

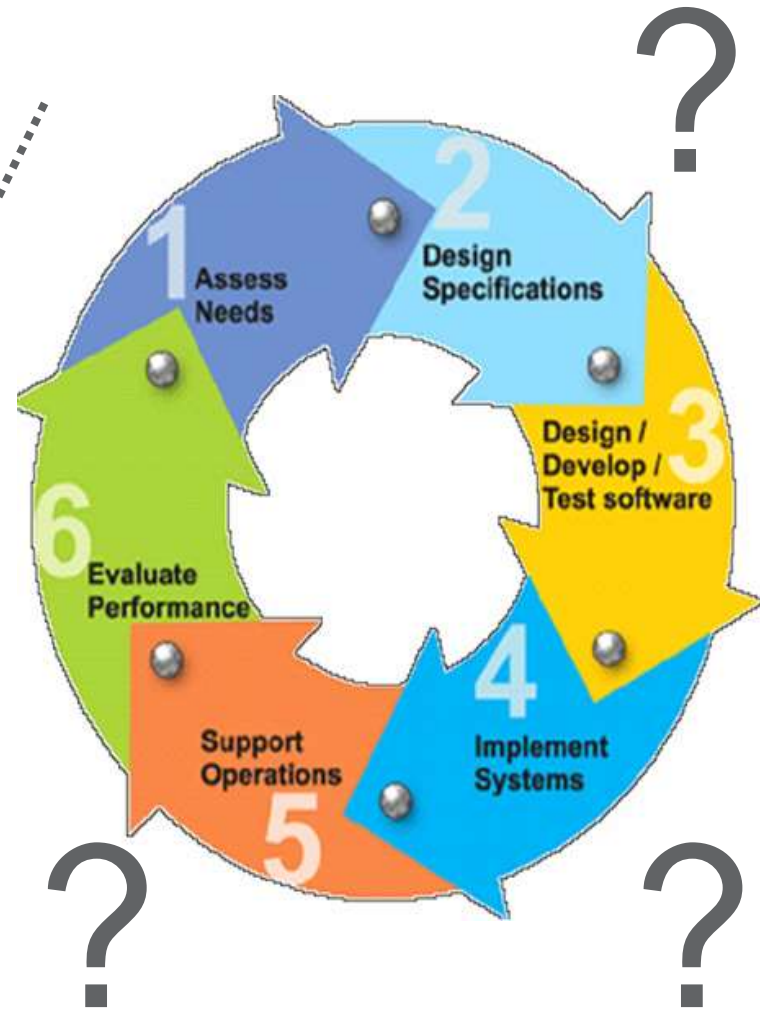
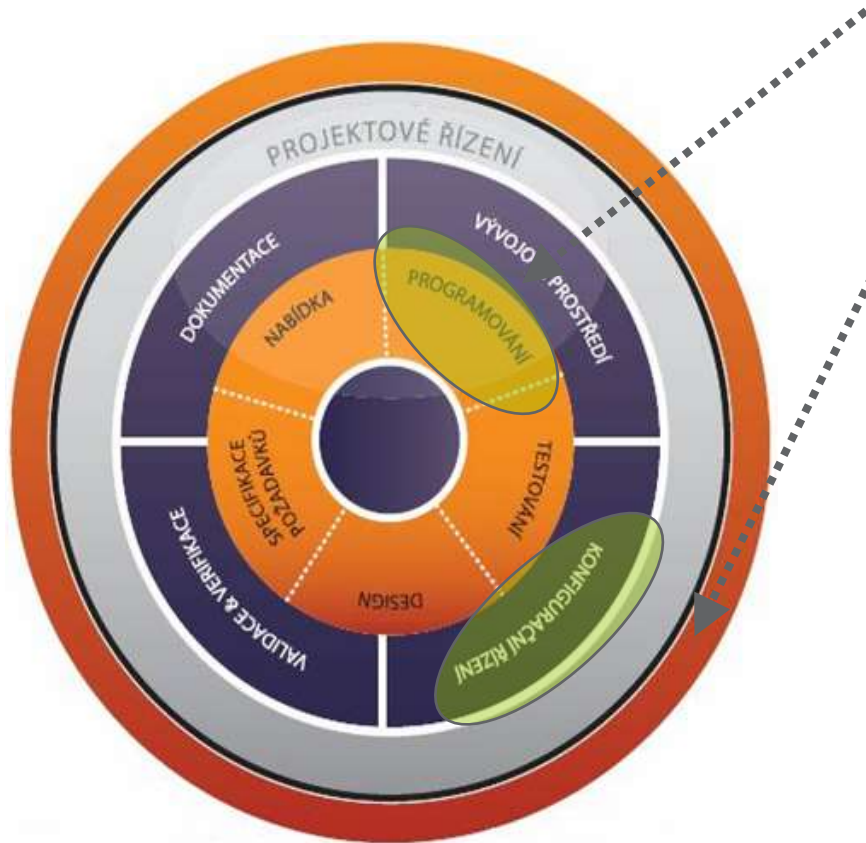
DĚLAT
DOBRÝ SOFTWARE
NÁS BAVÍ

