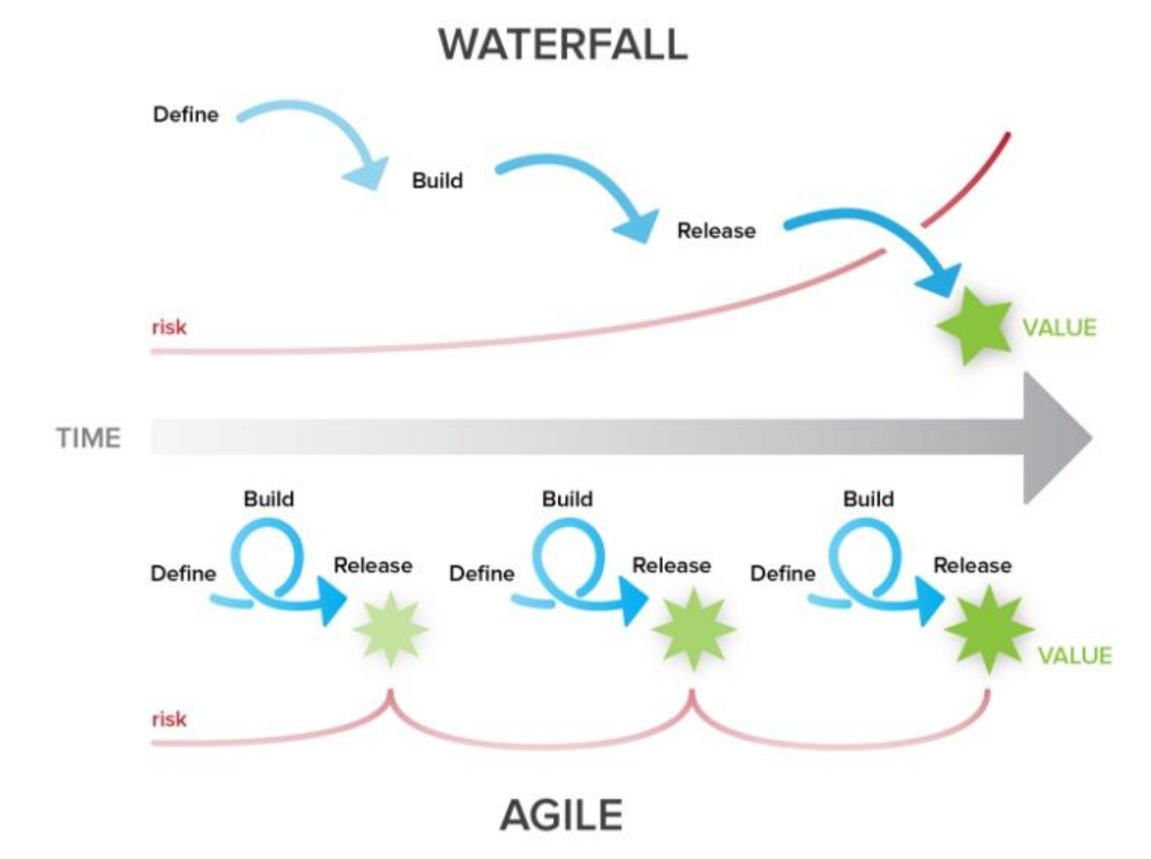
WHEN WATERFALL?

- Only when the requirements are known, understandable and fixed. There are no conflicting requirements.
- There are no problems with the availability of programmers of the right qualifications.
- In relatively small projects.

WHEN AGILE?

- When user needs are constantly changing in a dynamic business.
- Agile changes are implemented at a lower cost due to frequent increments.
- Unlike the waterfall model, in a flexible model, just a little planning is enough to start the project.

Waterfall vs Agile

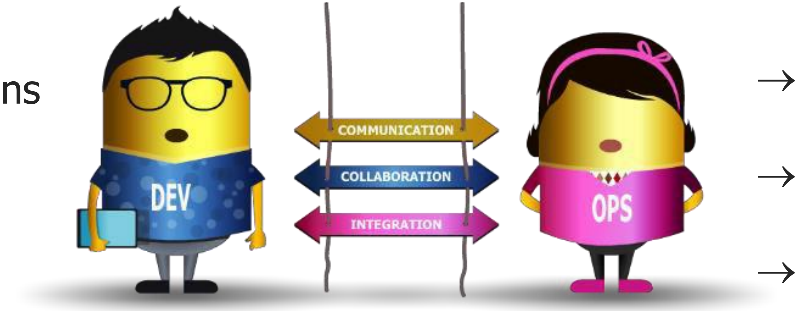


Development & Operation

Roles of Developers :

→ Develop/modify applications

→ Try new technologies



Roles of Operations :

→ Build, Deploy and release

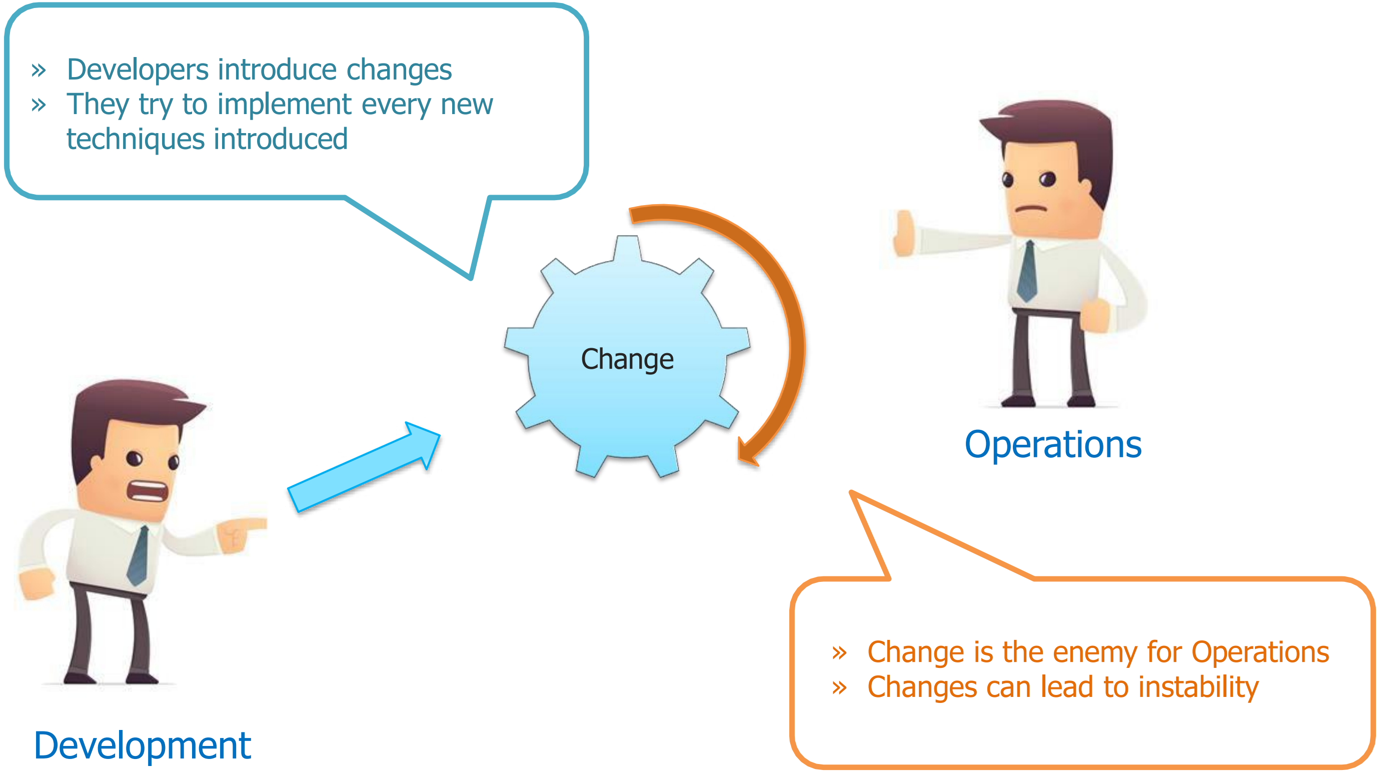
→ Performance and availability

→ Create or enhance services

“Dev” is used as a shorthand for developers in particular, but in practice it is even wider and it means that “all the people involved in developing the product”, that include the product, QA and other disciplines.

