

Přednáška 10

Provoz softwarových systémů

CORE013 Vývoj softwarových systémů: od myšlenky k funkčnímu řešení

10. Provoz softwarových systémů

- Příprava infrastruktury
- Nasazení na infrastrukturu, po částech (releases), automatizovaně (CI/CD)
- Testovací provoz
- Zaškolení uživatelů
- Ostrý provoz
- Kontinuální rozvoj, opravování a vylepšování systému

Domácí práce a příprava na dnešní přednášku

- Pročíst si články odkázané z předchozí prezentace

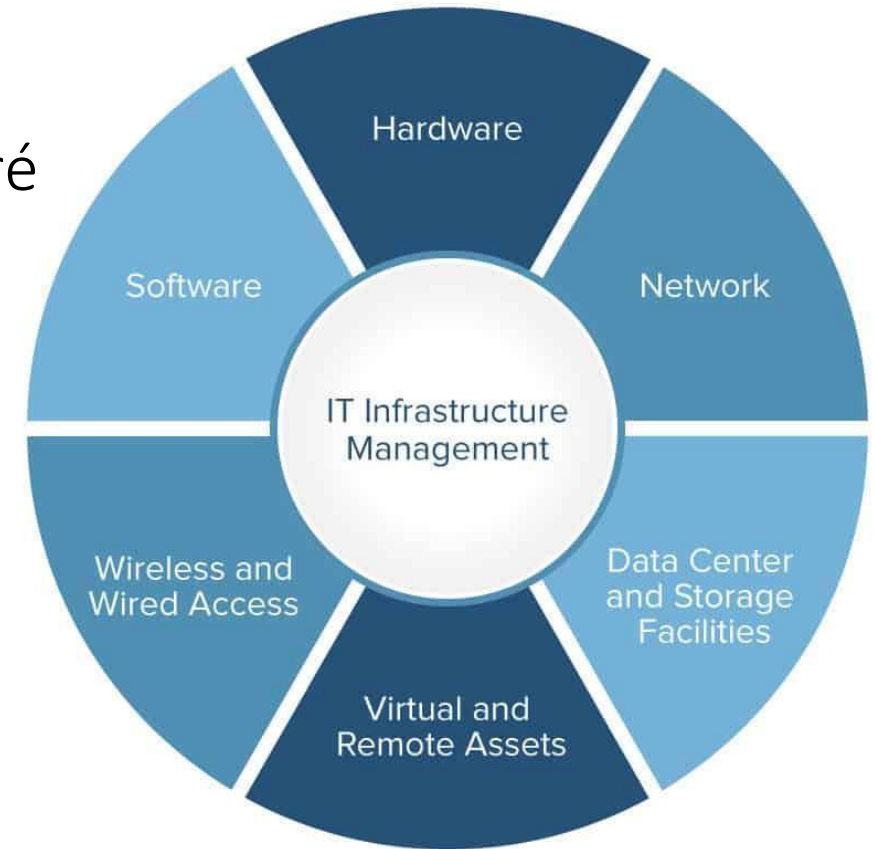
UVEDENÍ SYSTÉMU DO PROVOZU

Jaké jsou kroky uvedení systému do provozu?

- Příprava infrastruktury
- Nasazení systému na infrastrukturu
- Testovací provoz
- Zaškolení uživatelů
- Ostrý provoz
- Údržba / kontinuální rozvoj

Příprava infrastruktury

- **Infrastruktura** je souhrnné označení pro veškeré zdroje, které vyvíjený systém potřebuje pro své fungování.
- **Patří do ní:**
 - hardware (počítače, servery, routery), operační systémy, síťové systémy, datová úložiště.
- Je třeba zajistit všechny **nefunkční požadavky** (dostatečně velké úložiště, výkonné procesory).



