NÁSKOK
DÍKY
ZNALOSTEM

PROFINIT

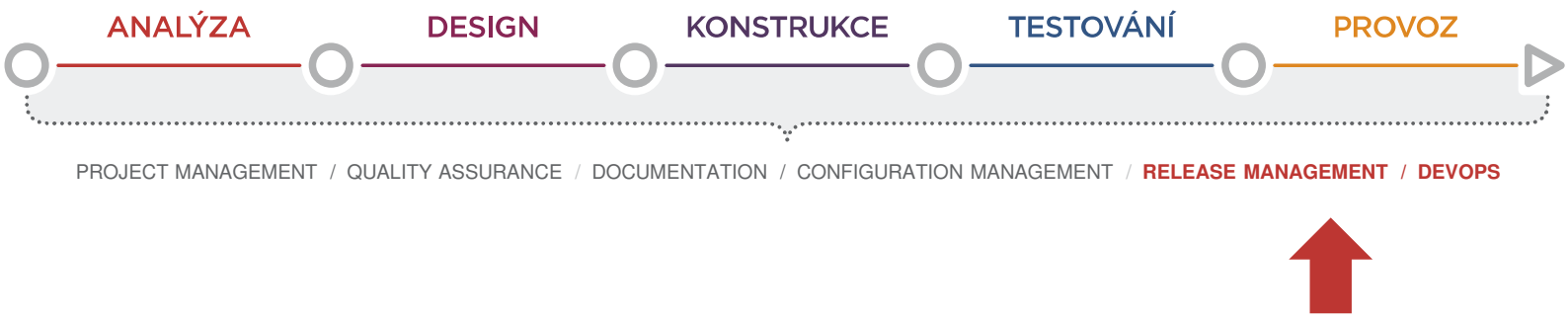
Release management, DevOps

Michal Petřík

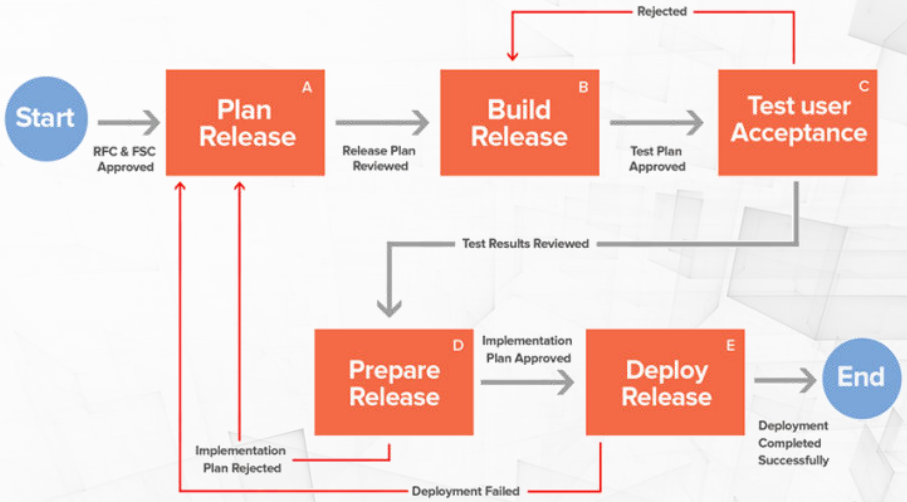
15. duben 2020

Téma dnešní přednášky

1. Release management
2. Continuous integration / delivery / deployment / DevOps
3. Ukázky z praxe
4. Diskuze



PROFINIT



Release management

Motivace

Než se váš systém/změna dostane do produkce:

- musí ho někdo vyvíjet a testovat,
- musí být *někde* před nasazením do produkce akceptován
- může existovat více produkčních prostředí
- …

Typicky existují další prostředí mimo cílové !

PROFINIT

Sladění terminologie

- Často je pojem Release managementu chápán mírně odlišně vzhledem k dnešní prezentaci
- *Oddělení RM má zajistit, že se stovky systémů dostanou do produkce včas, jsou integrovány, …*

- Aby mohl fungovat celek, musí fungovat i každá dílčí část
- → o tom bude dnešní přednáška

PROFINIT

Systém do produkce

Co musíme umět pro každý systém, než se dostane do produkce:

- **Vyrobit** dodávku
- Připravit dodávku pro **instalaci zadavatelem**
- **Nainstalovat** dodávku

- Dodat **systém jako celek**
- **Opravit** malou drobnost

  …a opravit ji **rychle a ekonomicky** …
- Poradit si s **různými typy prostředí**
  - Aplikační server, databázový a replikační server, operační systém, …

- Release **není** „jen o nasazení nové verze“

PROFINIT

Typy prostředí

Prostředí mohou být různého typu (různé pohledy):

- Typicky virtualizovaná

- Nově „kontejnerizovaná“

- On-premise x Cloud (SaaS, PaaS, …)

- Jednotky prostředí x tisíce

PROFINIT

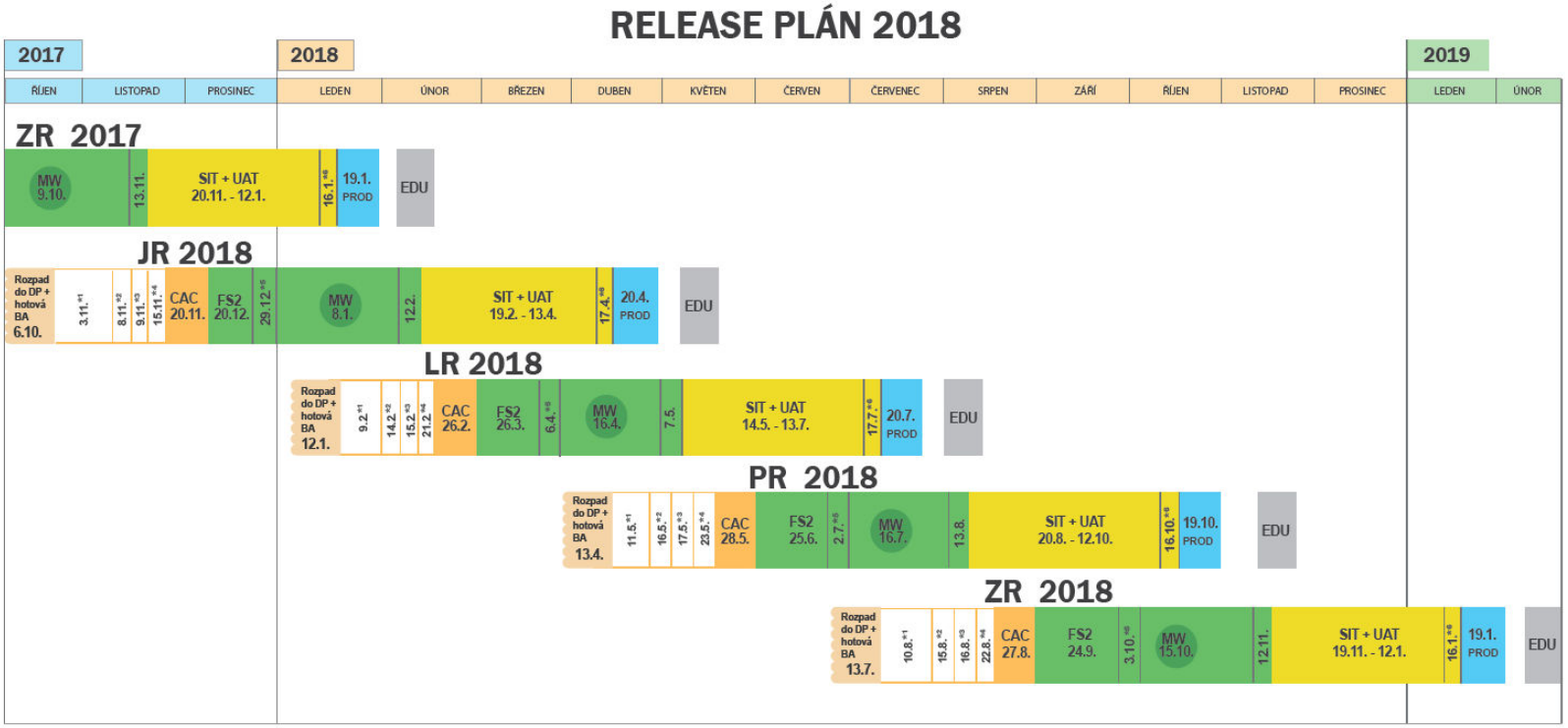
Komplexita

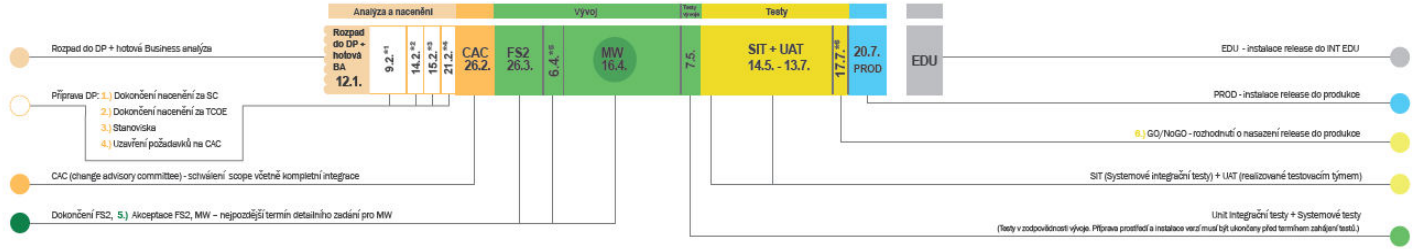
- Integrující se aplikace a jejich závislosti
  - Budou další systémy připraveny?
  - Bude možná integrace?
  - …
- Rozdílné vývojové cykly aplikací
  - Vodopád x agile
  - Front-end x back-end
  - Mobilní aplikace x web aplikace x „tlustý klient“
  - …
- … zpoždění v rozsahu dodávek
- Něco jiného se testuje, něco jiného je na produkci
  - Potřebuji testovat opravu produkce pro verzi X, ale na akceptačním prostředí mám nyní verzi X+1…