“Ops” is a blanket term for Systems engineers/ Administrators, Release engineers, DBAs, Network engineers, Security professionals and various other sub-disciplines and job titles

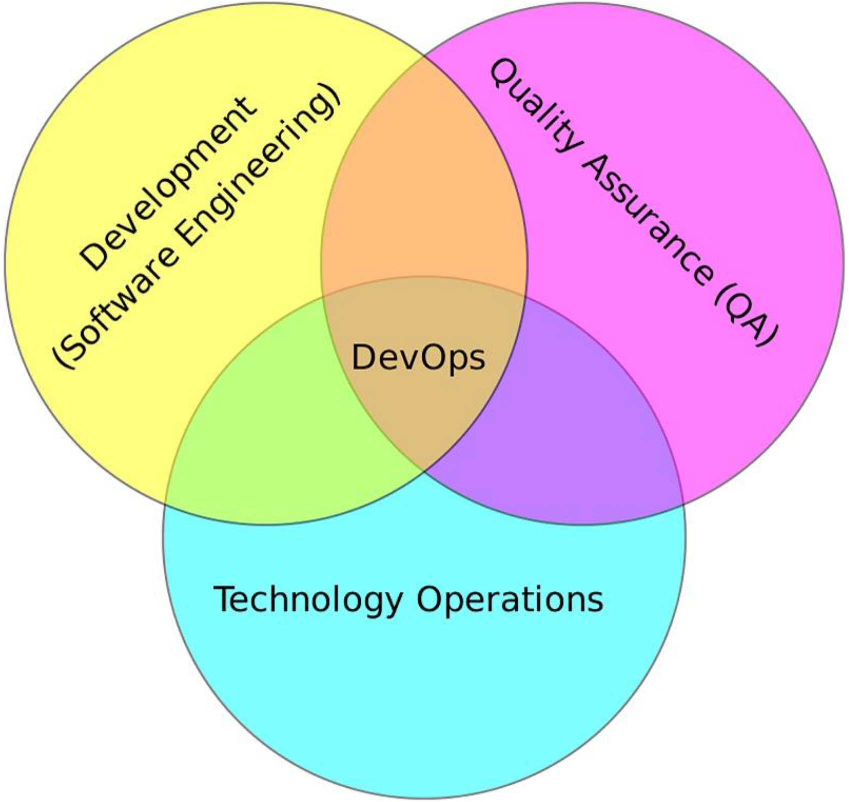
Development vs Operations



- Developers introduce changes
- They try to implement every new techniques introduced

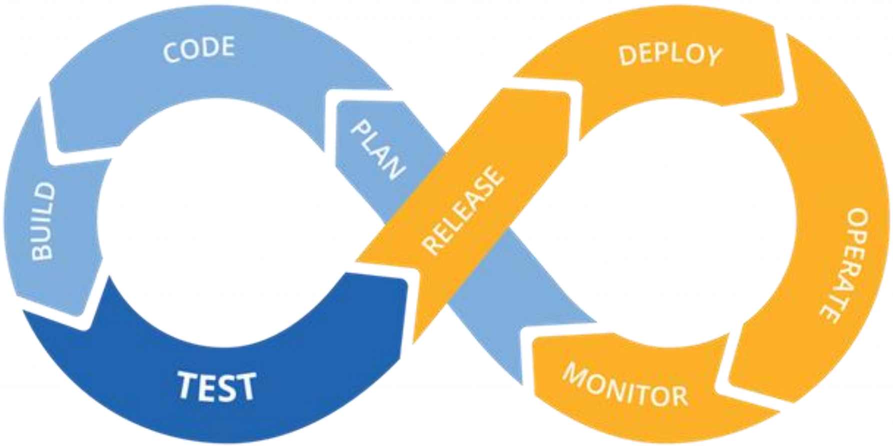
- Change is the enemy for Operations
- Changes can lead to instability

DevOps



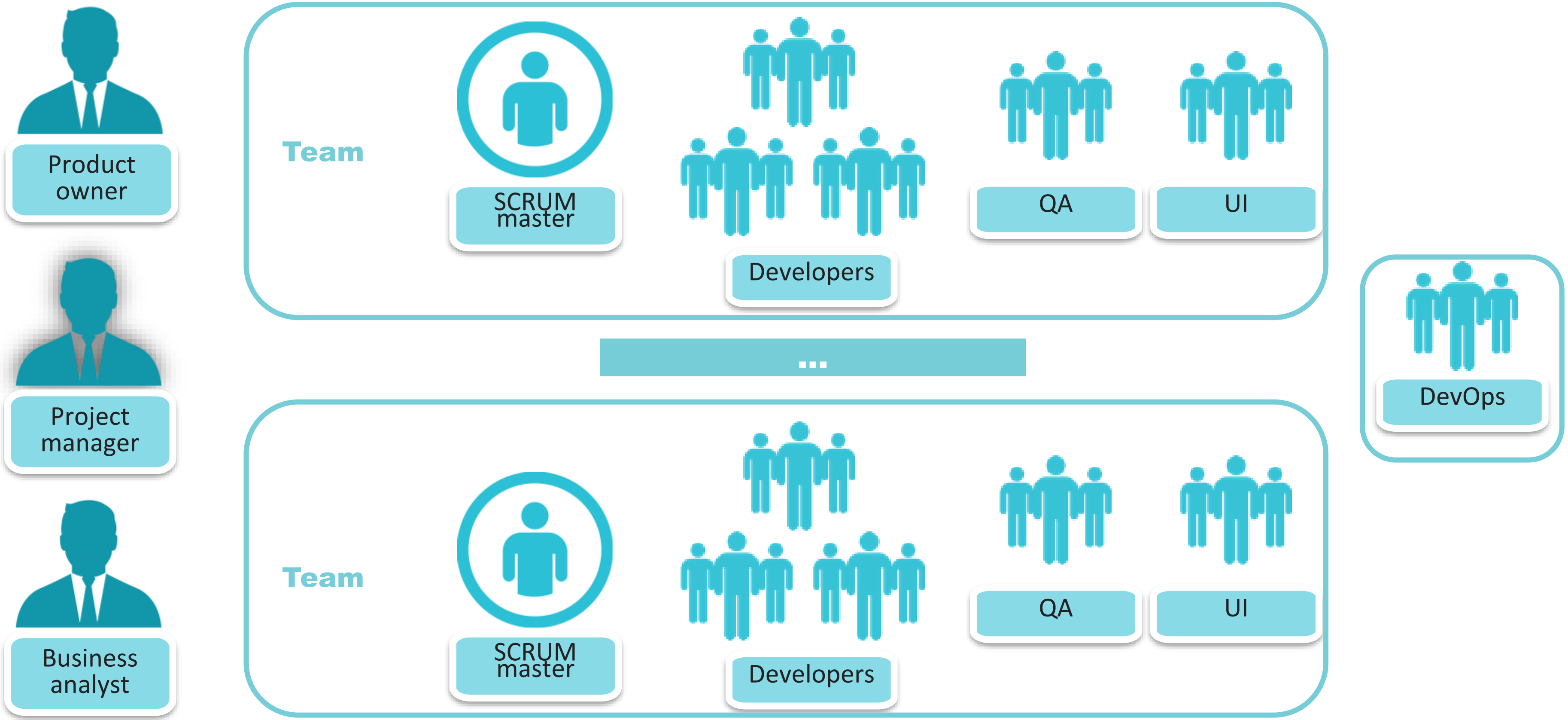
DEVelopment + **OP**erations

2009

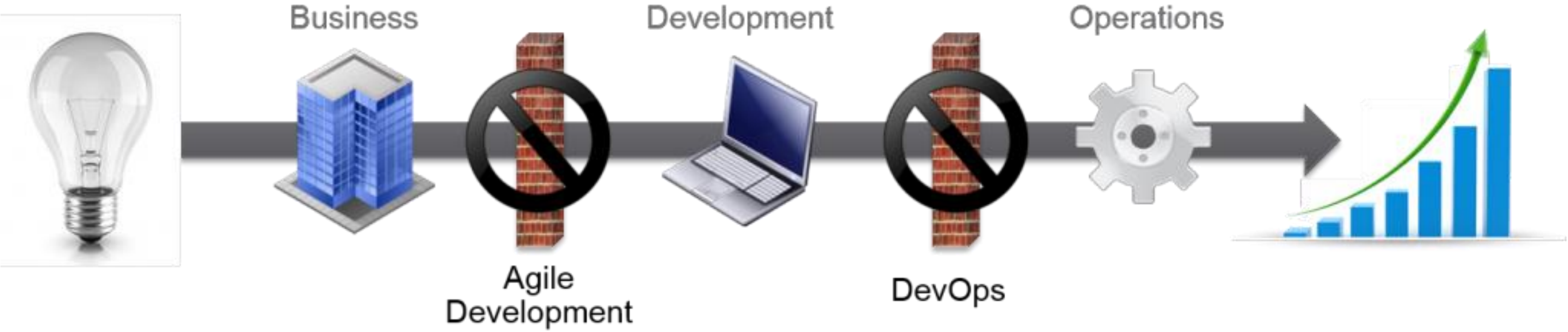


SDLC - Software Development Life Cycle

Typical Project Team



Main goal of implementing a DevOps approach



Main goal of implementing a DevOps approach

From planning through delivery, the goal of DevOps is to improve collaboration across the value stream by developing and automating a continuous delivery pipeline. In doing so, DevOps:

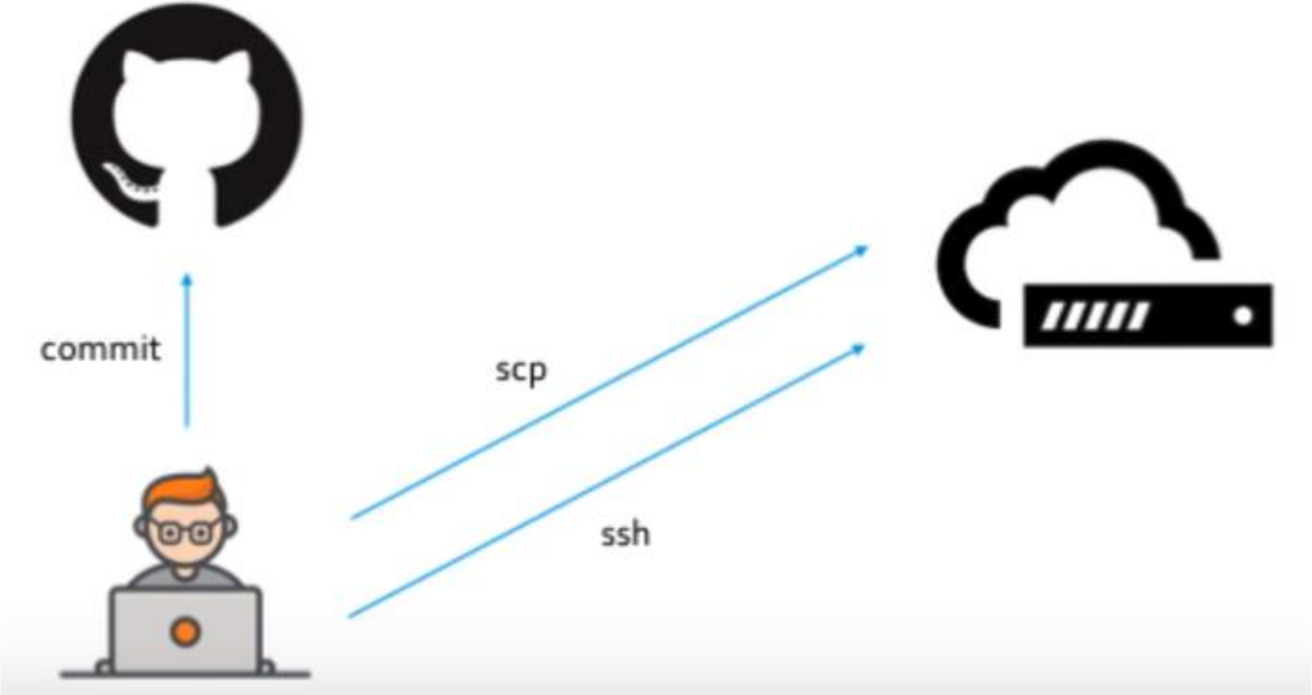
- Increases the frequency and quality of deployments;
- Improves innovation and risk-taking by making it safer to experiment;
- Realizes faster time to market;
- Improves solution quality and shortens the lead time for fixes;
- Reduces the severity and frequency of release failures;
- Improves the Mean Time to Recovery (MTTR).

Common DevOps roles and responsibilities

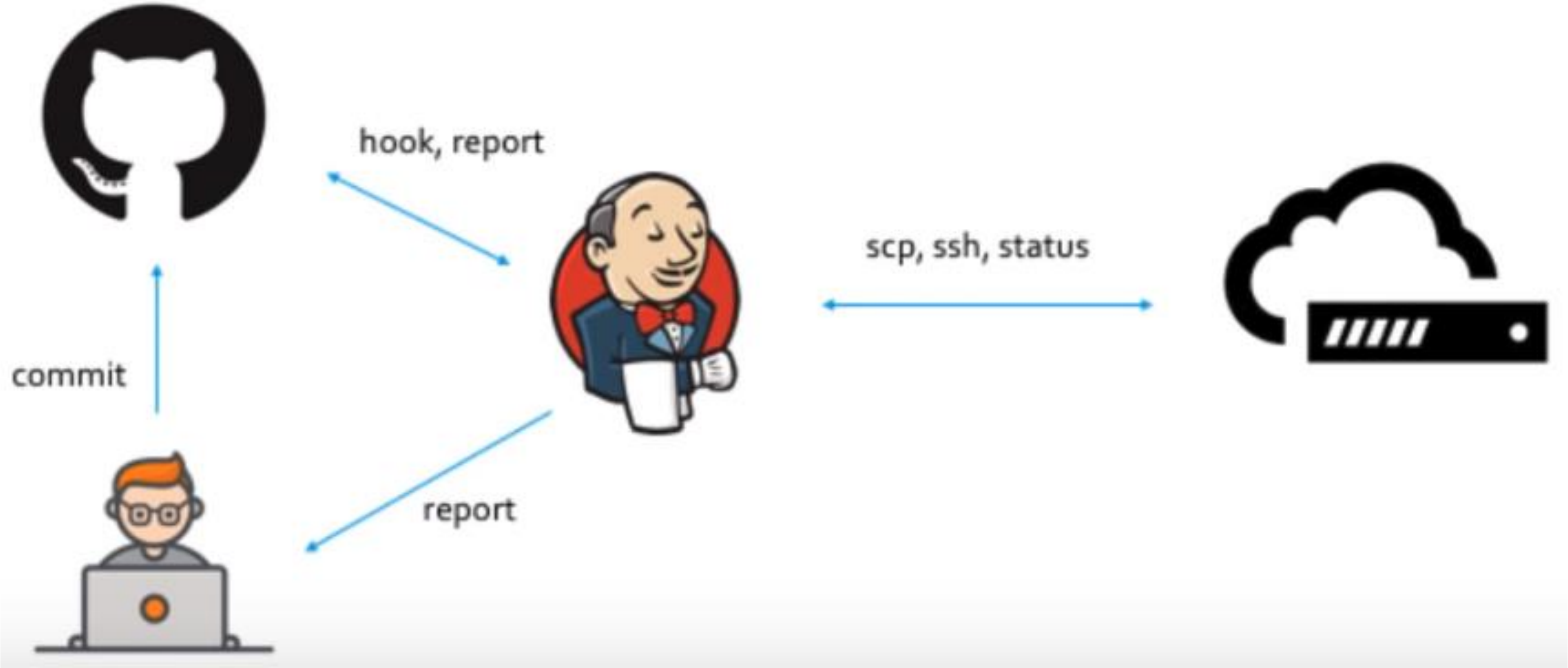
- Application and infrastructure planning, testing and development
- Maintaining CI/CD pipelines
- Automation implementation
- On-call, incident response and incident management
- Monitoring

- Integration Engineer
- Cloud Specialist
- Automation Engineer
- Release Engineer
- Security Engineer