<https://www.smartsheet.com/it-infrastructure-management-services-guide>

Nasazení systému

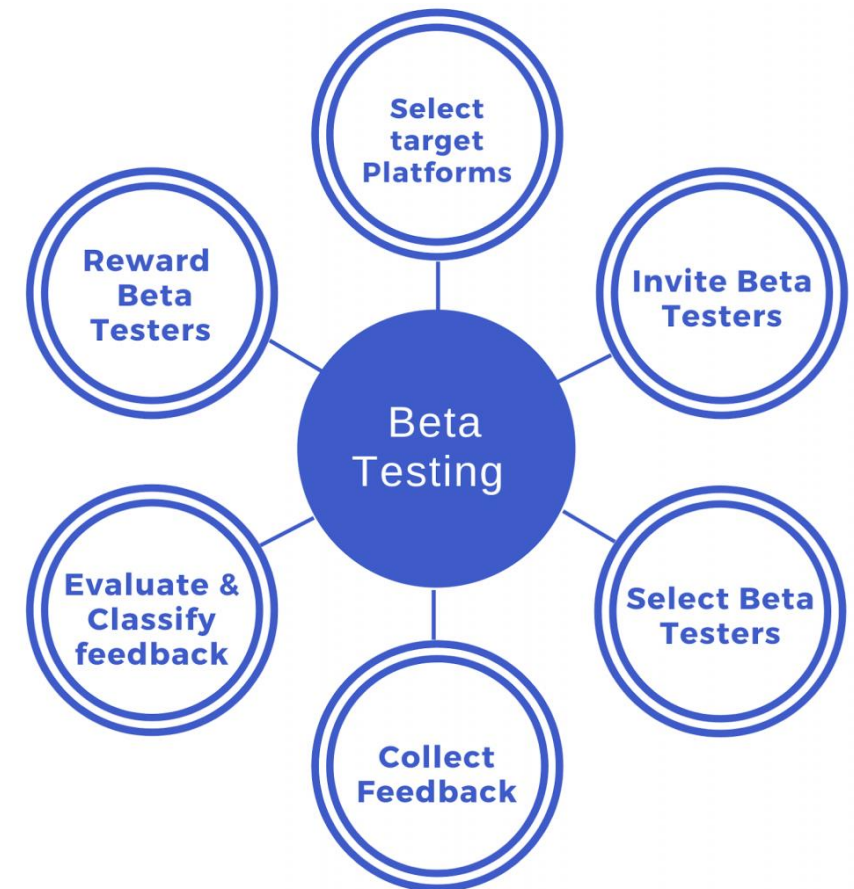
- Tento proces obsahuje veškeré **instalace** a **konfigurace** potřebné pro plynulý běh aplikace.
- Často jde o **automatizovaný proces**, při kterém aktualizace software prochází sérií automatizovaných testů a poté je nasazena přímo do ostrého provozu.
- **Schopnost rychle a plynule nasadit systém** nám umožňuje pružně reagovat na chyby a nové požadavky uživatelů.

Nasazení systému – zohlednit

- Nasazení/instalace na uživatelský počítač
 - **Lokálně** – instalace uživatelem, instalace technikem, admin přístup
 - **Vzdáleně** – nástroje vzdáleného přístupu, nutné zvolit dobu vhodnou pro uživatele, mít připravené řešení, když se instalace nepovede
 - **Automatická podpora nasazení** – podpora na úrovni operačního systému, aktualizace verze aplikace na Windows, nativně na mobilním telefonu
- Nasazení/instalace na server
 - **Jedna společná verze pro všechny** – nejsnazší na údržbu
 - **Různé verze pro různé skupiny uživatelů** – vhodné i u A/B testování, oprava chyby nezbytná ve všech stávajících verzích

Testovací provoz

- Před uvedením do ostrého provozu je třeba systém testovat v reálném prostředí.
- Testování je většinou prováděno **úzkou skupinou cílových uživatelů**, kteří dávají feedback na produkt vývojářům.
- Během testovacího provozu se opravují nalezené chyby, systém by ale měl být už kompletně funkční a stabilní.



https://www.qatouch.com/wp-content/uploads/2019/05/Beta_testing_-procedure-1006x1024.png

Zaškolení uživatelů

- Pro zvýšení efektivity práce se systémem je třeba vyškolit jeho uživatele.
- Podle případů užití můžeme cílové **uživatelé rozdělit do různých oblastí**,
 - jak budou systém používat a
 - k jakým funkcionalitám budou mít přístup.
- Uživatelé můžeme učit pomocí
 - psaných manuálů, videí, webinářů, prezenčních školení, one-to-one tréninků atd., metodu volíme dle velikosti cílové skupiny a dostupných prostředků.
- **Účastníky testovacího provozu** školíme ještě před jeho zahájením.