PROFINIT

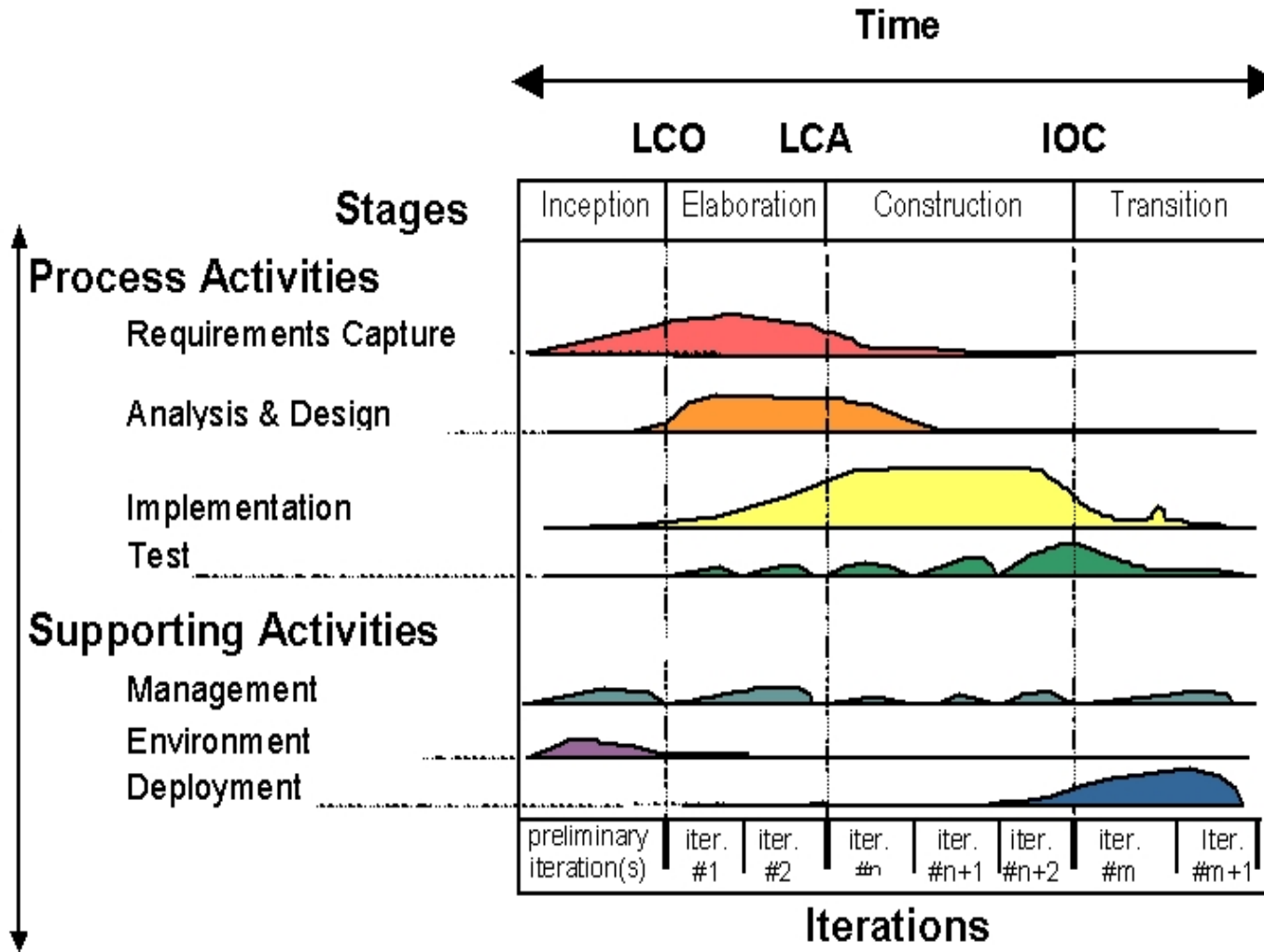
Development environment
Build process
DevOps