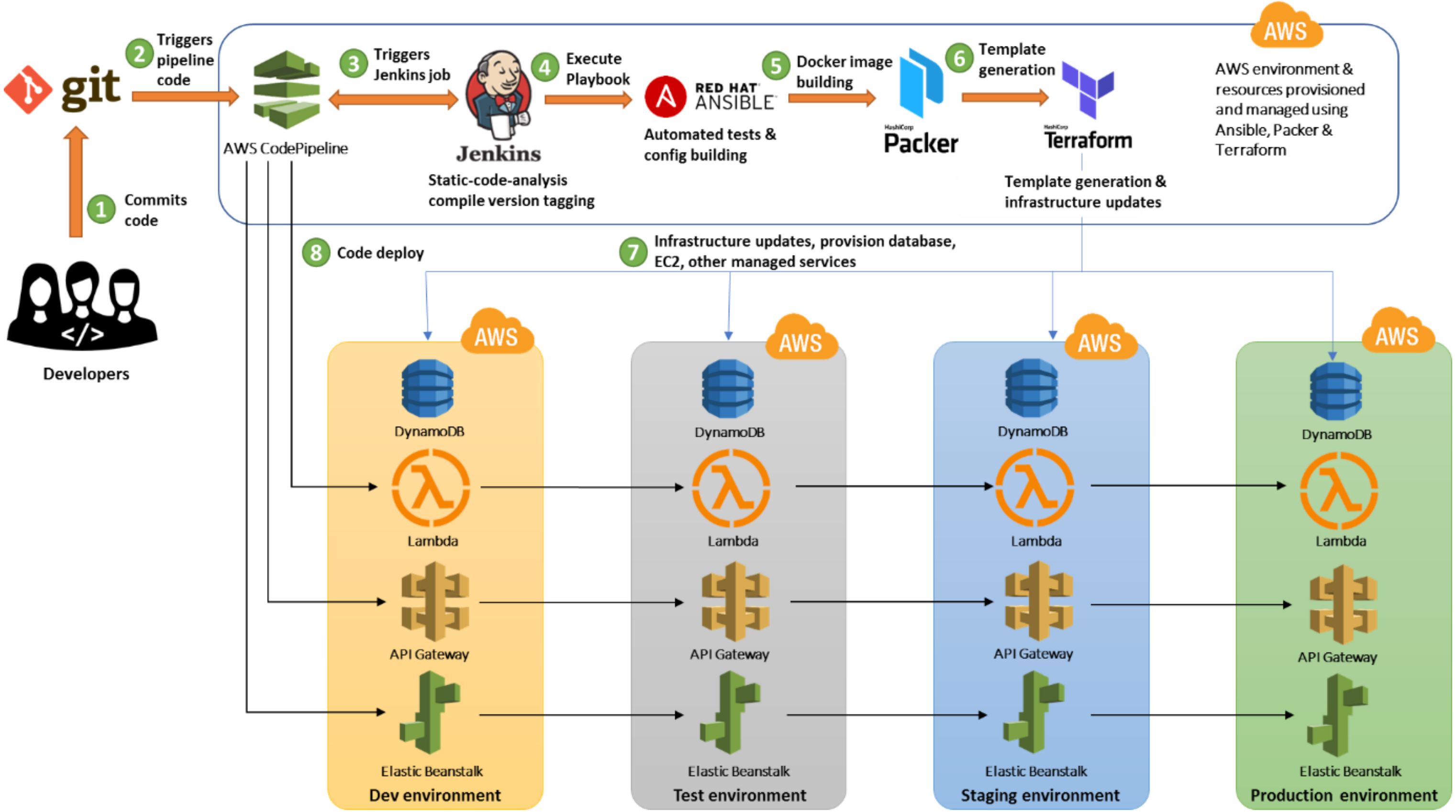
Manual deploy



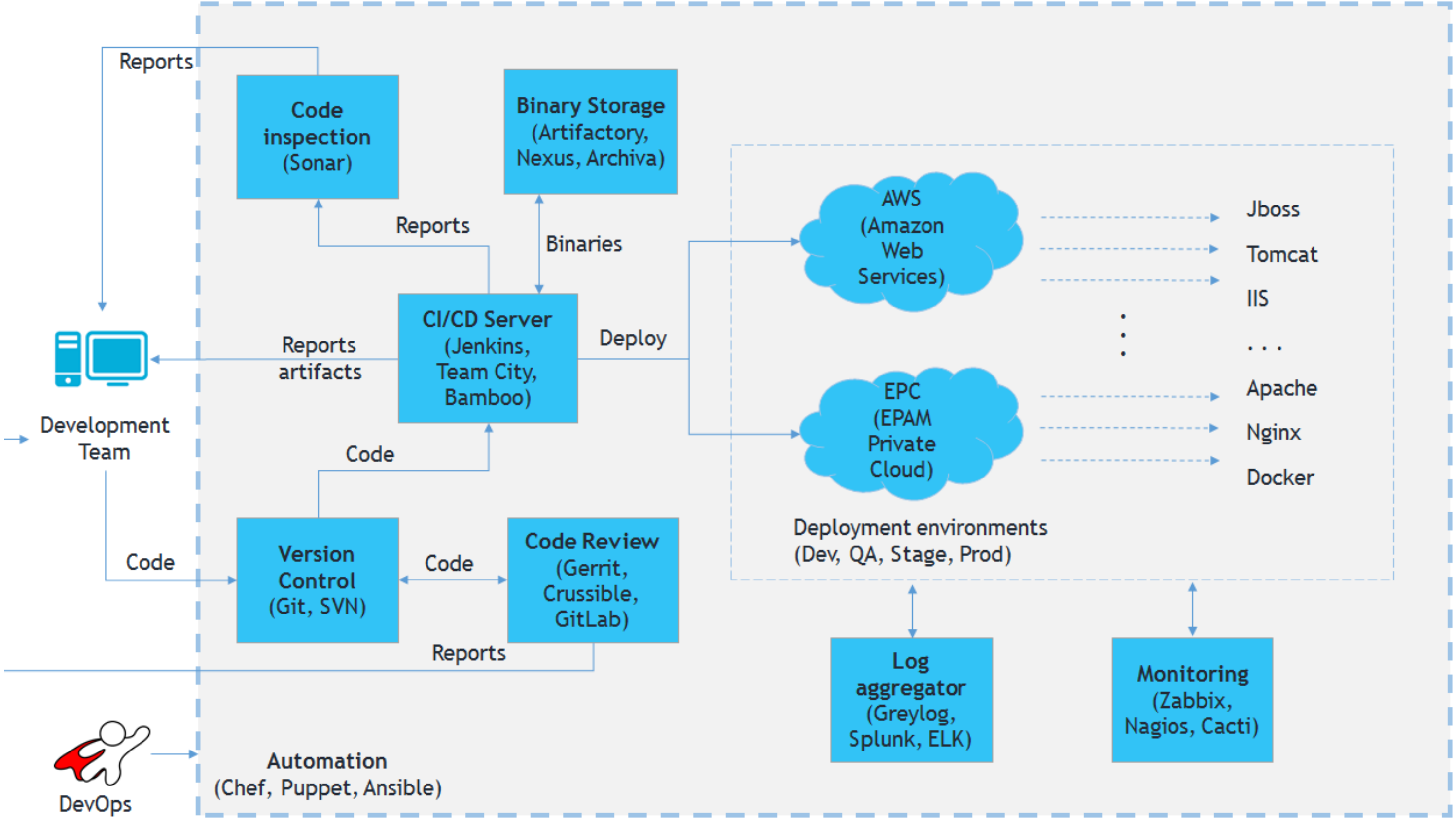
Automation deploy



Environment Design

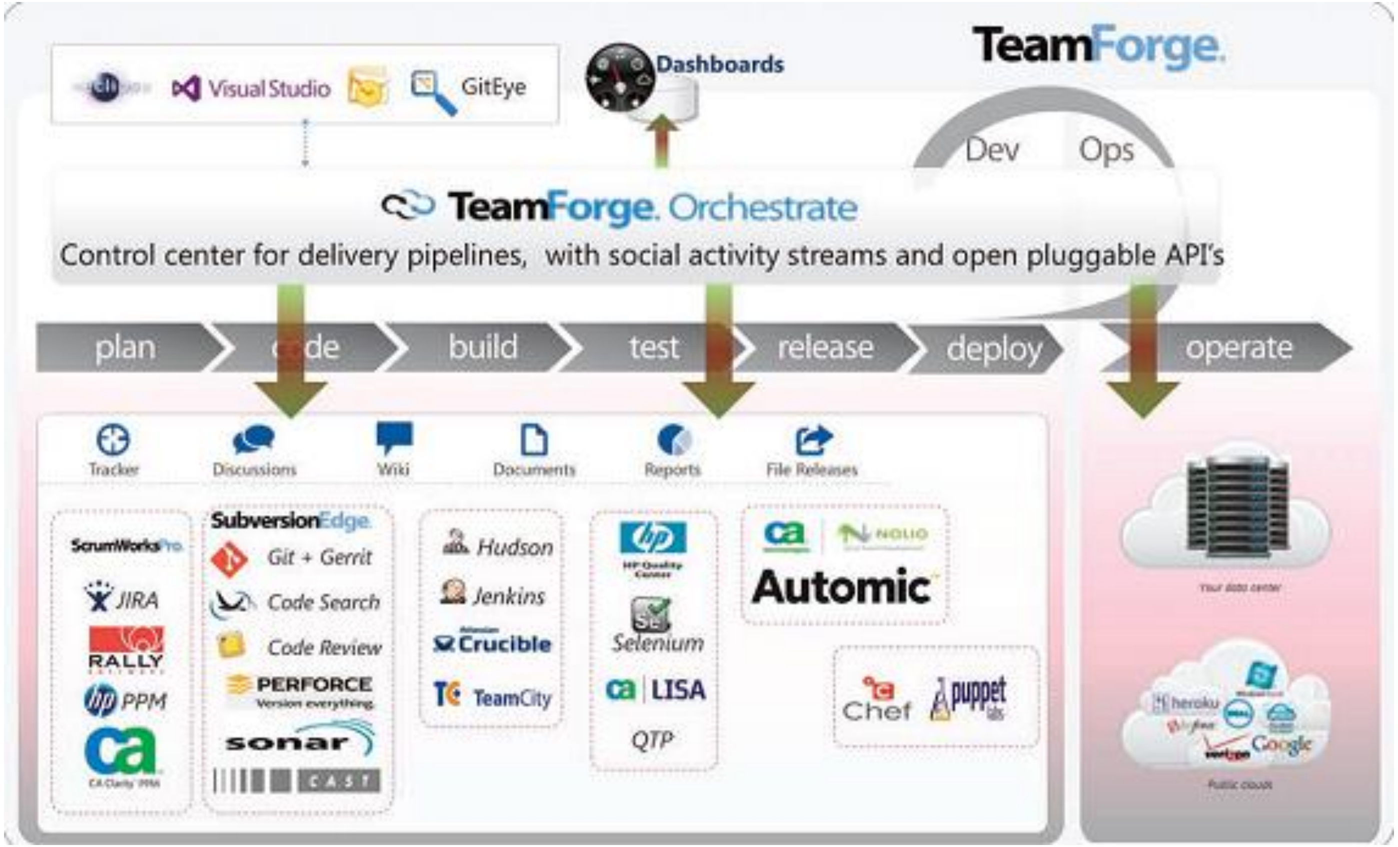


DevOps responsibility

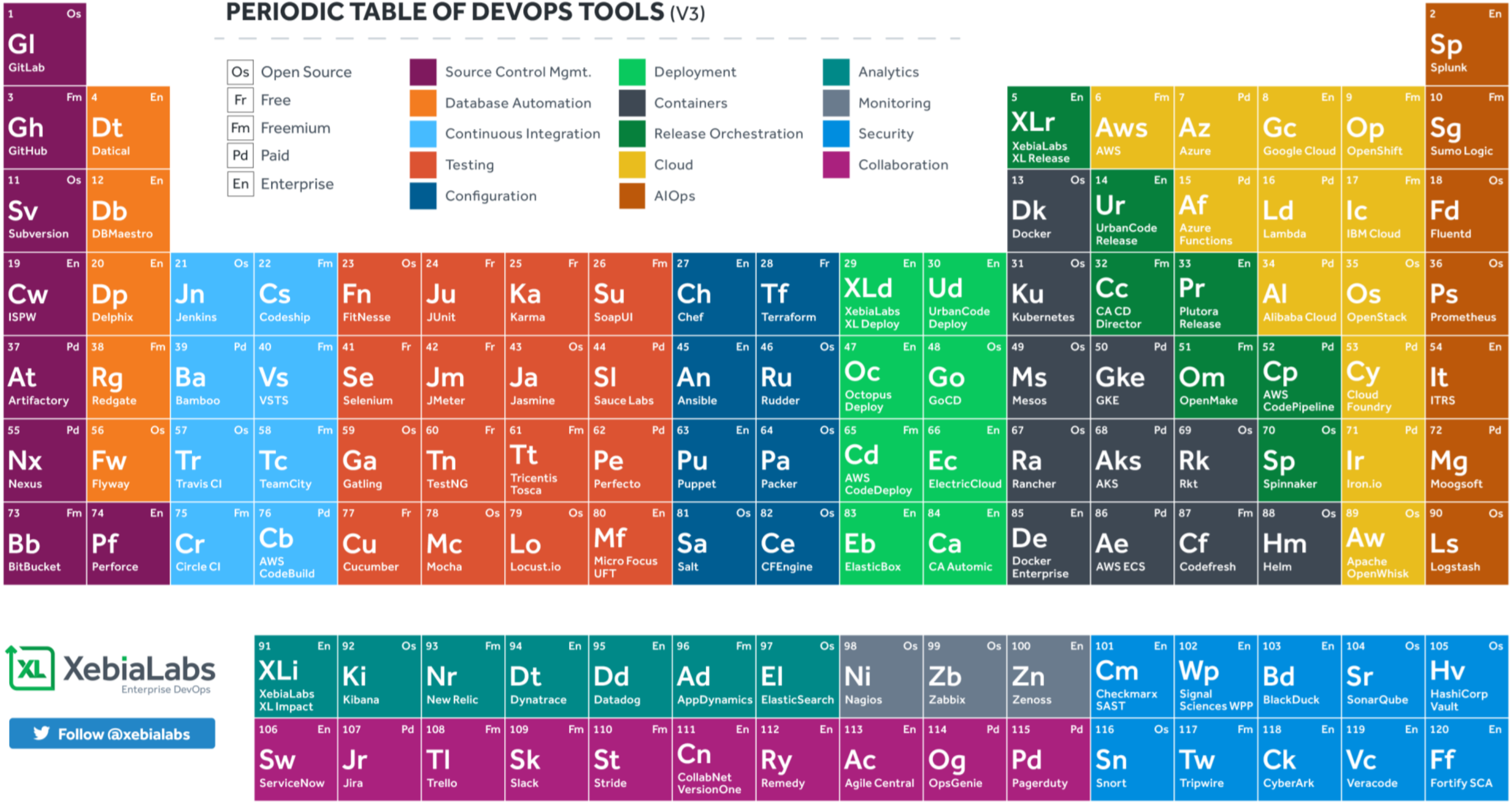


Reports
Code inspection (Sonar)
Binary Storage (Artifactory, Nexus, Archiva)
Reports
Binaries
AWS (Amazon Web Services)
Jboss
Tomcat
IIS
. . .
Apache
Nginx
Docker
EPC (EPAM Private Cloud)
CI/CD Server (Jenkins, Team City, Bamboo)
Deploy
Reports artifacts
Development Team
Code
Version Control (Git, SVN)
Code
Code Review (Gerrit, Crussible, GitLab)
Code
Reports
Deployment environments (Dev, QA, Stage, Prod)
Log aggregator (Greylog, Splunk, ELK)
Monitoring (Zabbix, Nagios, Cacti)
DevOps
Automation (Chef, Puppet, Ansible)

Application Lifecycle Manager (ALM)



PERIODIC TABLE OF DEVOPS TOOLS (V3)



Os	Open Source
Fr	Free
Fm	Freemium
Pd	Paid
En	Enterprise

- 🟪 Source Control Mgmt.
- 🟧 Database Automation
- 🟦 Continuous Integration
- 🟥 Testing
- 🟦 Configuration
- 🟩 Deployment
- ⬛ Containers
- 🟩 Release Orchestration
- 🟨 Cloud
- 🟫 AIOps
- 🟦 Analytics
- ⬜ Monitoring
- 🟦 Security
- 🟪 Collaboration

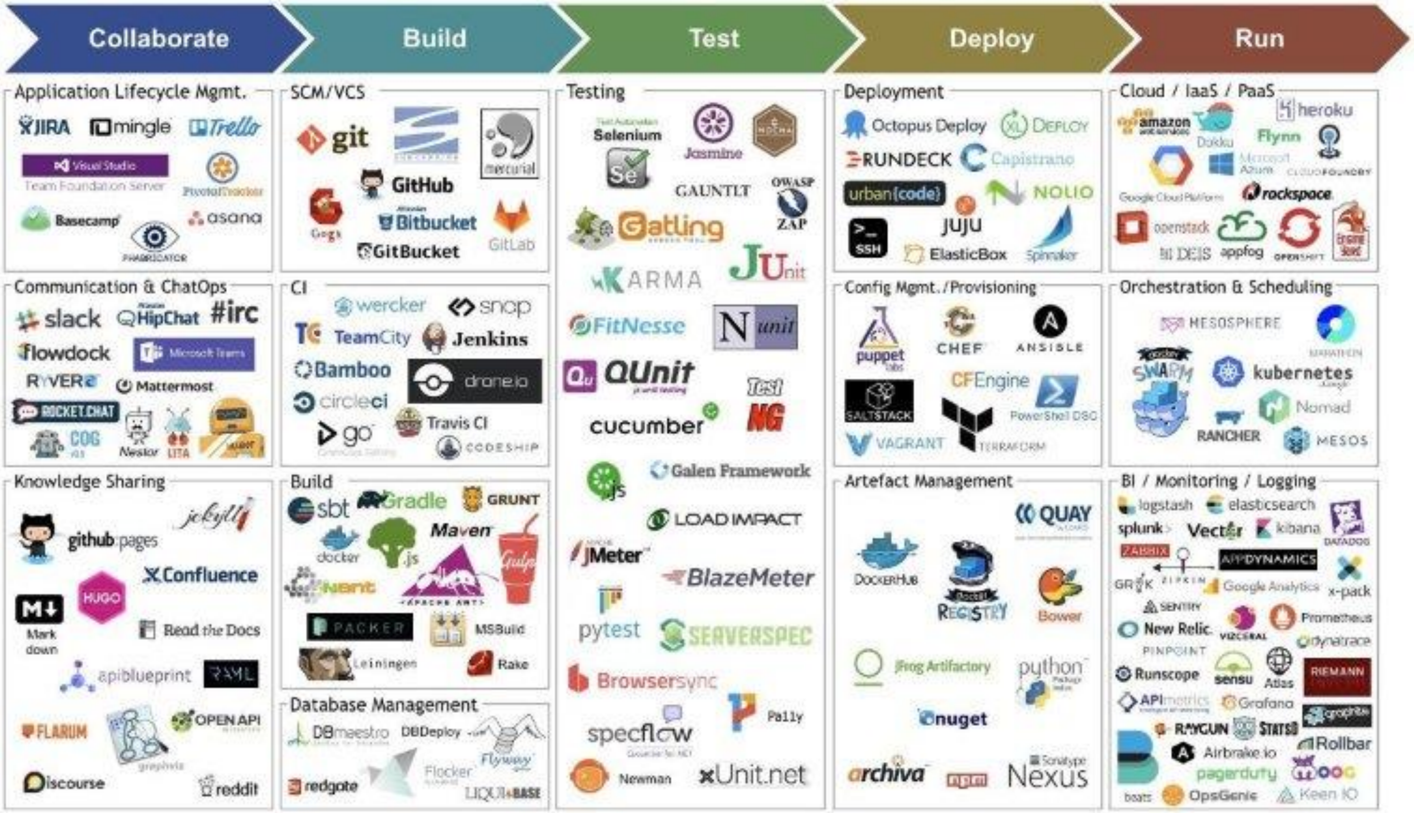
1 Os Gl GitLab																	2 En Sp Splunk
3 Fm Gh GitHub	4 En Dt Datical											5 En XLr XebiaLabs XL Release	6 Fm Aws AWS	7 Pd Az Azure	8 En Gc Google Cloud	9 Fm Op OpenShift	10 Fm Sg Sumo Logic