JE POTŘEBA ŘÁD A PLÁN

PROFINIT

Typické řešení



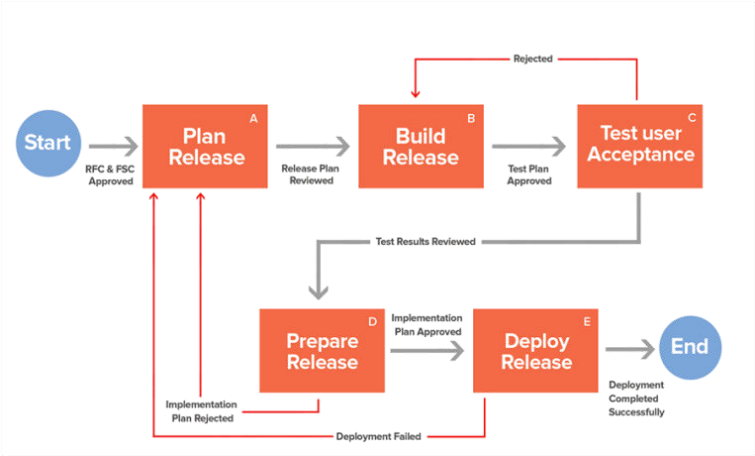


PROFINIT

Typické řešení

Je nutné postihnout minimálně:

- Celkový proces
  - Odpovědnosti, styčné osoby
  - Komunikační matice
  - Formální náležitosti
  - …



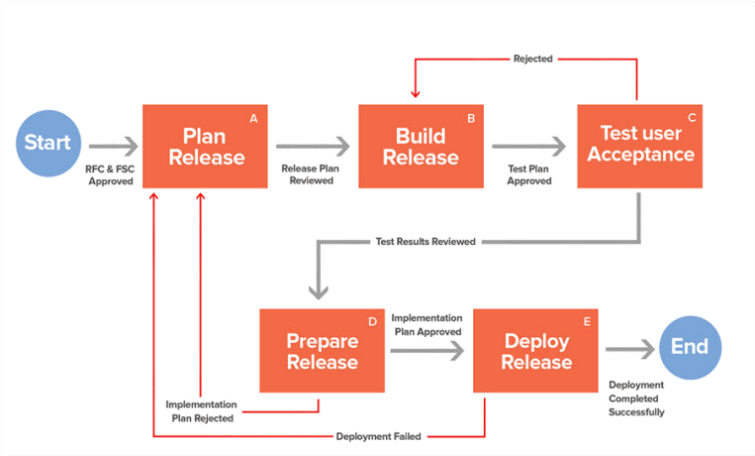
- Klíčové milníky
- Dopady na okolní systémy
- Prerekvizity
- Kvalitativní nároky

- … a to vše pro všechny systémy se zanesením jejich vazeb, …

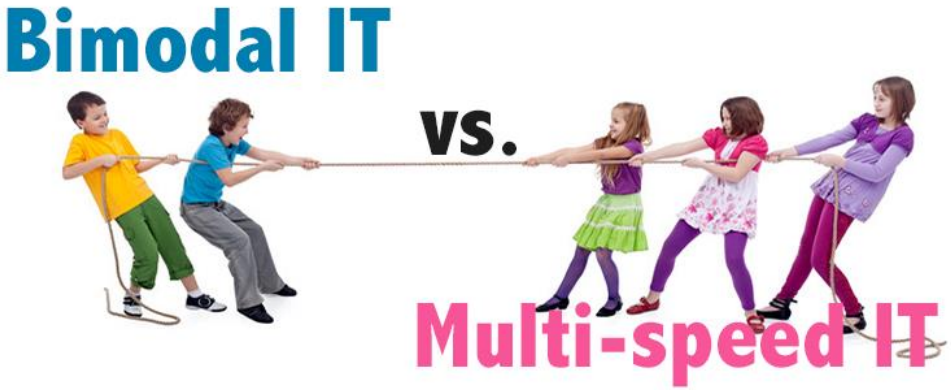
PROFINIT

Klíčové pojmy

- Release (Build)
- Oprava buildu (patch)
- Instalační set
  - En-bloc
  - Inkrementální

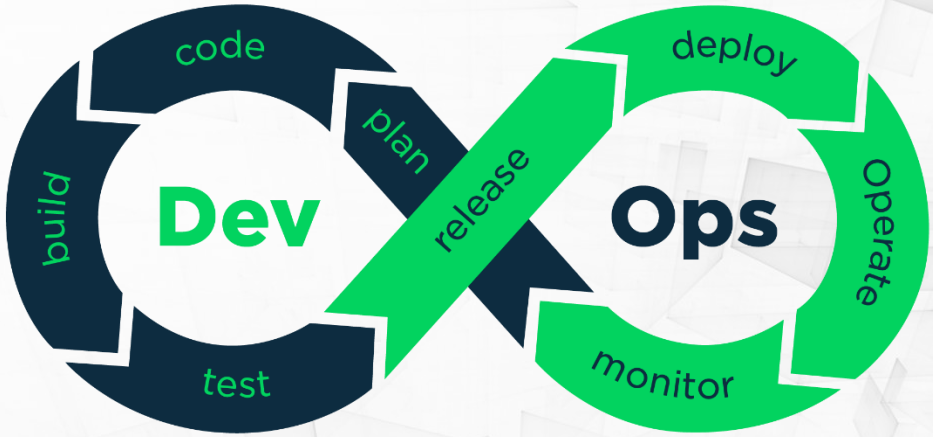


- … slyšeli jste již o multi-speed / Bi-modal IT?



Zdroj: http://www.devops.com

PROFINIT



Continuous integration / delivery / deployment / DevOps

O čem se vlastně bavíme?

- Vytvářený software má typicky tisíce/miliony řádků kódu
- Všechny artefakty musí někdo:

  tzv. sestavit (build) → otestovat → nasadit → znovu otestovat …

- Lze toto realizovat ručně?

  Ano → **ALE…**

- Člověk se může „uklepnout“ (…a pak je průšvih…)
- Ruční práce se nemůže měřit s výhodami automatizace
  - Rychlost zpětné vazby
  - Opakovatelnost
  - Bezpečnost

PROFINIT

O čem se vlastně bavíme?

- Komplexita systémů a technologií vede na nutnost automatizace
  → jsme jen lidé a děláme chyby

- Vše, co lze automatizovat, by mělo být automatizováno

- Některé technologie již ani nepředpokládají ruční zásah

- Co lze automatizovat?
  - Generování kódu
  - Generování dokumentace
  - Testy
  - Sestavení aplikace
  - Nasazení aplikace
  - **… téměř vše …**

PROFINIT

Co je DevOps?

- **Dev**elopment & **Op**eration**s**

By Daniel Johnson, FORMULA ONE CORRESPONDENT

19 JUNE 2016 • 6:26PM

After 23 hours and 55 minutes, having covered more than 3,200 miles, Anthony Davidson's dream of winning the Le Mans 24 Hours for the first time seemed certain to finally come true. But in the cruellest of denouements, his Toyota broke down on the final lap, losing him the lead and giving victory to Porsche with just three minutes to go.

Davidson and the other two drivers in his car, Sébastien Buemi and Kazuki Nakajima, were having to come to terms last night with one of the most heartbreaking defeats in the history of the famous, gruelling race.

PROFINIT

Co je DevOps?

- **Dev**elopment & **Op**eration**s**

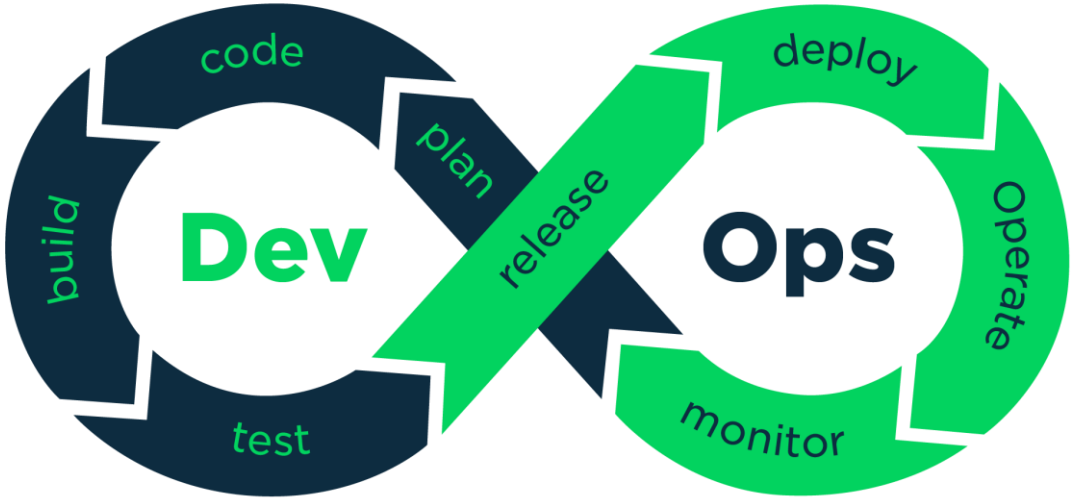
PROFINIT

Co je DevOps?

- **Dev**elopment & **Op**eration**s**

Často pracujeme například na analýze/vývoji a ani netušíme, kdo bude naše výstupy nasazovat do produkce

Představte si ale, že jste najednou součástí jednoho týmu, který pracuje dohromady za jediným cílem → funkční software



PROFINIT

Co je DevOps?