Tomáš Krátký, Bohumír Zoubek

Softwarový proces



Softwarový proces



Vývojové prostředí

Motivace

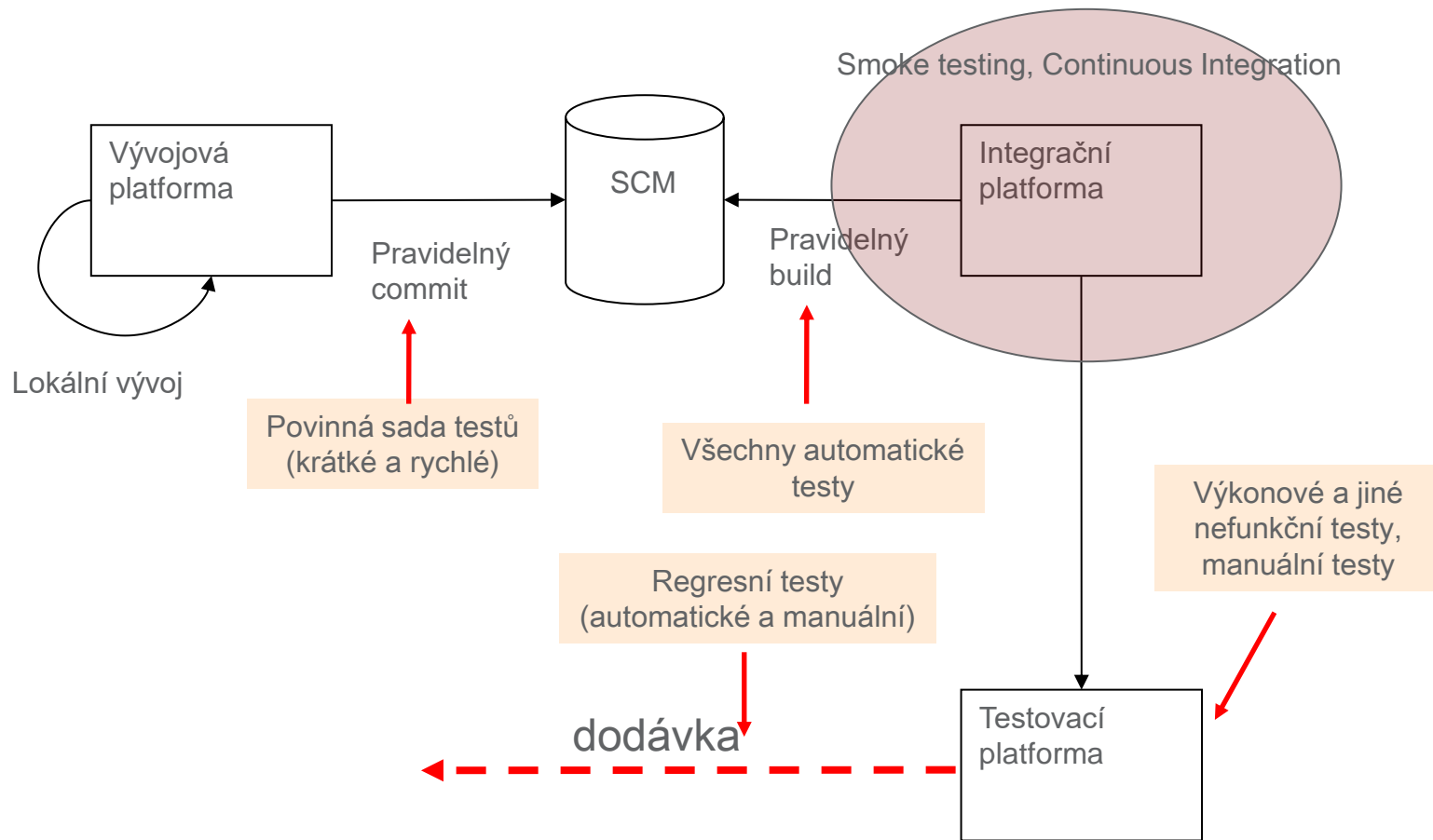
- Váš systém je určen pro cílové prostředí u zákazníka, ale ...
 - musíte ho někde vyvíjet a testovat,
 - zákazník ho musí někde před nasazením do produkce akceptačně otestovat,
 - zákazník může mít více produkčních prostředí
- ...