11 Os Sv Subversion	12 En Db DBMaestro											13 Os Dk Docker	14 En Ur UrbanCode Release	15 Pd Af Azure Functions	16 Pd Ld Lambda	17 Fm Ic IBM Cloud	18 Os Fd Fluentd
19 En Cw ISPW	20 En Dp Delphix	21 Os Jn Jenkins	22 Fm Cs Codeship	23 Os Fn FitNesse	24 Fr Ju JUnit	25 Fr Ka Karma	26 Fm Su SoapUI	27 En Ch Chef	28 Fr Tf Terraform	29 En XLd XebiaLabs XL Deploy	30 En Ud UrbanCode Deploy	31 Os Ku Kubernetes	32 Fm Cc CA CD Director	33 En Pr Plutora Release	34 Pd Al Alibaba Cloud	35 Os Os OpenStack	36 Os Ps Prometheus
37 Pd At Artifactory	38 Fm Rg Redgate	39 Pd Ba Bamboo	40 Fm Vs VSTS	41 Fr Se Selenium	42 Fr Jm JMeter	43 Os Ja Jasmine	44 Pd Sl Sauce Labs	45 En An Ansible	46 Os Ru Rudder	47 En Oc Octopus Deploy	48 Os Go GoCD	49 Os Ms Mesos	50 Pd Gke GKE	51 Fm Om OpenMake	52 Pd Cp AWS CodePipeline	53 Pd Cy Cloud Foundry	54 En It ITRS
55 Pd Nx Nexus	56 Os Fw Flyway	57 Os Tr Travis CI	58 Fm Tc TeamCity	59 Os Ga Gatling	60 Fr Tn TestNG	61 Fm Tt Tricentis Tosca	62 Pd Pe Perfecto	63 En Pu Puppet	64 Os Pa Packer	65 Fm Cd AWS CodeDeploy	66 En Ec ElectricCloud	67 Os Ra Rancher	68 Pd Aks AKS	69 Os Rk Rkt	70 Os Sp Spinnaker	71 Pd Ir Iron.io	72 Pd Mg Moogsoft
73 Fm Bb BitBucket	74 En Pf Perforce	75 Fm Cr Circle CI	76 Pd Cb AWS CodeBuild	77 Fr Cu Cucumber	78 Os Mc Mocha	79 Os Lo Locust.io	80 En Mf Micro Focus UFT	81 Os Sa Salt	82 Os Ce CFEngine	83 En Eb ElasticBox	84 En Ca CA Automic	85 En De Docker Enterprise	86 Pd Ae AWS ECS	87 Fm Cf Codefresh	88 Os Hm Helm	89 Os Aw Apache OpenWhisk	90 Os Ls Logstash