Daily Build

- Compile & Assemble
- Smoke test

Continuous Integration

- Build & Test
- Every Commit
- Static Analysis

Continuous Delivery

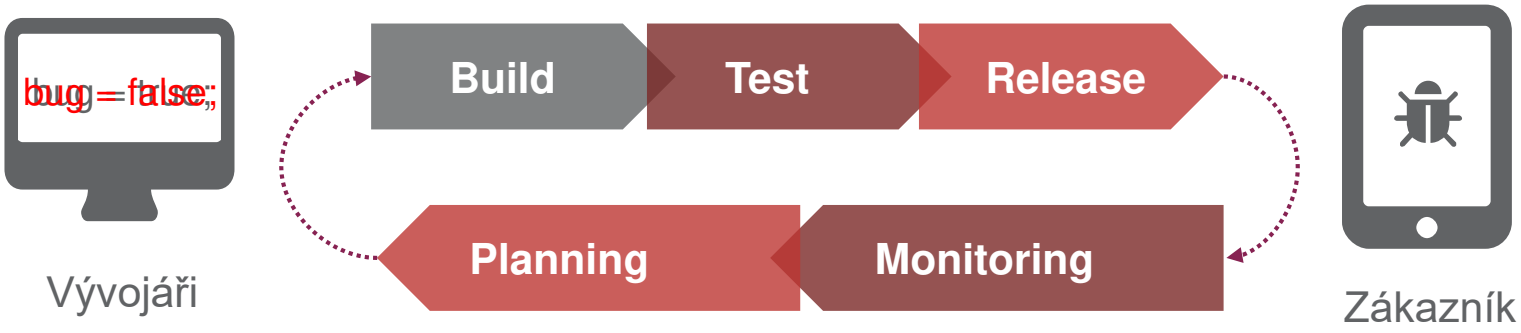
- Continous integration
- Provisioning
- Configuration management
- Deployment

Počátek devadesátých let → **Konec devadesátých let** → **Přelom století**

PROFINIT

Co je DevOps?

- Forma vývojového cyklu
  - Každý krok je maximálně automatizován
  - Vše je verzováno a testováno (nejen kód, ale i model databáze, data, ...)
  - Na všechna prostředí se používá jeden unifikovaný proces
  - Celý cyklus řešen formou malých kroků → Deployment Pipeline
  - Rychlá a maximální zpětná vazba
  - → podporuje Agilní vývoj (Agile bez DevOps lze jen obtížně realizovat)



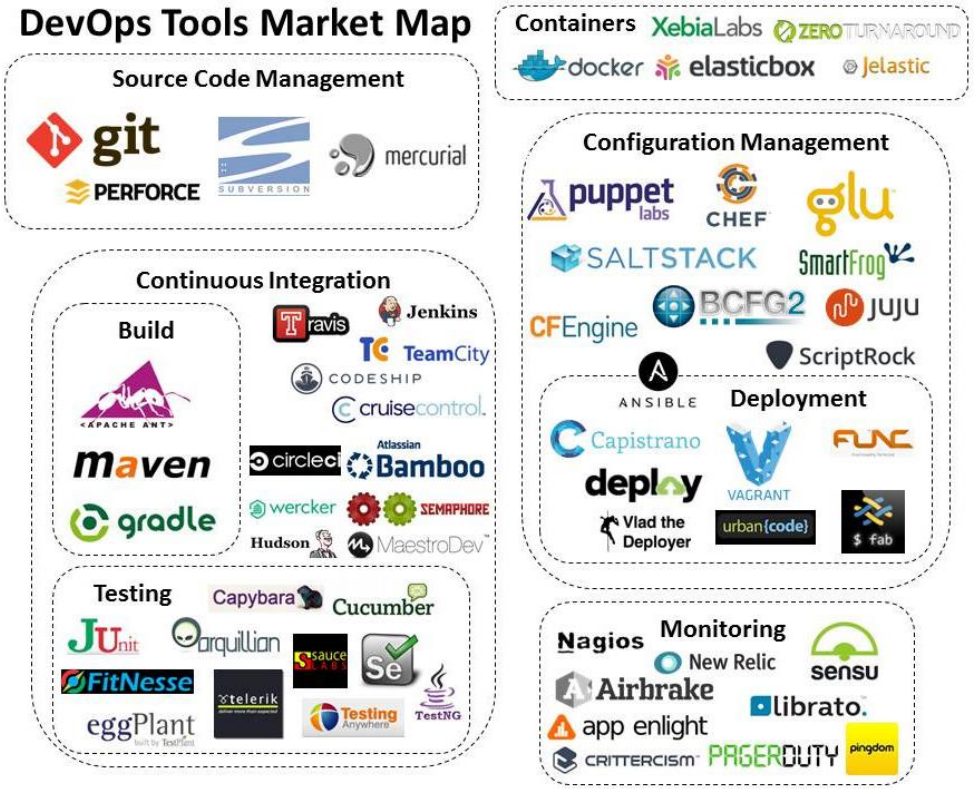
- Je nutná odpovídající kultura ve vývojovém týmu

Inspirováno http://www.slideshare.net/AmazonWebServices/dvo202-devops-at-amazon-a-look-at-our-tools-processes

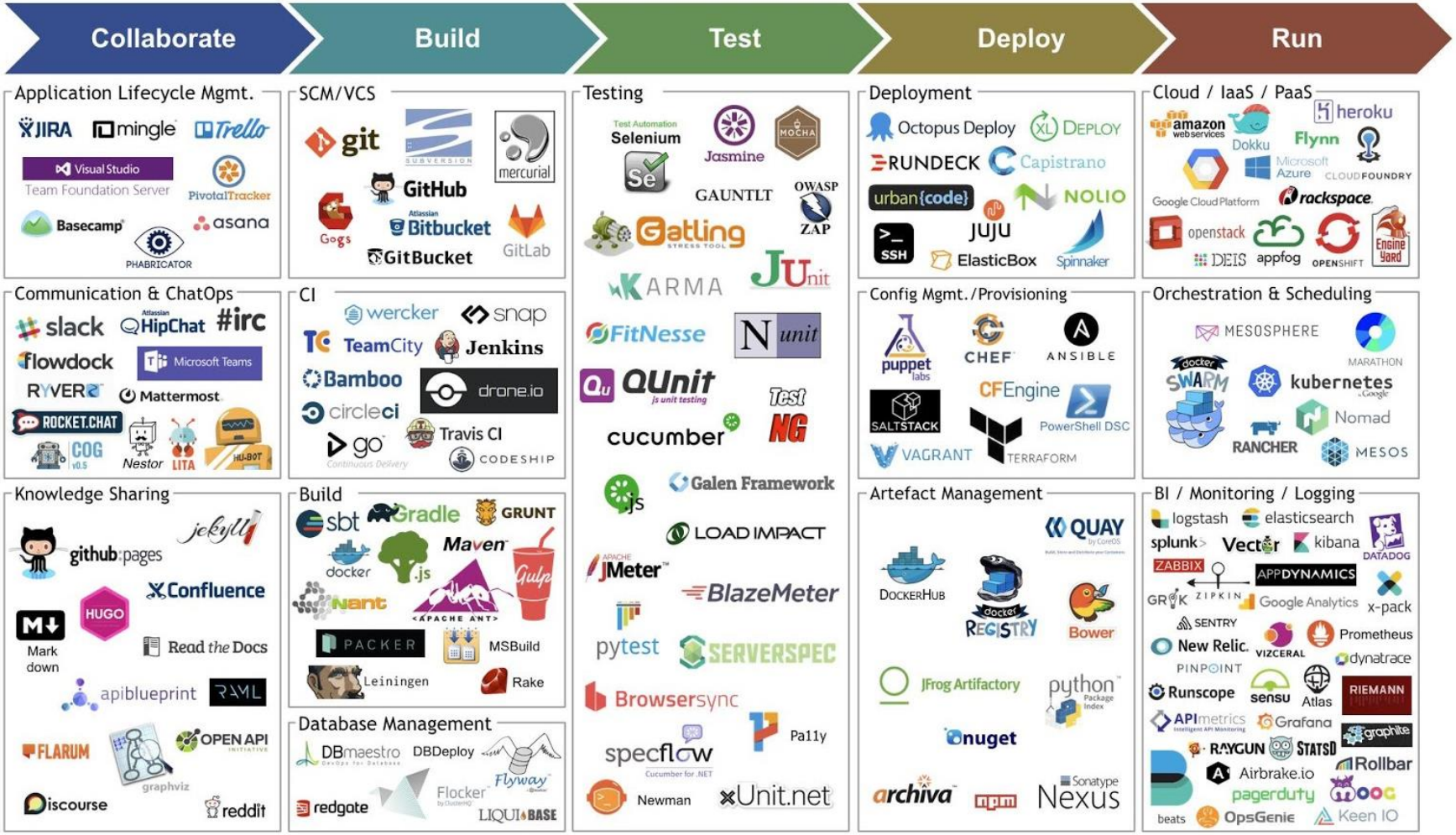
PROFINIT

Co je DevOps?

- Efektivní využití nástrojů
  - Version Control
  - Provisioning
  - Configuration Management
  - Build Automation
  - Artifact Repository
  - Static Analysis
  - Automated Testing
  - Test Data
  - Continuous Integration and Delivery
  - System Monitoring & Analytics
  - **... a vůle zlepšovat**

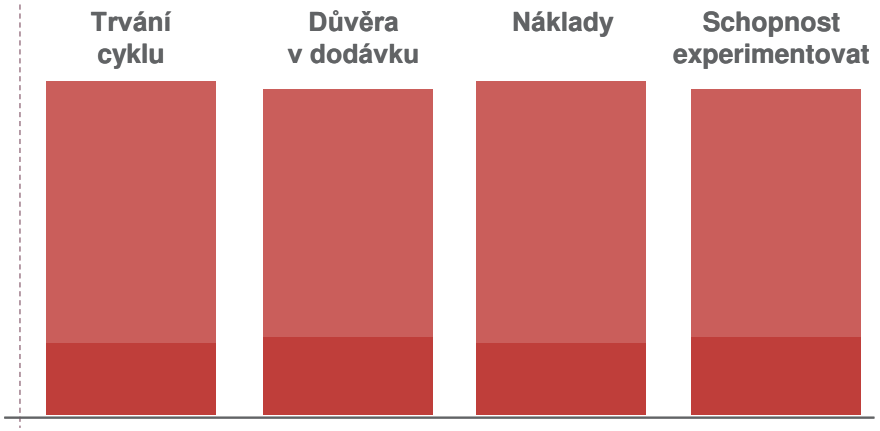


Co je DevOps?