Zaškolení uživatelů – zohlednit

- Zaškolením může uživatele provést i aplikace sama
 - Opakovat instrukce dokud uživatel potřebuje
- **Rizika spojená se zaškolením uživatelů**
 - Uživatelé systém nepochopí, nechtějí pochopit, nedůvěřují mu
- **Plán adopce systému (přijetí a užívání)**
 - Vhodné naplánovat a sledovat
- Budování důvěry uživatele v systém

Ostrý provoz

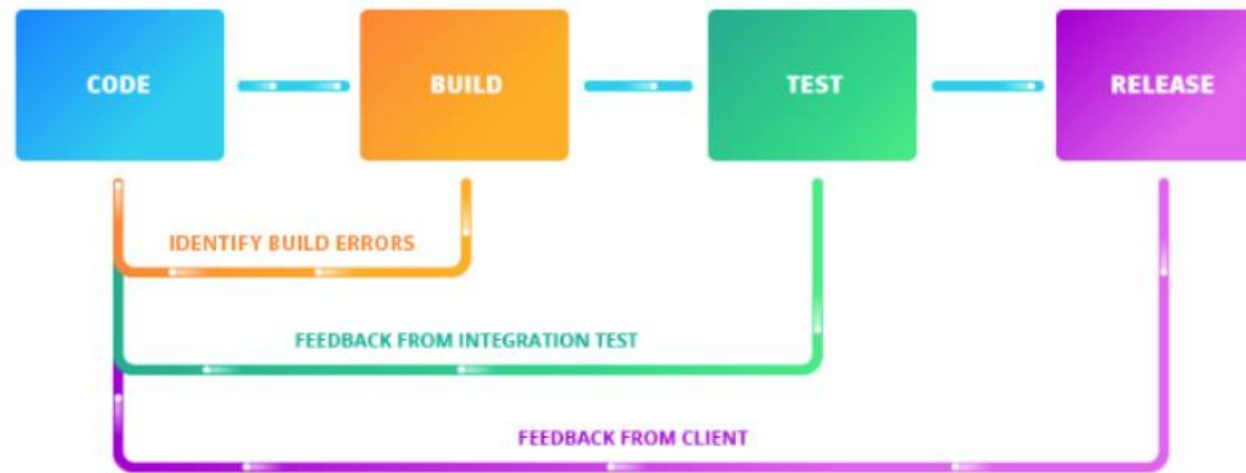
- Po testovacím provozu a vyškolení uživatelů je systém nasazen do ostrého provozu.
- V tuto chvíli je důležité **zajistit uživatelům podporu** při práci se systémem a sbírat data o případných chybách.
- Zároveň se monitoruje využívání zdrojů a je potřeba udržovat dostupnost systému.



<http://abnasia.org/marketplace/happy360-customer-support-service/>

Údržba / kontinuální rozvoj

- Během provozu systému se objevují nové chyby a podněty na zlepšení.
- Systém potřebuje dlouhodobou údržbu a aktualizace, aby udržel krok s měnícím se prostředím a požadavky uživatel.