91 En XLi XebiaLabs XL Impact	92 Os Ki Kibana	93 Fm Nr New Relic	94 En Dt Dynatrace	95 En Dd Datadog	96 Fm Ad AppDynamics	97 Os El ElasticSearch	98 Os Ni Nagios	99 Os Zb Zabbix	100 En Zn Zenoss	101 En Cm Checkmarx SAST	102 En Wp Signal Sciences WPP	103 En Bd BlackDuck	104 Os Sr SonarQube	105 Os Hv HashiCorp Vault
106 En Sw ServiceNow	107 Pd Jr Jira	108 Fm Tl Trello	109 Fm Sk Slack	110 Fm St Stride	111 En Cn CollabNet VersionOne	112 En Ry Remedy	113 En Ac Agile Central	114 Pd Og OpsGenie	115 Pd Pd Pagerduty	116 Os Sn Snort	117 Fm Tw Tripwire	118 En Ck CyberArk	119 En Vc Veracode	120 En Ff Fortify SCA

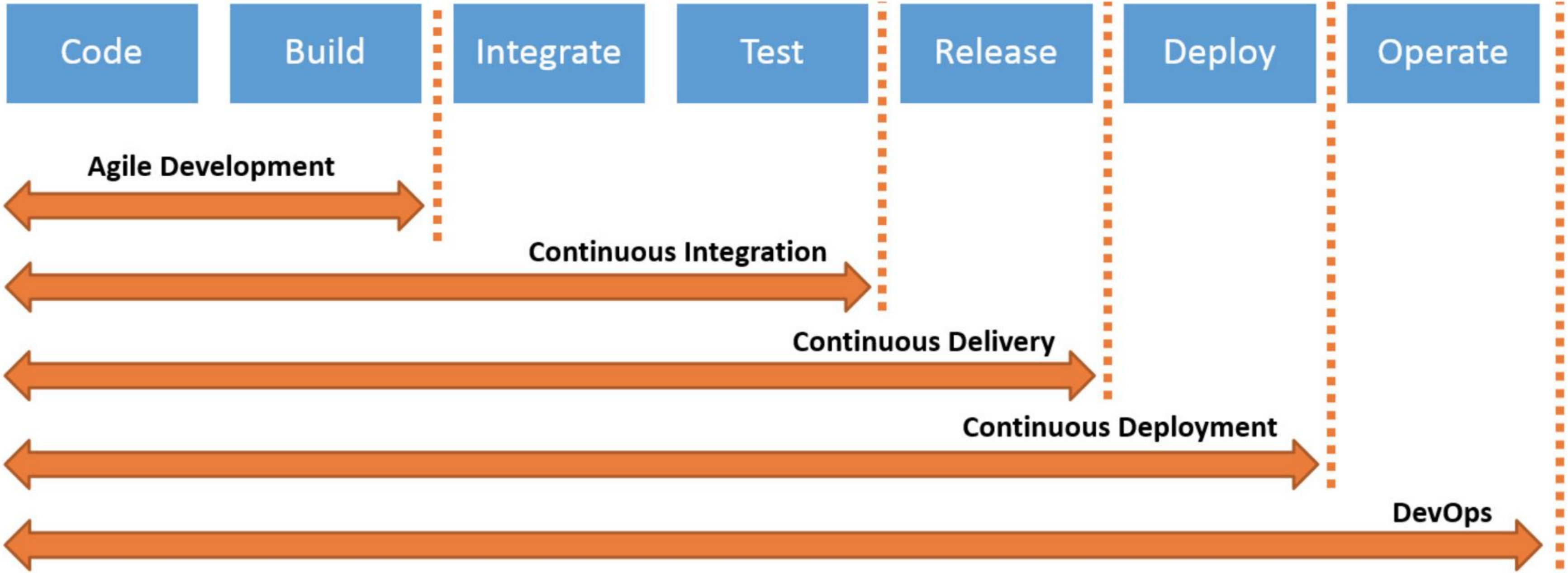
XebiaLabs
Enterprise DevOps

Follow @xebialabs

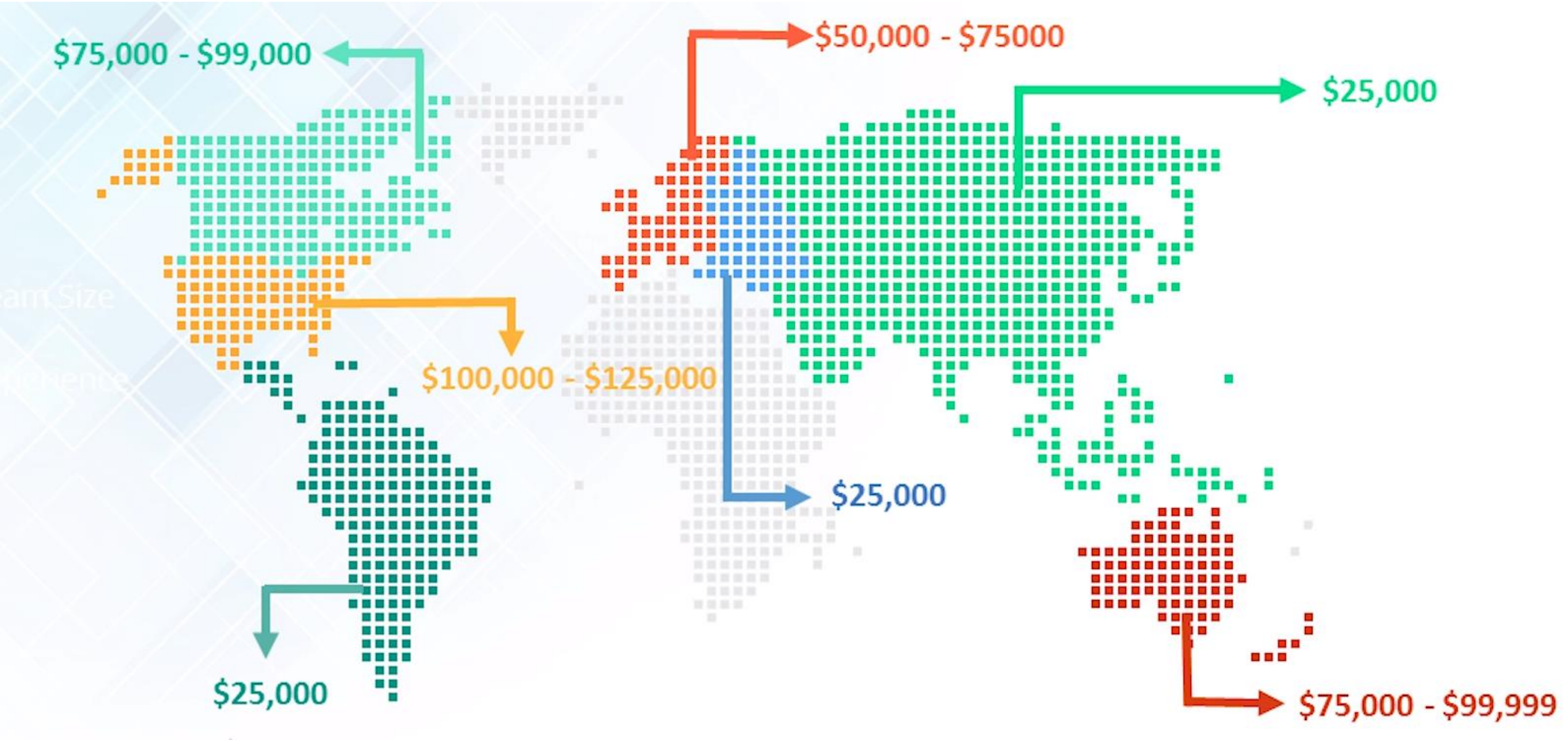
Top DevOps Tools



DevOps covers all stages of working with a software



Average Salary of DevOps



Main areas of knowledge for DevOps and examples

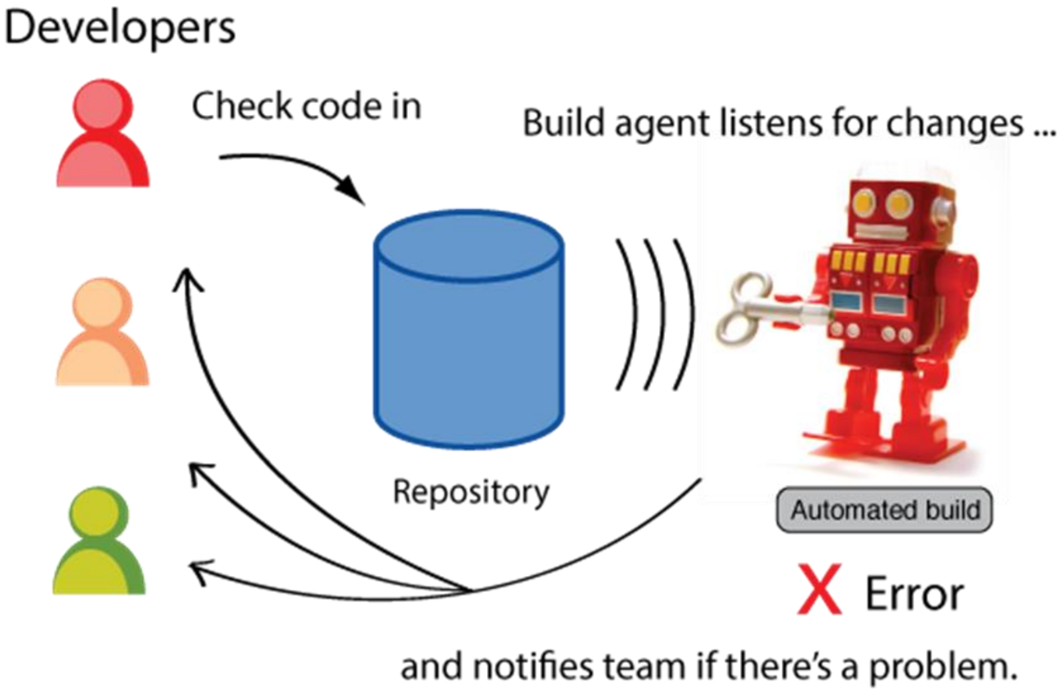
- Programming languages (Bash, Python, Groovy, PowerShell, Ruby, Go.)
- Operation Systems (Linux, Windows)
- Version Control System (Git)
- Cloud Computing (AWS, Google, Azure)
- Containerization (Docker, Kubernetes)
- Infrastructure as code (Ansible, Chef, Puppet, SaltStack)
- CI/CD (Jenkins, Bamboo, TeamCity)

Core practices

- **Continuous Integration (CI)**
- Test Automation (Automated Testing)
- **Infrastructure as Code (IaC),** Configuration Management
- **Continuous Deployment (CD)**
- Load Testing
- Application Performance Monitoring

Continuous Integration (CI)

Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. The CI process is comprised of automatic tools that assert the new code's correctness before integration. A source code version control system is the crux of the CI process. The version control system is also supplemented with other checks like automated code quality tests, syntax style review tools, and more.


Developers
Check code in