→ Je nezbytná existence dalších prostředí mimo cílové !

Typy prostředí

- vývojová,
- integrační,
- testovací,
- předakceptační,
- akceptační,
- produkční #1,
- produkční #2,
- ...

Continuous Integration



Dílčí úkoly

- všechny typy konfiguračních jednotek
- zálohy a jejich obnova
- evidence a zálohy (podpůrného) SW
- dodávky ("malé"/ "velké")
- postupy (instalace, modifikace, přístupy ...)
- záloha použitých zdrojových kódů
- zdrojové kódy použitých knihoven
- audit, ...

Přestanou-li existovat sdílené
disky, repository, PC lidí na
projektu, vývojové prostředí, pak

...

**ze záloh je možno postavit
vývojové prostředí**

...

a to ekonomicky.



Build proces

Souvislost s CM

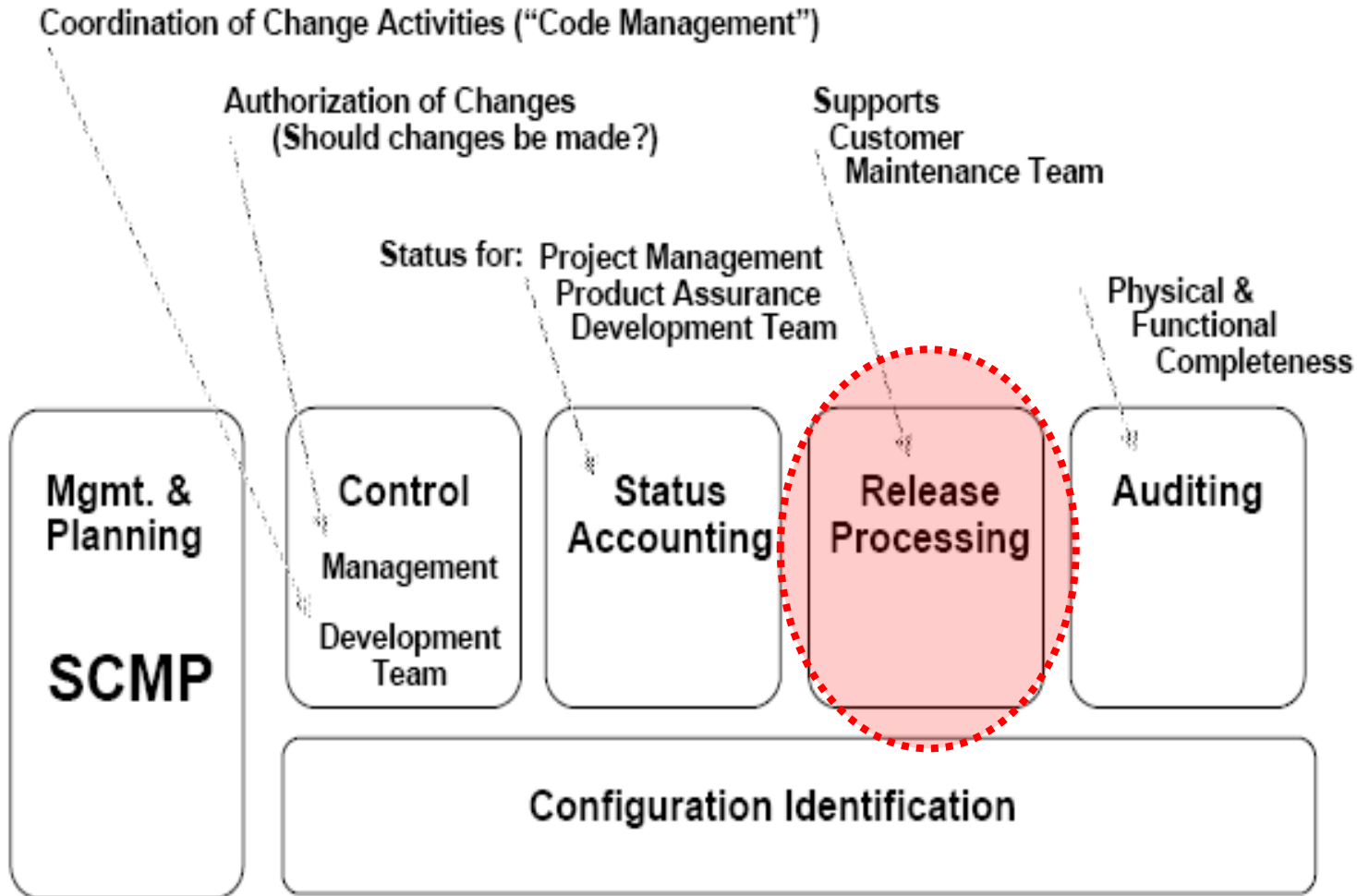


Figure 1. SCM Activities

Co musím umět

- **Vyrobít** dodávku
- **Nainstalovat** dodávku
- Připravit dodávku pro **instalaci zákazníkem**
- Dodat **system 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, ...
- ...

Podstatné pojmy

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

Poznatky z praxe

Poznatky z praxe

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





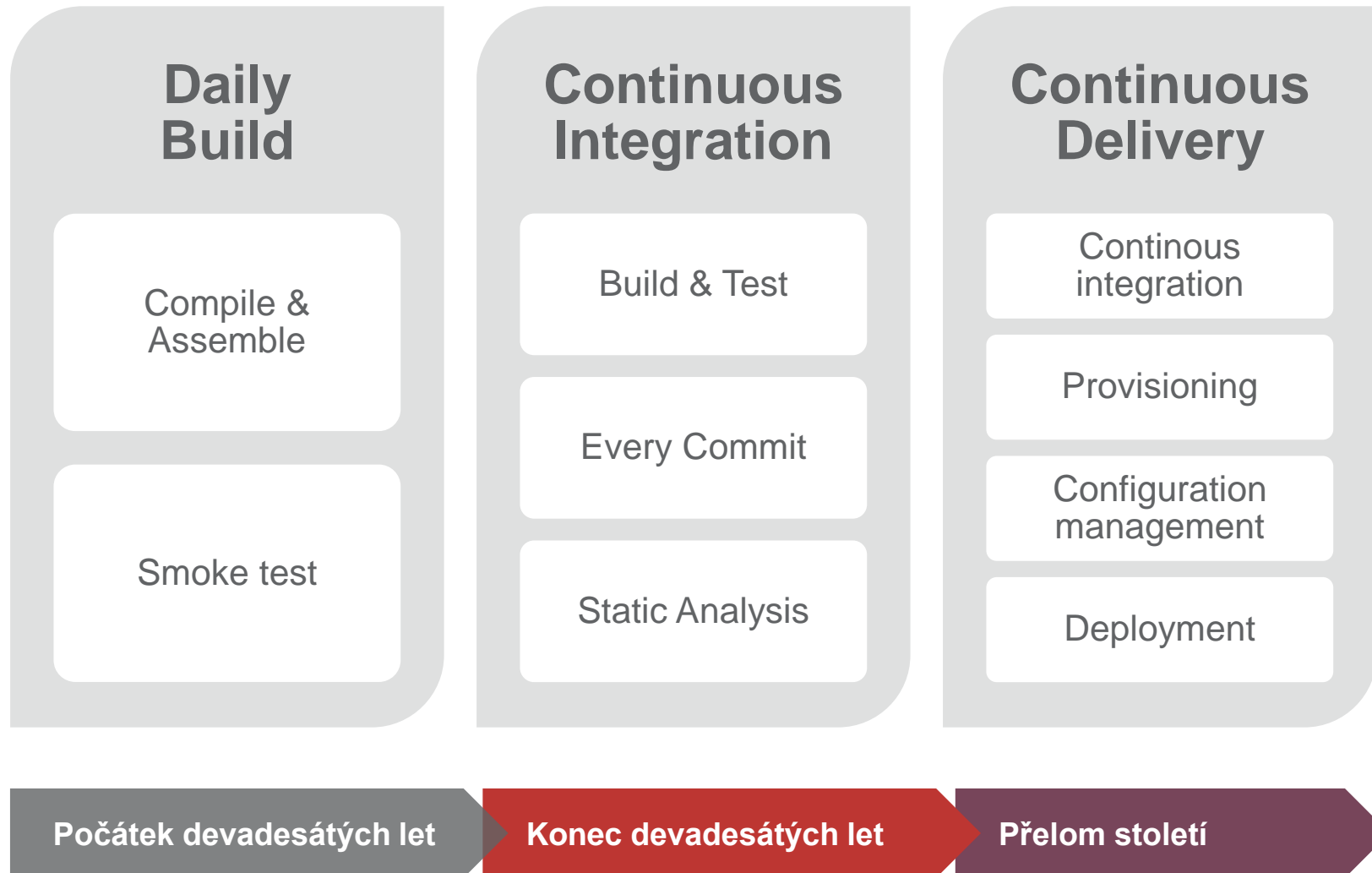
DevOps

Co je DevOps?