Collaborate
Build
Test
Deploy
Run
Application Lifecycle Mgmt.
JIRA
mingle
Trello
Visual Studio
Team Foundation Server
PivotalTracker
Basecamp
asana
PHABRICATOR
SCM/VCS
git
SUBVERSION
mercurial
GitHub
Gogs
Atlassian Bitbucket
GitLab
GitBucket
Testing
Test Automation Selenium
Jasmine
MOCHA
GAUNTLT
OWASP ZAP
Gatling
JUnit
KARMA
FitNesse
N unit
QUnit
Test NG
cucumber
Galen Framework
LOAD IMPACT
JMeter
BlazeMeter
pytest
SERVERSPEC
Browsersync
Pa11y
specflow
Newman
xUnit.net
Deployment
Octopus Deploy
XL DEPLOY
RUNDECK
Capistrano
urban{code}
NOLIO
SSH
JUJU
ElasticBox
Spinnaker
Cloud / IaaS / PaaS
amazon web services
heroku
Dokku
Flynn
Microsoft Azure
CLOUDFOUNDRY
Google Cloud Platform
rackspace
openstack
DEIS
appfog
OPENSHIFT
Engine Yard
Communication & ChatOps
slack
HipChat
#irc
flowdock
Microsoft Teams
RYVER
Mattermost
ROCKET.CHAT
COG
Nestor
LITA
HU-BOT
CI
wercker
snap
TeamCity
Jenkins
Bamboo
drone.io
circleci
Travis CI
go
CODESHIP
Config Mgmt./Provisioning
puppet labs
CHEF
ANSIBLE
SALTSTACK
CFEngine
PowerShell DSC
VAGRANT
TERRAFORM
Orchestration & Scheduling
MESOSPHERE
MARATHON
docker SWARM
kubernetes
Nomad
RANCHER
MESOS
Knowledge Sharing
github:pages
jekyll
Confluence
HUGO
Mark down
Read the Docs
apiblueprint
RAML
FLARUM
OPEN API
graphviz
Discourse
reddit
Build
sbt
Gradle
GRUNT
docker
Maven
.js
Gulp
Nant
<APACHE ANT>
PACKER
MSBuild
Leiningen
Rake
Artefact Management
QUAY
DockerHub
docker REGISTRY
Bower
JFrog Artifactory
python Package Index
nuget
archiva
npm
Sonatype Nexus
BI / Monitoring / Logging
logstash
elasticsearch
splunk>
Vector
kibana
DATADOG
ZABBIX
APPDYNAMICS
GROK
ZIPKIN
Google Analytics
x-pack
SENTRY
New Relic.
VIZCERAL
Prometheus
dynatrace
PINPOINT
Runscope
sensu
Atlas
RIEMANN
APImetrics
Grafana
graphite
RAYGUN
STATSD
Airbrake.io
Rollbar
pagerduty
beats
OpsGenie
Keen IO
Database Management
DBmaestro
DBDeploy
Flyway
redgate
Flocker
LIQUIBASE

Hlavní přínosy

- Snížení TTM business požadavků
- Snížení množství chyb se současným zvýšení rychlosti jejich oprav
- Snížení nákladů na zdroje (development i operations)
- Jednodušší zapojení nových lidí do týmu
  - silná zpětná vazba
  - tzv. Self-service
- Možnost „bezpečného experimentování“



PROFINIT

DevOps a architektura
(aneb... *ale u nás by to nešlo*)

DevOps a architektura

Velmi často pracujeme s tímto:

...A good old monolith

PROFINIT

DevOps a architektura

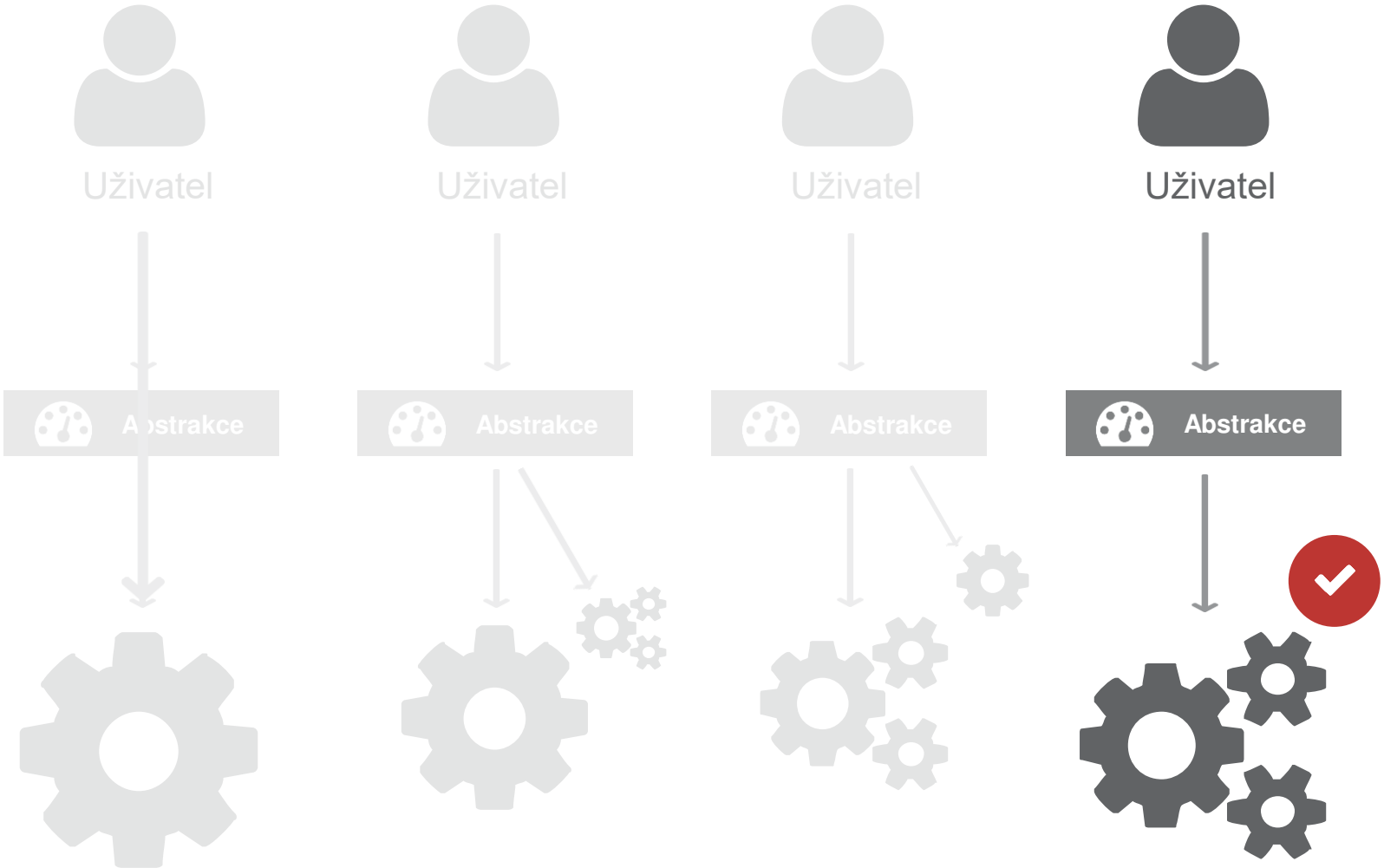
Raději bychom možná ale pracovali s tímto:

... brand new
cool Microservices

PROFINIT

DevOps a architektura

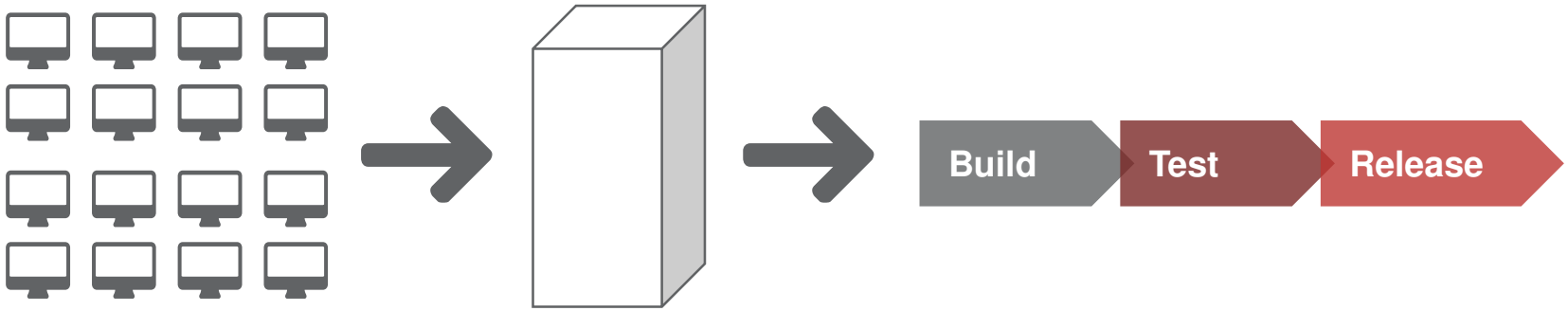
Ve spojení s DevOps lze aplikovat tzv. „Strangler Pattern“