<https://deploybot.com/blog/continuous-development>

ÚDRŽBA / KONTINUÁLNÍ ROZVOJ Produktová vs. Servisní ekonomika

ÚDRŽBA SOFTWARE

Systemy dodané na míru

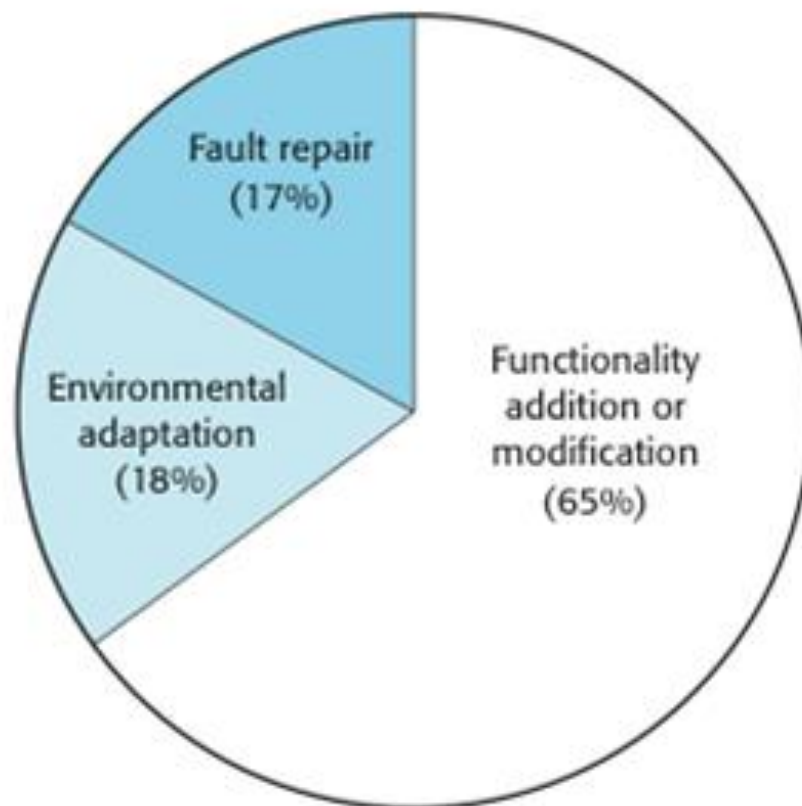
Údržba softwaru

- Úprava programu po jeho uvedení do provozu.
- Tento termín se většinou používá pro změnu **vlastního softwaru**. O obecných softwarových produktech se říká, že se vyvíjejí a vytvářejí nové verze.
- Údržba obvykle **nezahrnuje zásadní změny** architektury systému.
- Změny se provádějí úpravou stávajících komponent a přidáním nových komponent do systému.

Typy údržby

- **Náprava:** Údržba k opravě chyb softwaru
 - Změna systému za účelem odstranění nedostatků ve způsobu plnění jeho požadavků.
- **Adaptivní:** Údržba pro přizpůsobení softwaru jinému provoznímu prostředí
 - Změna systému tak, aby fungoval v jiném prostředí (počítač, operační systém atd.), než v jakém byl původně implementován.
- **Evoluční:** Údržba za účelem doplnění nebo úpravy funkcí systému
 - Úprava systému tak, aby vyhovoval novým požadavkům.