- › **Development & Operations**

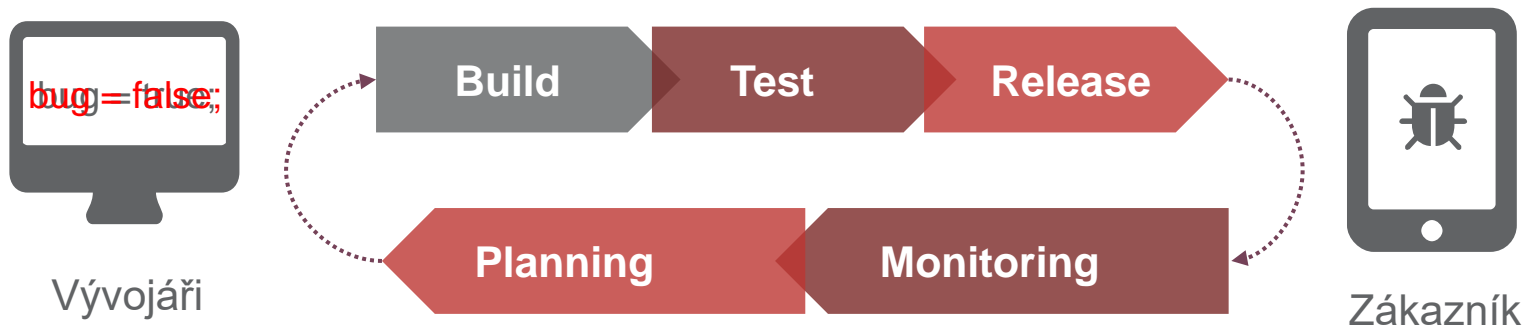


Co je DevOps?



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)

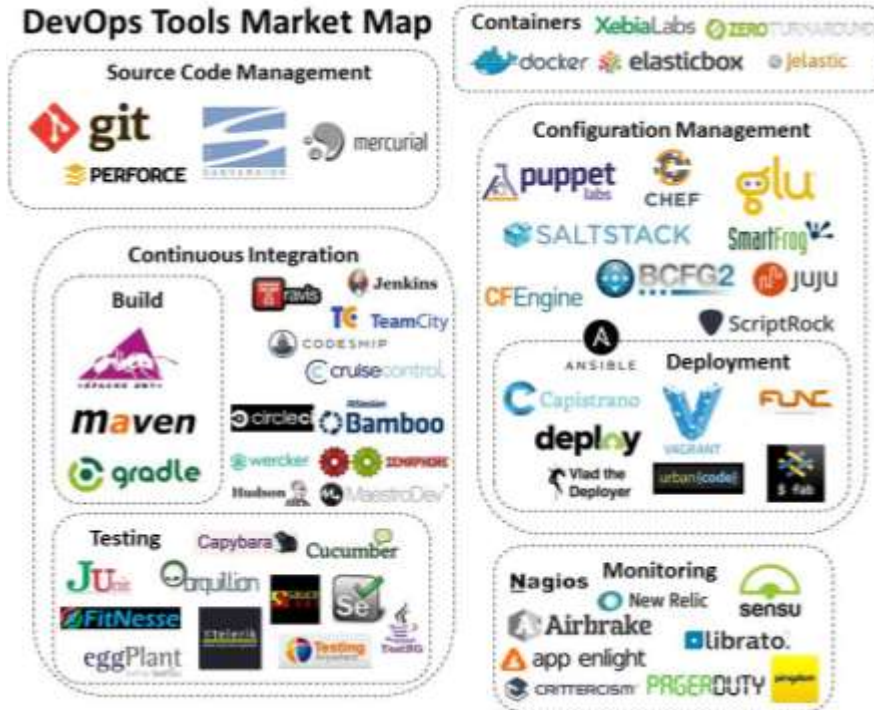


- › Je nutná odpovídající kultura ve vývojovém týmu

Co je DevOps?