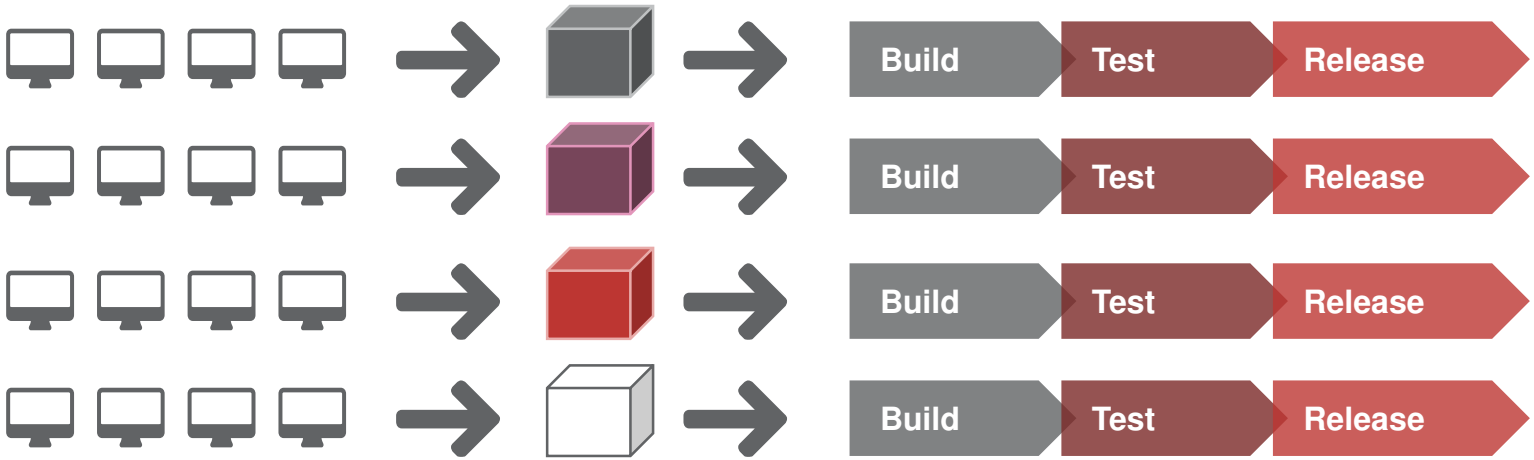


PROFINIT

DevOps a architektura

Změna architektury jde ruku v ruce s release cyklem





PROFINIT

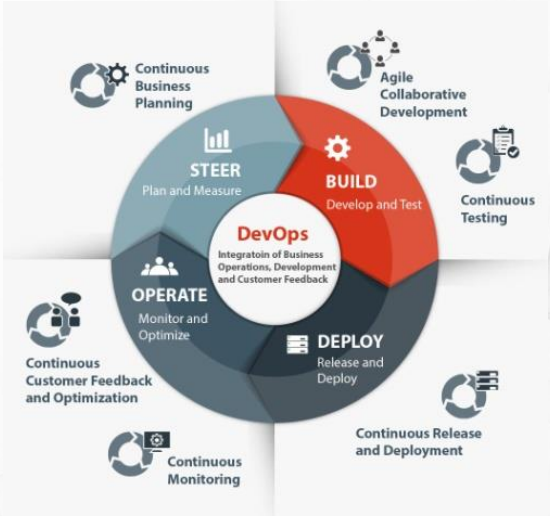
DevOps a architektura

- Stejně jako u změny architektury lze DevOps zavádět postupně
- Postupovat lze „z obou stran“
  - Ze strany vývoje například ve formě automatického buildu, unit testů, ...
  - Ze strany automatizace nasazení na prostředí, apod.
- Je vhodné zvolit / identifikovat oblasti, ve kterých změna nejvíce prospěje
  - Předpokládá sběr a vyhodnocování správných metrik, například zdroje chyb, důvody odstávek → monitoring