Build agent listens for changes ...
Repository
Automated build
X Error
and notifies team if there's a problem.

Continuous Integration

BUSINESS VALUE

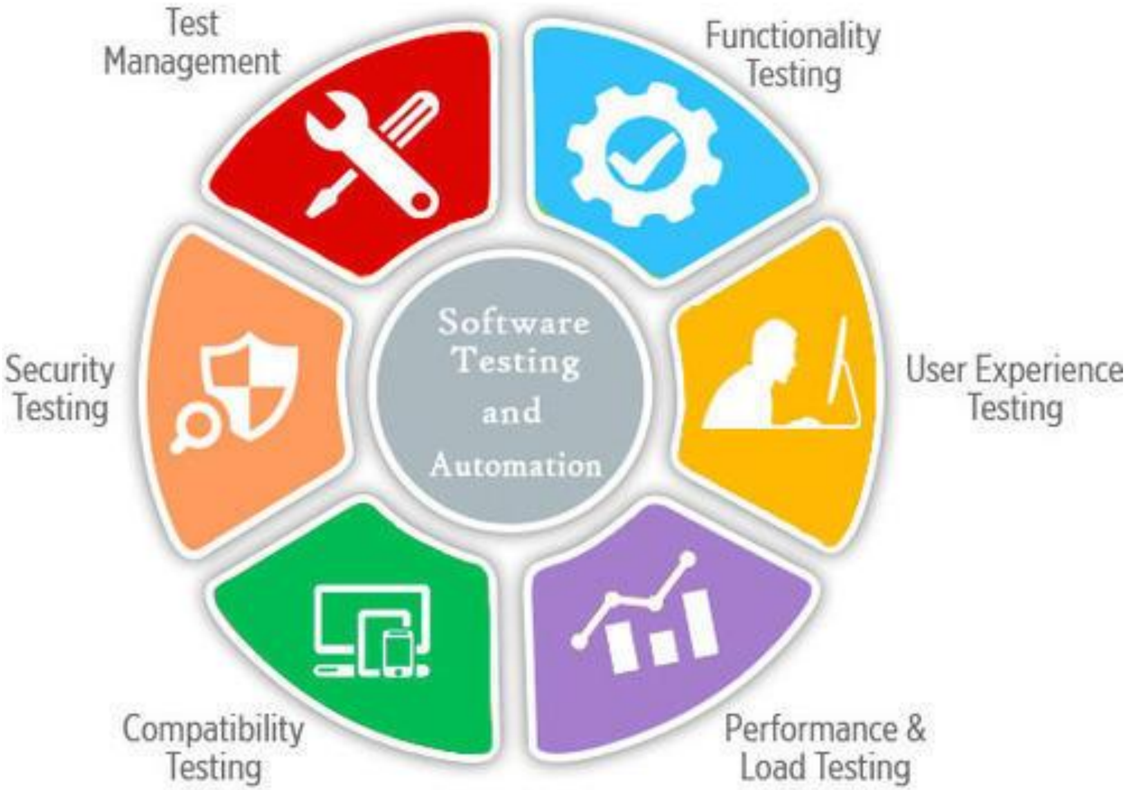
- **Accelerate Delivery** - achieved by the fact that we immediately find out about the build error and, accordingly, we can begin to fix it faster.
- **Repeatability** - the whole process is repeatable, that is, if no changes have occurred, then the assembly will also be successful (or not successful). There is no such problem as the fact that one developer is going to everything, while the other is not.
- **Optimized Resources** - there is no need to manually start the assembly, on a person's computer or build server, there is no need to prepare the assembly - pump out the sources from source control, etc.

MEASURABILITY

- **Deployment Lead Time** - The time it takes to build a project.
- **MTTR** (Mean Time To Repair). You can measure the time elapsed from reporting an incorrect assembly to a fix that removes the error.
- **MTTD** (Mean Time To Detect). measured the time that elapsed from the introduction of the error, to determine that the problem arose and what it is.

Test automation vs Automated Testing

Automated testing is the act of conducting specific tests via automation (e.g. a set of regression tests) as opposed to conducting them manually, while **test automation** refers to automating the process of tracking and managing the different tests.



Test automation and Automated Testing

BUSINESS VALUE

- **Accelerate Delivery** We quickly get information about whether the assembly is valid and whether it can be released.
- **Repeatability** - the test always runs in the same sequence, in the same scenario, so the result will be the same.
- **Optimized Resources** . Automatic tests are cheaper than manual tests due to the fact that their execution is much cheaper than verification using manual testing.