Efektivní využití nástrojů

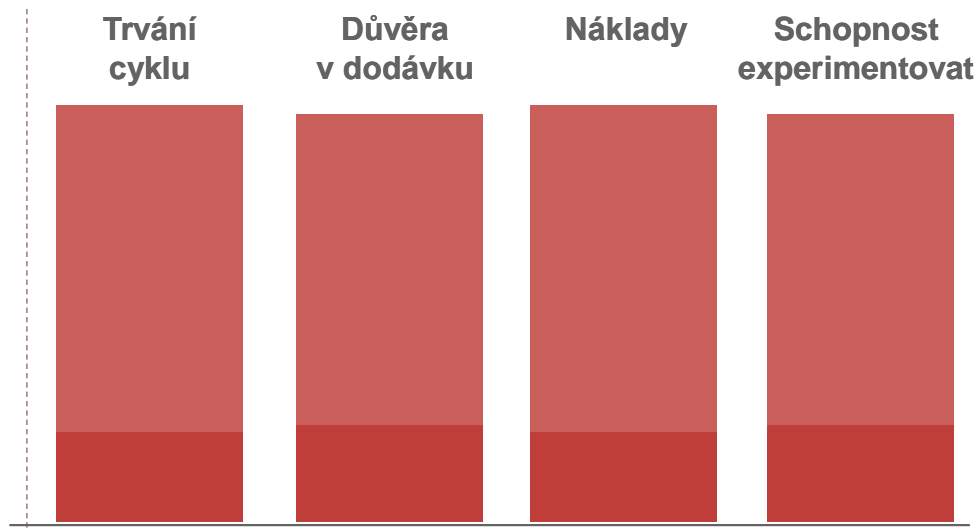
- › Version Control
- › Provisioning
- › Configuration Management
- › Build Automation
- › Artifact Repository
- › Static Analysis
- › Automated Testing
- › Test Data
- › Continuous Integration and Delivery
- › System Monitoring & Analytics
- › ... a vůbec zlepšovat



Hlavní přínosy

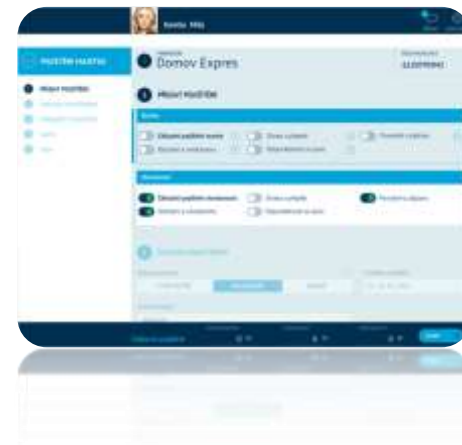


- › Snížení TTM business požadavků
- › Snížení množství chyb se současným zvýšením rychlosti jejich oprav
- › Snížení nákladů na zdroje (development i operations)
- › Jednodušší zapojení nových lidí do týmu
 - silná zpětná vazba
 - tzv. Self-service
- › Možnost „bezpečného experimentování“