PROFINIT

DevOps a architektura

PROFINIT



Ideální podoba

Evoluce automatizace

Agilní vývoj

Continuous Integration

Continuous Delivery

Continuous Deployment

DevOps

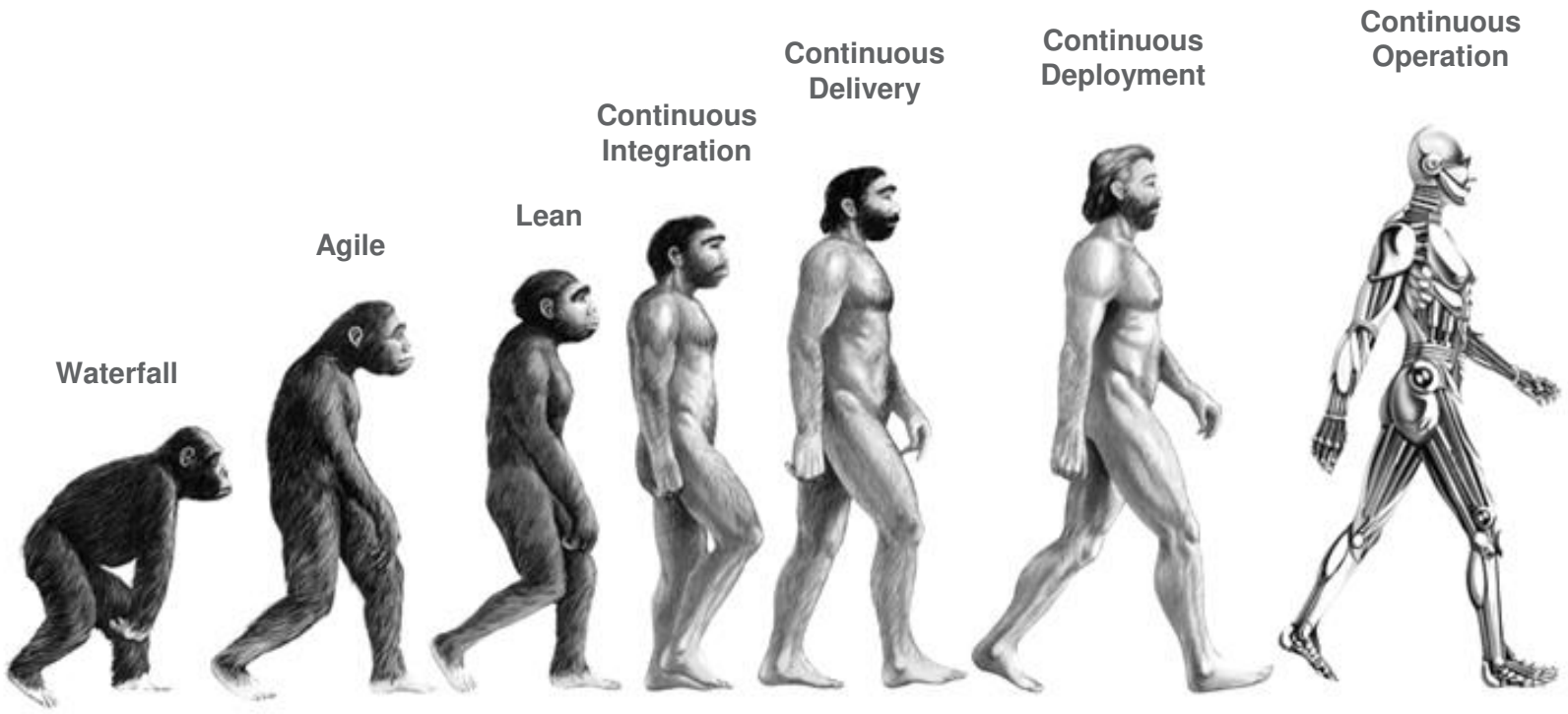
PROFINIT

Evoluce automatizace

Code → Build → Integrate → Test → Release → Deploy → Operate

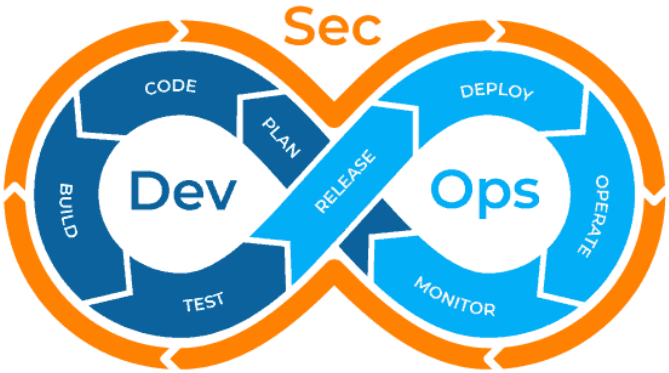
PROFINIT

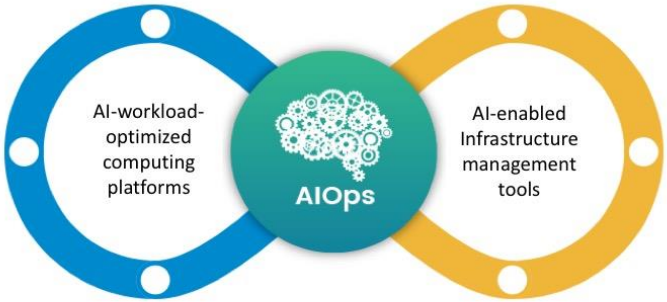
DevOps Movement



Evoluce automatizace – Současnost / budoucnost

Code → Build → Integrate → Test → Release → Deploy → Operate

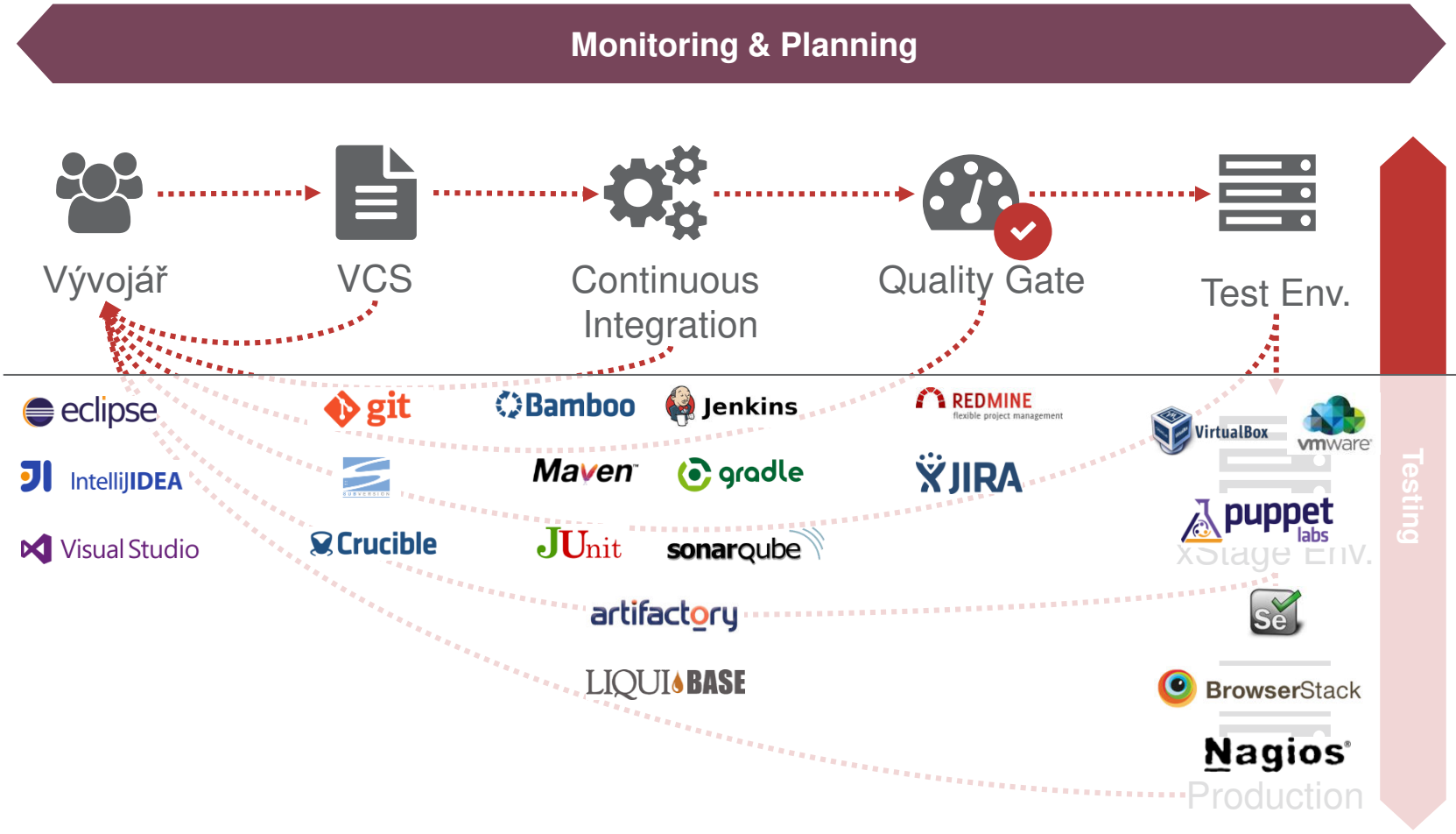




NoOps.

PROFINIT

Ideální podoba cesty jednoho řádku kódu



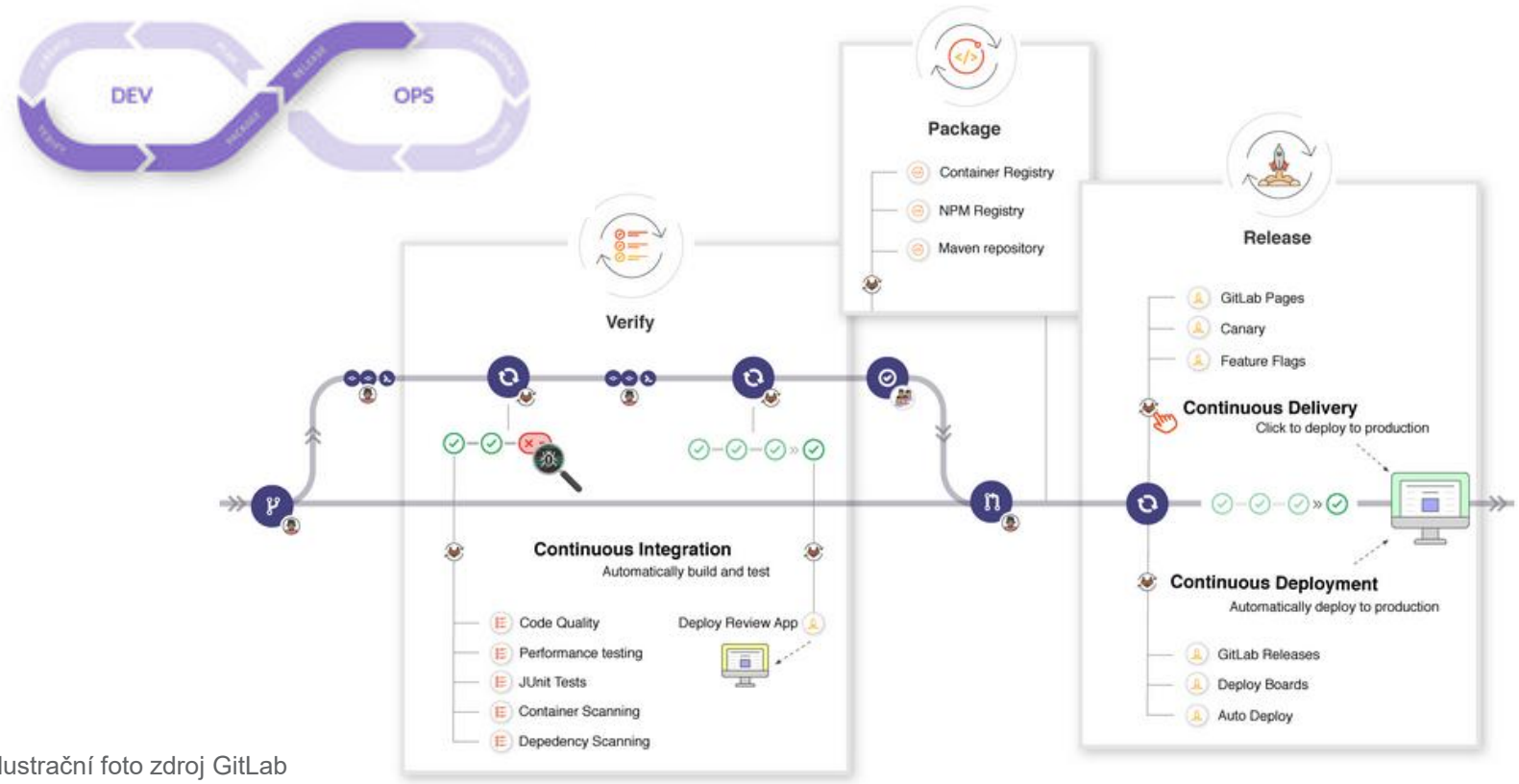
PROFINIT

GitLab CI a CD – from idea to production

- Have an idea
- Create an issue to discuss it with your team
- Ship the code within a merge request
- Run automated scripts (sequential or parallel)
  - Build, test **and deploy** to a **staging environment**
  - Preview the changes
- Review the code and get it approved
- Merge the feature-branch into `master`
  - **Deploy** your changes **automatically** to a **production environment**
- Rollback if something goes wrong

PROFINIT

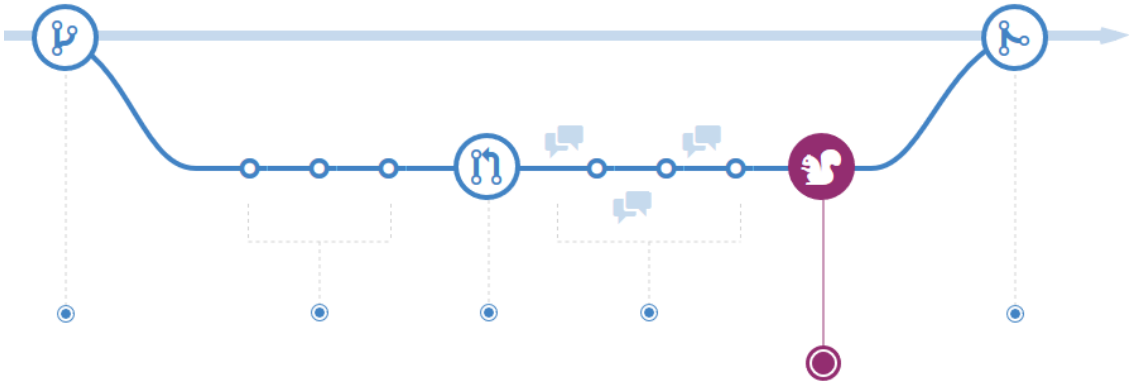
GitLab CI/CD - stages



Ilustrační foto zdroj GitLab

PROFINIT

GitHUB Flow



tmm1
/queue me for github

Hubot
`tmm1: Ok, I added you to the queue for github. There are 3 people ahead of you.`

tmm1
/deploy github/my-feature to production

```
tmm1 is deploying github/my-feature (deadbeef) to production.
Check out the haystack firehose so you're the first to know of any new exceptions.
tmm1's production deployment of github/my-feature (deadbeef) is done! (82s)
```

Hubot
`tmm1, make sure you watch for exceptions in haystack and perf issues at graphme`

Ilustrační foto zdroj GitHUB

PROFINIT

Verzování databáze?