MEASURABILITY

- **Deployment Lead Time** - time required for deployment (assembly, inspection).
- **MTTR** - In this case, the time is measured from diagnosing the error to correcting it (passing the test successfully).
- **MTTD** - since the tests are automatic, it is possible to measure the time from the assembly of the project to the receipt of an error report according to the test results.

Infrastructure as Code (IaC), Configuration Management

- Infrastructure as code, also referred to as IaC, is a type of IT setup wherein developers or operations teams automatically manage and provision the technology stack for an application through software, rather than using a manual process to configure discrete hardware devices and operating systems. Infrastructure as code is sometimes referred to as programmable or software-defined infrastructure.

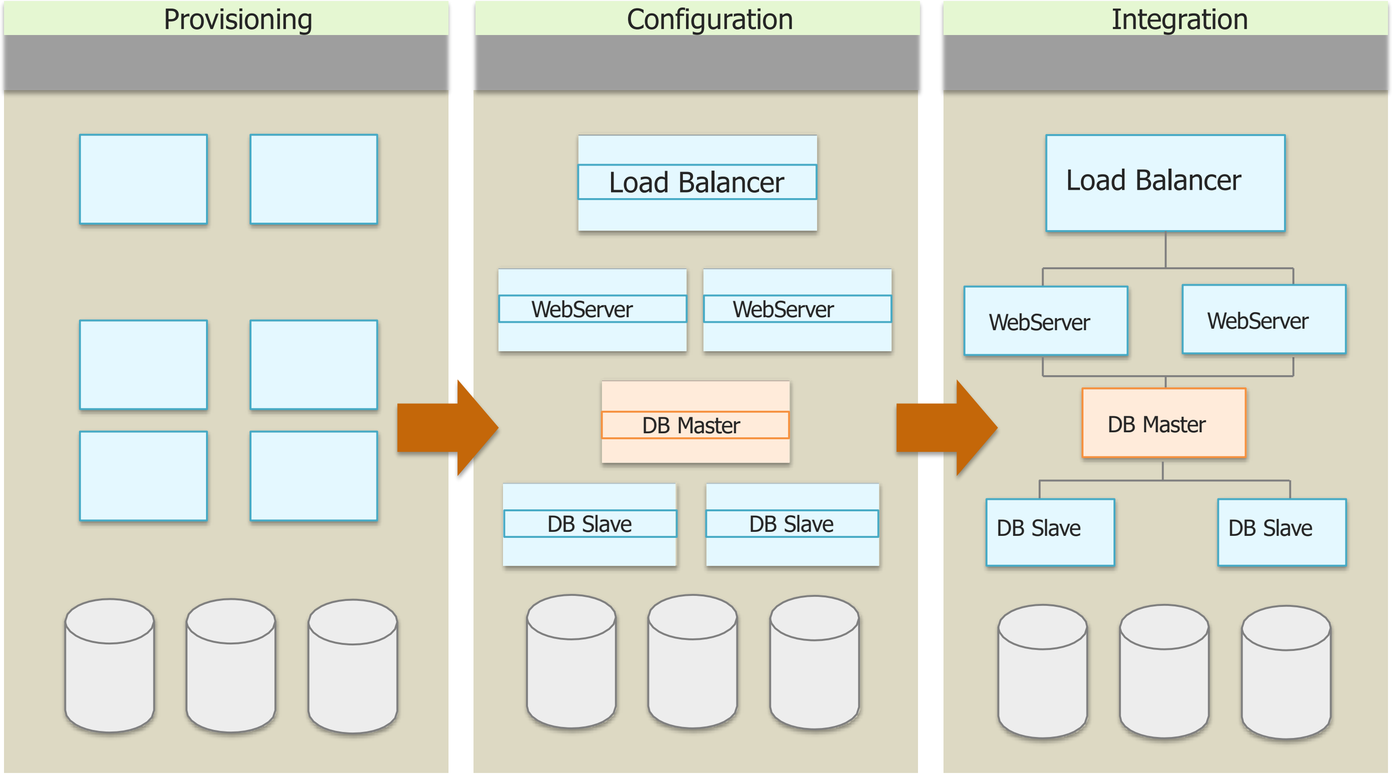
ANSIBLE

CHEF™

puppet

SALTSTACK

DevOps Life Cycle



Continuous deployment (CD) vs. continuous delivery (CD)

- **Continuous deployment** is a strategy for software releases wherein any code commit that passes the automated testing phase is automatically released into the production environment, making changes that are visible to the software's users
- Continuous integration, delivery and deployment are collectively referred to as continuous software development, and they are associated with the Agile and DevOps methodologies. Continuous delivery and deployment originate from continuous integration, a method to develop, build and test new code rapidly with automation so that only code that is known to be good becomes part of a software product.
- Continuous deployment is not the same thing as continuous delivery, although the two terms are often confused and, indeed, share the acronym of CD.
- **Continuous delivery** occurs when developers frequently hand off new code to the quality assurance (QA) and operations teams for testing. Continuous delivery usually involves a production-like staging area, and there is often a time lag between a release and when it is reviewed, when changes are manually accepted and when the new code is released to production

Release Management

- Release Management **is the management of the software delivery lifecycle across multiple projects** and departments within a large organization. It is the orchestration of activities and resources across multiple, interdependent software releases and changes initiatives to deliver software at scale. While managing both the technical and organizational complications that accompany delivering changes to enterprise-scale, composite systems within a large organization.
- An example of the criteria by which the assembly is ready, and if they are met, the delivery of Continuous Deployment automatically starts can be

- **DEV environment** - the assembly went without errors.
- **STAGE environment** - the assembly was installed on the DEV environment and unit tests were successful.
- **PROD environment** - the assembly was tested on the STAGE environment, there are no more than 5% minor bugs, there are no major bugs, QA Lead and Dev Lead set the Confirm build to readiness for the PROD environment.

Load Testing

- Load testing is a kind of Performance Testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

This testing usually identifies -

- The maximum operating capacity of an application
- Determine whether the current infrastructure is sufficient to run the application
- Sustainability of application with respect to peak user load
- Number of concurrent users that an application can support, and scalability to allow more users to access it.

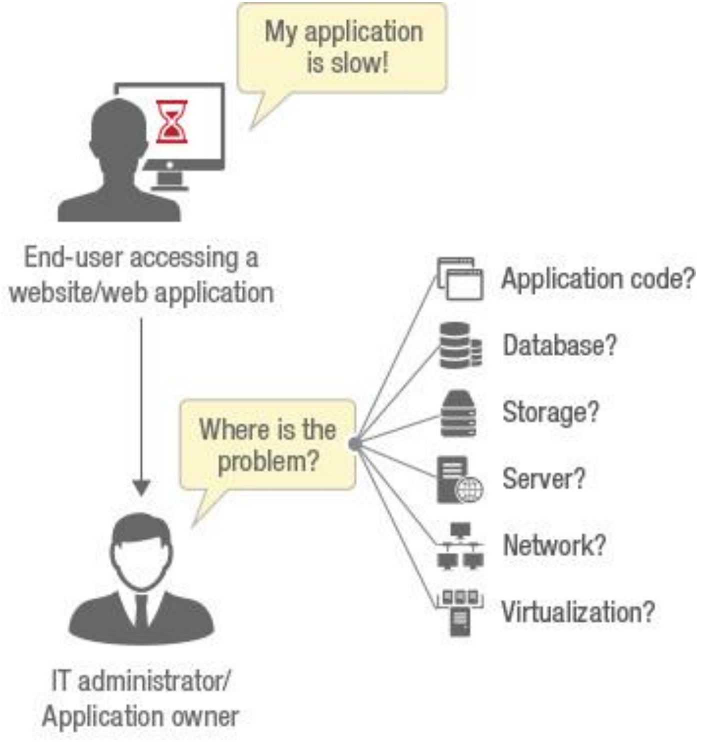
It is a type of non-functional testing. In Software Engineering, Load testing is commonly used for the Client/Server, Web-based applications - both Intranet and Internet.

Application Performance Monitoring

- Application performance monitoring is the strategy and practice of continuously monitoring and tracking the performance of business applications and the user experience of end users as they access the applications to understand trends, isolate anomalies, and get actionable insight for problem resolution and code optimization.

Application Performance Monitoring solutions ensure performance of business critical applications and give the capabilities to proactively ensure that application performance meets user expectations and business priorities. It helps:

- Monitor and manage performance and availability of application delivered business functions
- Gain accurate troubleshooting and performance optimization
- Recognize unwanted patterns in transaction performance or user behavior
- Efficiently collaborate on and quickly resolve application performance issues impacting the business
- Detect and diagnose application performance problems to maintain an expected level of service
- Translation of IT metrics into business meaning (value)
- Proactively identify the individual contributors (transactions) and culprits of performance degradation
- Predictive analysis based on the historical performance trends



The CALMS Framework for DevOps

Culture
Automation
Lean
Measurement
Sharing

https://www.atlassian.com/en/devops

Q&A

Thank you!