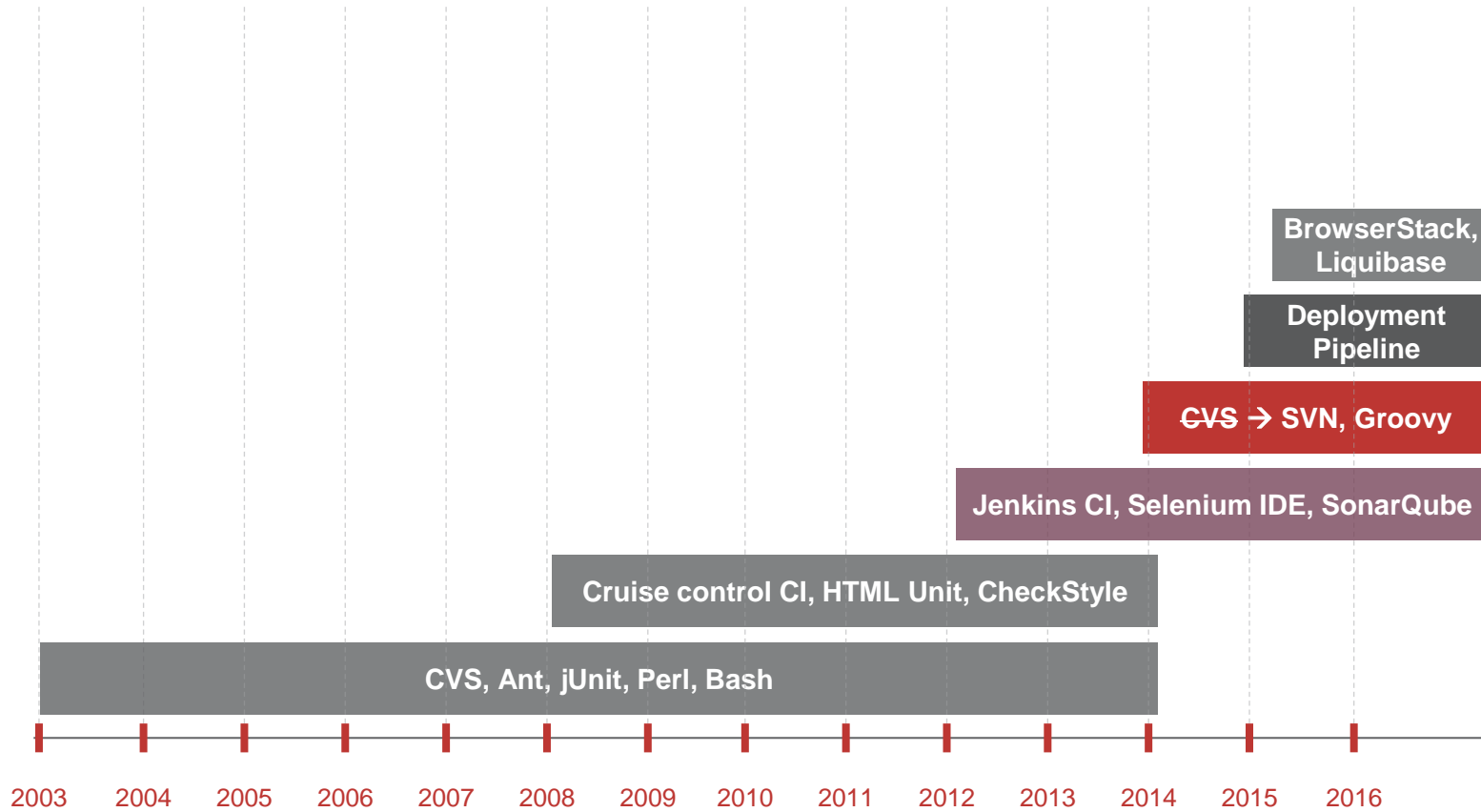


Insurance Core System

- › Systém pro komplexní správu neživotního pojištění
- › Technologie:
 - Java 7
 - Sybase ASE 15.7 a ASA 11
 - Struts, jQuery
 - JBoss/Tomcat
- › Jeden kód je použit pro tvorbu webové i offline aplikace
- › Sada nástrojů
 - SVN
 - Apache Ant
 - Liquibase
 - Jenkins
 - Groovy/Bash
 - junit, SonarQube
 - Selenium / BrowserStack
 - Membrane Proxy, SoapUI



Insurance Core System



Insurance Core System – zajímavé oblasti

Deployment pipeline

Raspberry View

Balicky-aktualizace	Balicky-AktualizatorTest	Balicky-Build
Balicky-CSV-format-test	Balicky-denniDavka	Balicky-deploy-testy-browserstack
Balicky-deploy-testy-externi	Balicky-deploy-testy-Interni	Balicky-deploy-testy-offline
Balicky-Dodavka	Balicky-Kontrola-Changesetu-Indexy	Balicky-Liquibase
Balicky-Offline	Balicky-OfflineSetup	Balicky-promazDB
Balicky-RegeneraceDbInterni	Balicky-UnitTesty	Balicky-UnitTesty-Smoke
Balicky-UpdateKonfigurace	Deploy-EAP-EXT2	Deploy-EAP-INT2
Kontrola-Dodanych-Changesetu-Liquibase	Kontrola-Offline-Updates	ReplikaceTest
ReplikaceTestQA	Test-CodeConventions	Test-Nasazenych-procedur
Test-sql-CodeConventions	Test-sql-permissions	Zaloha databazi - devices - main

Insurance Core System – zajímavé oblasti

Verzování databázového modelu

- › Použití nástroje Liquibase **LIQUIBASE**
 - 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
 - Integrovaná databáze slouží výhradně k integraci





Diskuze

Děkujeme za pozornost

PROFINIT

Profinit, 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