- Použití nástroje typu Liquibase / Flyway, …

LIQUIBASE Flyway android ROOM PERSISTENCE LIBRARY

  - Podpora verzování ve větvích a možnosti slučování změn
  - Abstrakce změn (zápis v XML, YAML, JSON nebo native SQL)
  - Možnost nastavení logiky dle kontextů → **write once deploy anywhere**
  - Možnost rozšiřitelnosti o vlastní pluginy
    (například generování rollback, auditačních triggerů, ...)

- Implementace přístupem: „od teď“ (oproti „od nuly“)
- → nový přístup k vývoji v DB:
  - Jednoduchá aktualizace lokálních databází
  - Integrační databáze slouží výhradně k integraci

PROFINIT

Verzování databáze?

```xml
<?xml version="1.0" encoding="UTF-8"?>

<databaseChangeLog
        xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
        xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.0.xsd
        http://www.liquibase.org/xml/ns/dbchangelog-ext http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-ext.xsd">

    <preConditions>
        <runningAs username="liquibase"/>
    </preConditions>

    <changeSet id="1" author="nvoxland">
        <createTable tableName="person">
            <column name="id" type="int" autoIncrement="true">
                <constraints primaryKey="true" nullable="false"/>
            </column>
            <column name="firstname" type="varchar(50)"/>
            <column name="lastname" type="varchar(50)">
                <constraints nullable="false"/>
            </column>
            <column name="state" type="char(2)"/>
        </createTable>
    </changeSet>

    <changeSet id="2" author="nvoxland">
        <addColumn tableName="person">
            <column name="username" type="varchar(8)"/>
        </addColumn>
    </changeSet>
    <changeSet id="3" author="nvoxland">
        <addLookupTable
            existingTableName="person" existingColumnName="state"
            newTableName="state" newColumnName="id" newColumnDataType="char(2)"/>
    </changeSet>

</databaseChangeLog>
```

PROFINIT

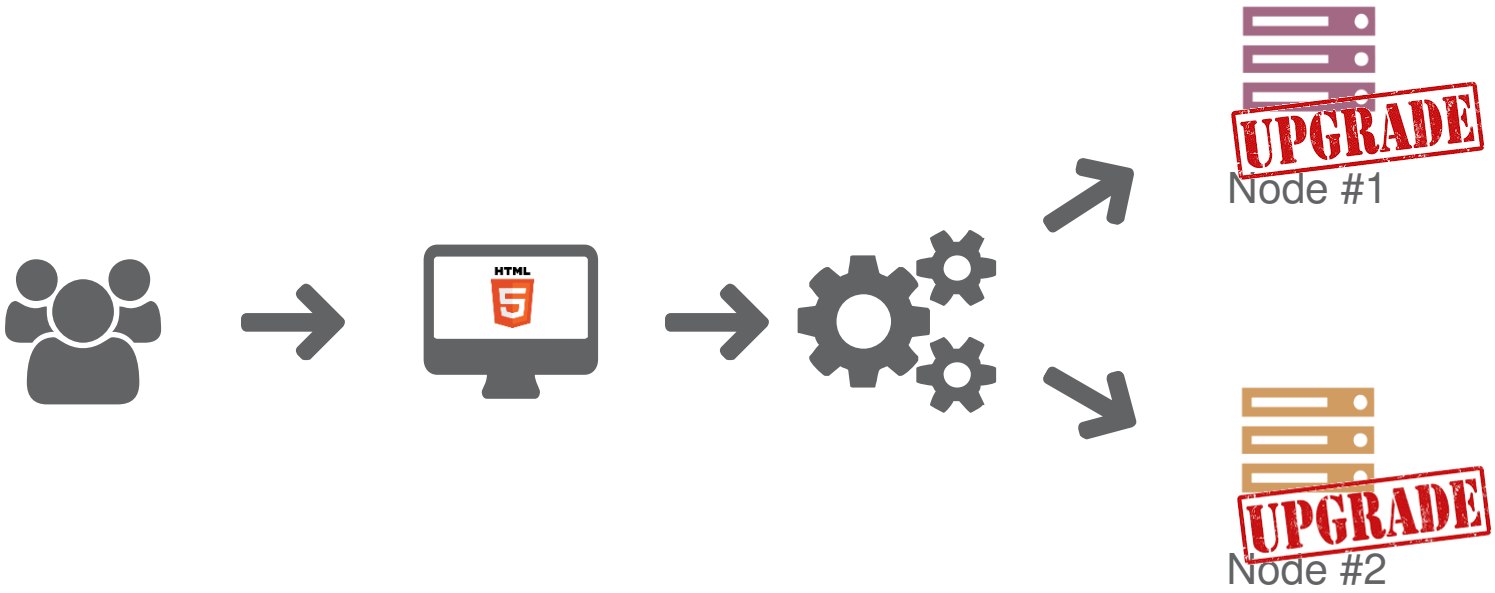
Bezodstávkové nasazování

- … všichni známe odstávky systému vzhledem k nutnosti upgrade
- … a s tím spojené nervy a občas probdělé noci

PROFINIT

Bezodstávkové nasazování

- Princip je vlastně jednoduchý….



- … nezbytnou prerekvizitou je však vyspělá automatizace

  → tzv. „**ONE CLICK DEPLOYMENT**“

PROFINIT

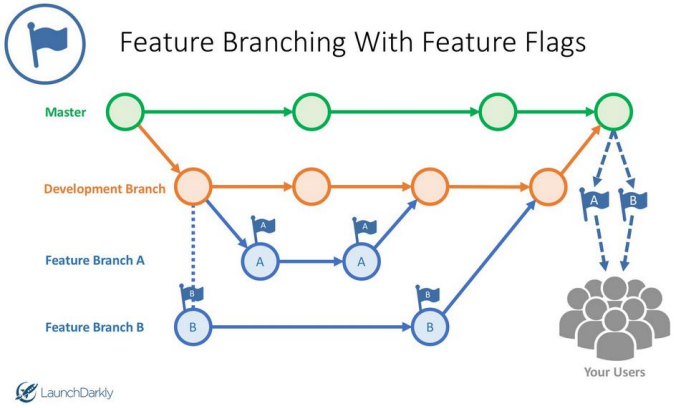
Strategie nasazování

Strategy	ZERO DOWNTIME	REAL TRAFFIC TESTING	TARGETED USERS	CLOUD COST	ROLLBACK DURATION	NEGATIVE IMPACT ON USER	COMPLEXITY OF SETUP
RECREATE version A is terminated then version B is rolled out	✗	✗	✗	■□□	■■■	■■■	□□□
RAMPED version B is slowly rolled out and replacing version A	✓	✗	✗	■□□	■■■	■□□	■□□
BLUE/GREEN version B is released alongside version A, then the traffic is switched to version B	✓	✗	✗	■■■	□□□	■■□	■■□
CANARY version B is released to a subset of users, then proceed to a full rollout	✓	✓	✗	■□□	■□□	■□□	■■□
A/B TESTING version B is released to a subset of users under specific condition	✓	✓	✓	■□□	■□□	■□□	■■■
SHADOW version B receives real world traffic alongside version A and doesn't impact the response	✓	✓	✗	■■■	□□□	□□□	■■■

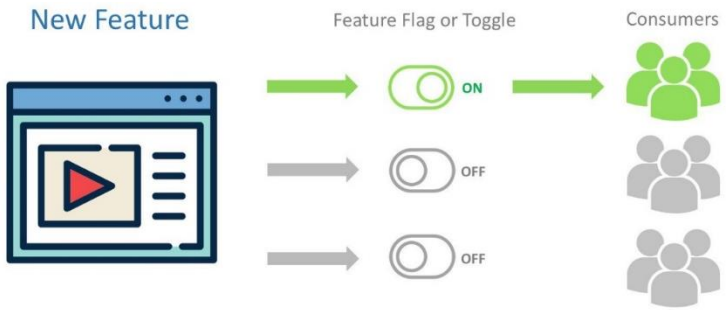
PROFINIT

Přepínače

Zpřístupnění jedné funkcionality



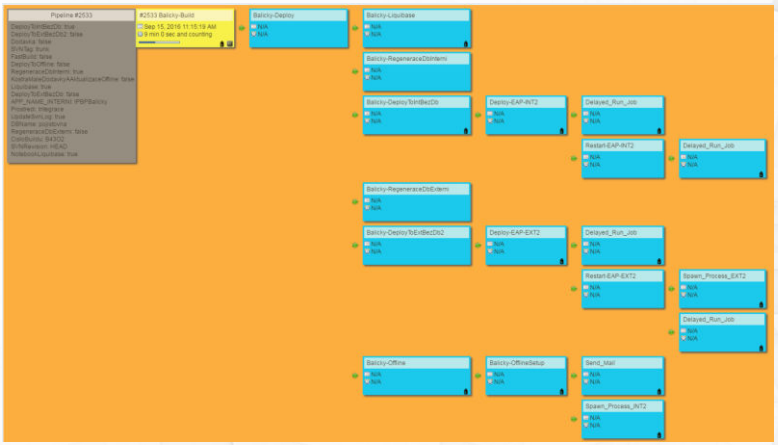
Zpřístupnění skupině lidí



PROFINIT

Chaos Monkey

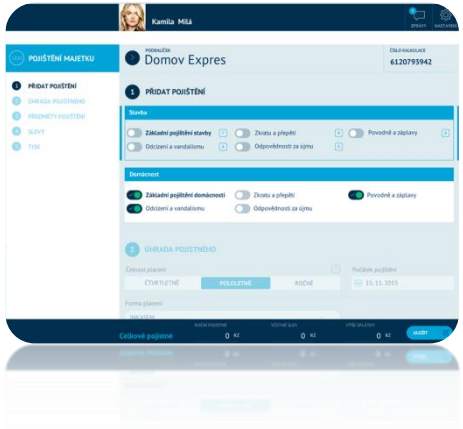
PROFINIT



Ukázky z praxe

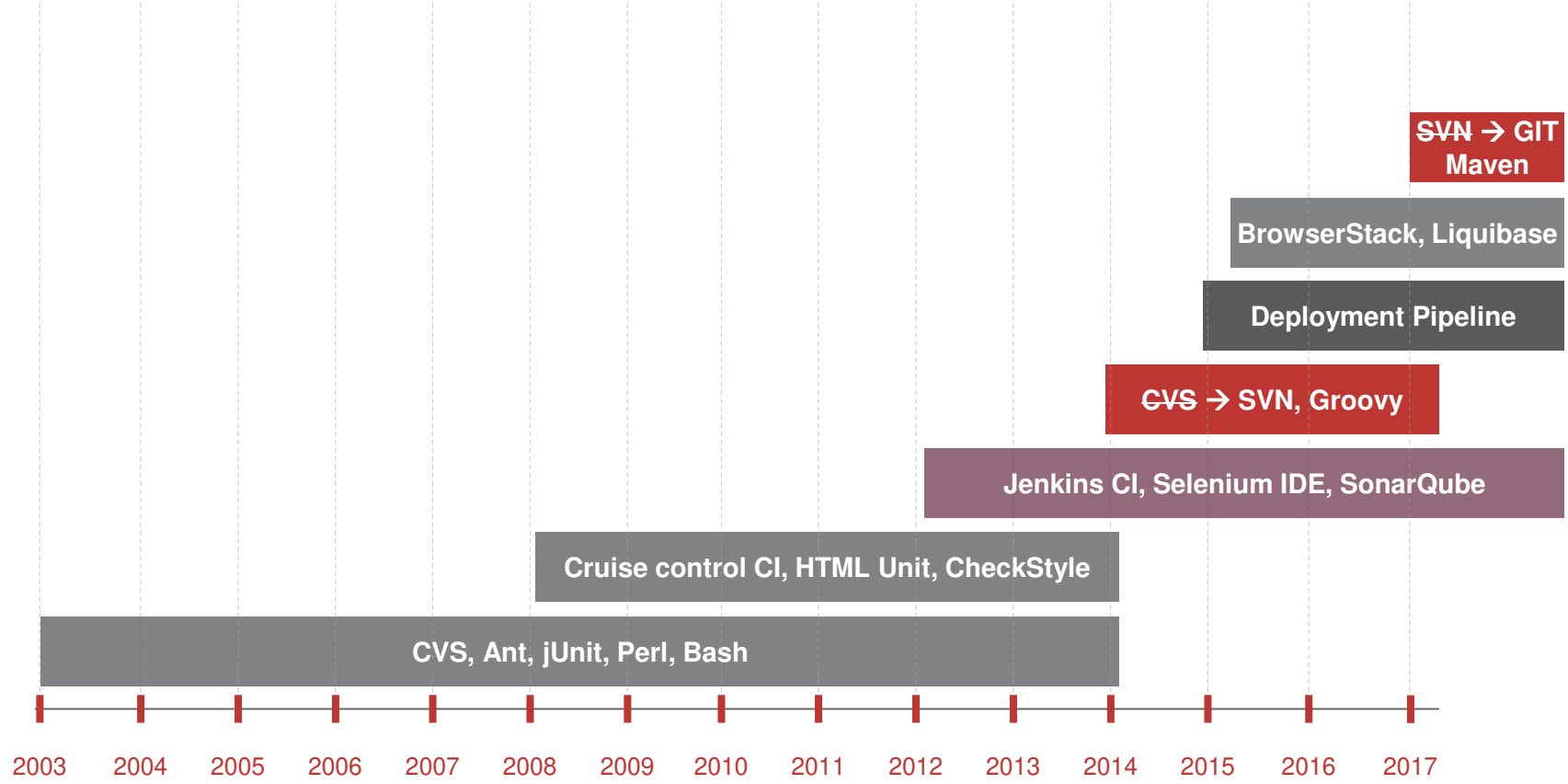
Insurance Core System

- Systém pro komplexní správu neživotního pojištění
- Technologie:
  - Java 8
  - Spring
  - Sybase ASE 15.7
  - Struts, jQuery
  - JBoss/Tomcat
- Sada nástrojů
  - Git
  - Maven
  - Liquibase
  - Jenkins
  - Groovy/Bash
  - jUnit, SonarQube
  - Selenium / BrowserStack
  - Membrane Proxy, SoapUI



PROFINIT

Insurance Core System



PROFINIT

Insurance Core System – zajímavé oblasti

Deployment pipeline

PROFINIT

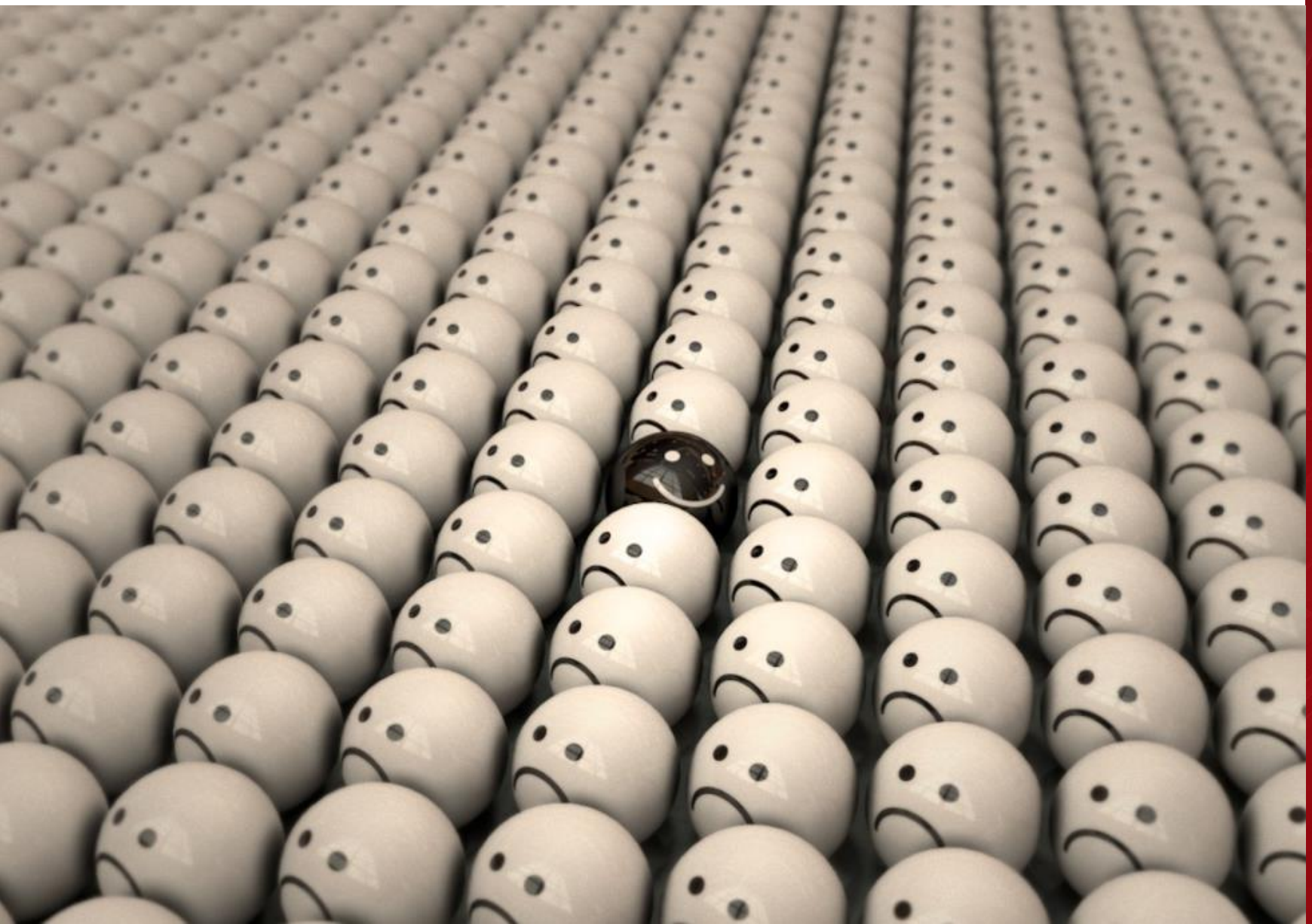
Poznatky z praxe

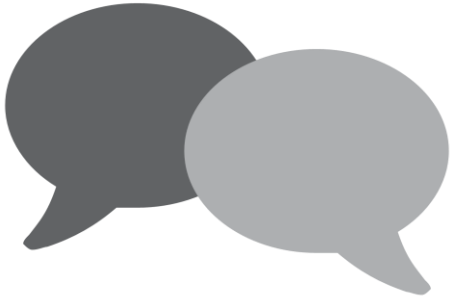
- Maximálně **věrné prostředí** vývojové, testovací, akceptační, …
- Denní build
- Proces dodávek
  - jednoduchý
  - **automatizovaný**
- Kontrolované
  - zálohovací logy
  - reporty automatických testů

- Když není k dispozici hotové řešení → vlastní microskripty/pluginy
- Nic ale není černobílé
- Velmi těžko dosažitelné bez „týmového nadšení“

PROFINIT

Poznatky z praxe





Diskuze

Děkujeme
za pozornost

PROFINIT
NÁSKOK DÍKY ZNALOSTEM

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6

Telefon
+ 420 224 316 016

Web
www.profinit.eu

LinkedIn
linkedin.com/company/profinit

Twitter
twitter.com/Profinit_EU