Rozdělení údržby



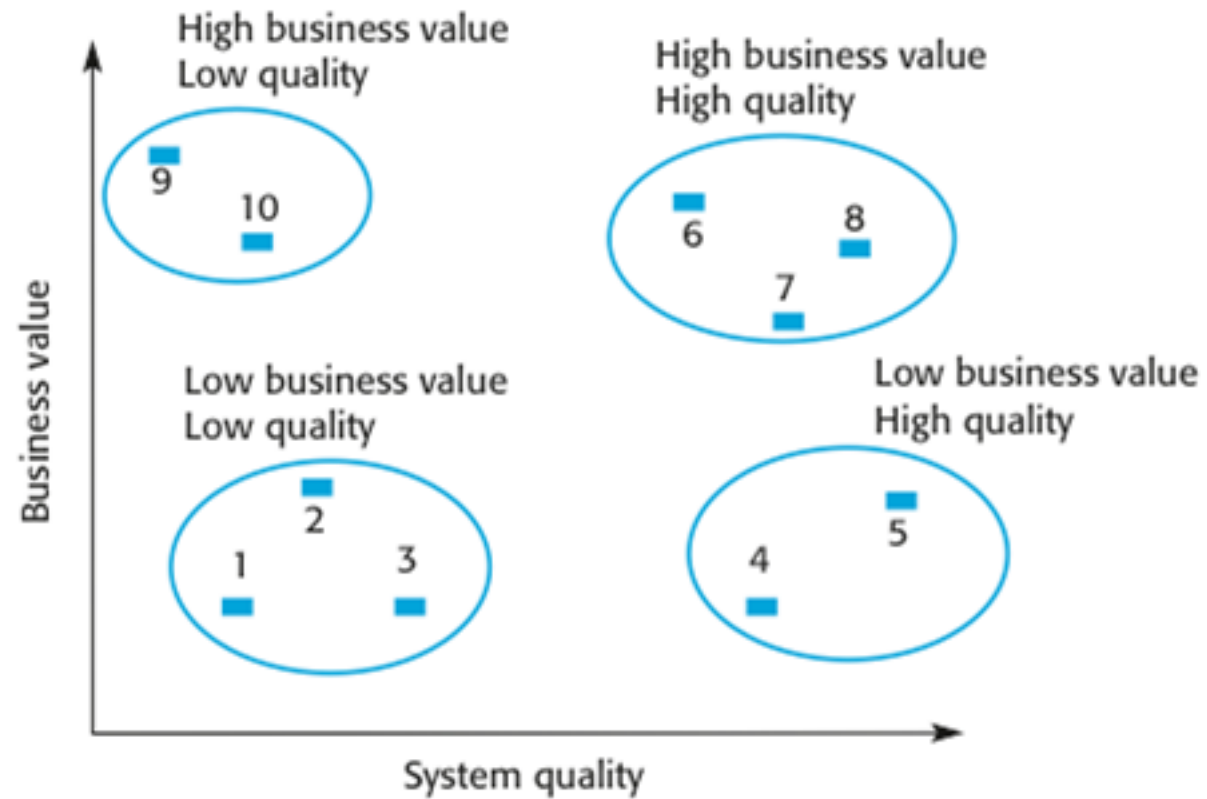
Faktory nákladů na údržbu

- **Stabilita týmu**
 - Náklady na údržbu se snižují, pokud se jim věnují stejní zaměstnanci po určitou dobu.
- **Smluvní odpovědnost**
 - Vývojáři systému nemusí mít žádnou smluvní odpovědnost za údržbu, takže nejsou motivováni k tomu, aby navrhovali budoucí změny.
- **Dovednosti zaměstnanců**
 - Pracovníci údržby jsou často nezkušení a mají omezené znalosti v dané oblasti.
- **Stáří a struktura programu**
 - Se stárnutím programů se jejich struktura zhoršuje a je obtížnější je pochopit a měnit.

Správa starších systémů

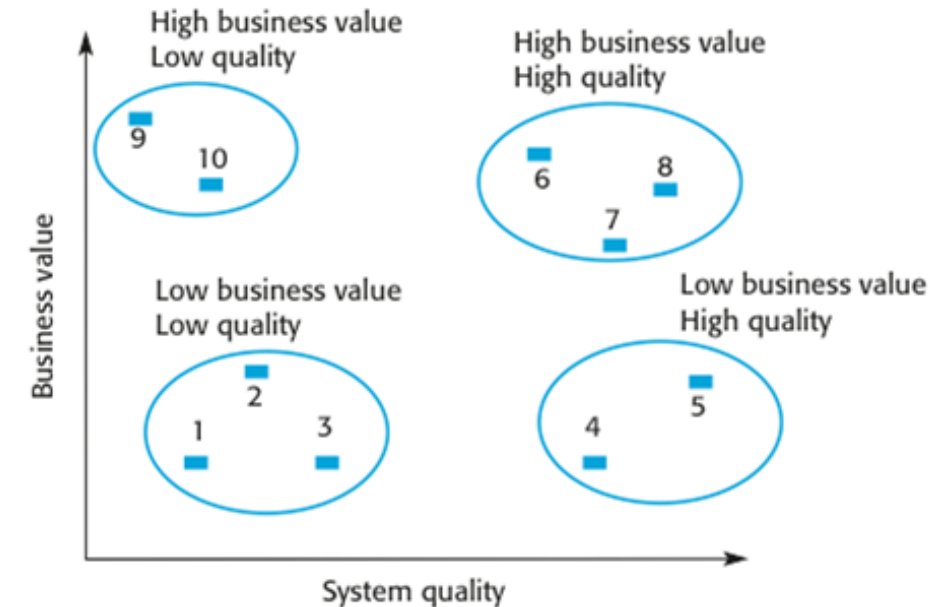
- Organizace, které se spoléhají na starší systémy, musí zvolit strategii pro vývoj těchto systémů.
 - **Odstranit:** Zcela vyřadit systém a upravit obchodní procesy tak, aby již nebyl potřeba;
 - **Udržovat:** Pokračujte v údržbě systému;
 - **Reengineer:** Transformujte systém reengineeringem, abyste zlepšili jeho udržitelnost;
 - **Nahradit:** Vyměňte systém za nový.
- Zvolená strategie by měla záviset na **kvalitě systému** a jeho **obchodní hodnotě**.

Příklad posouzení staršího systému



Posouzení staršího systému

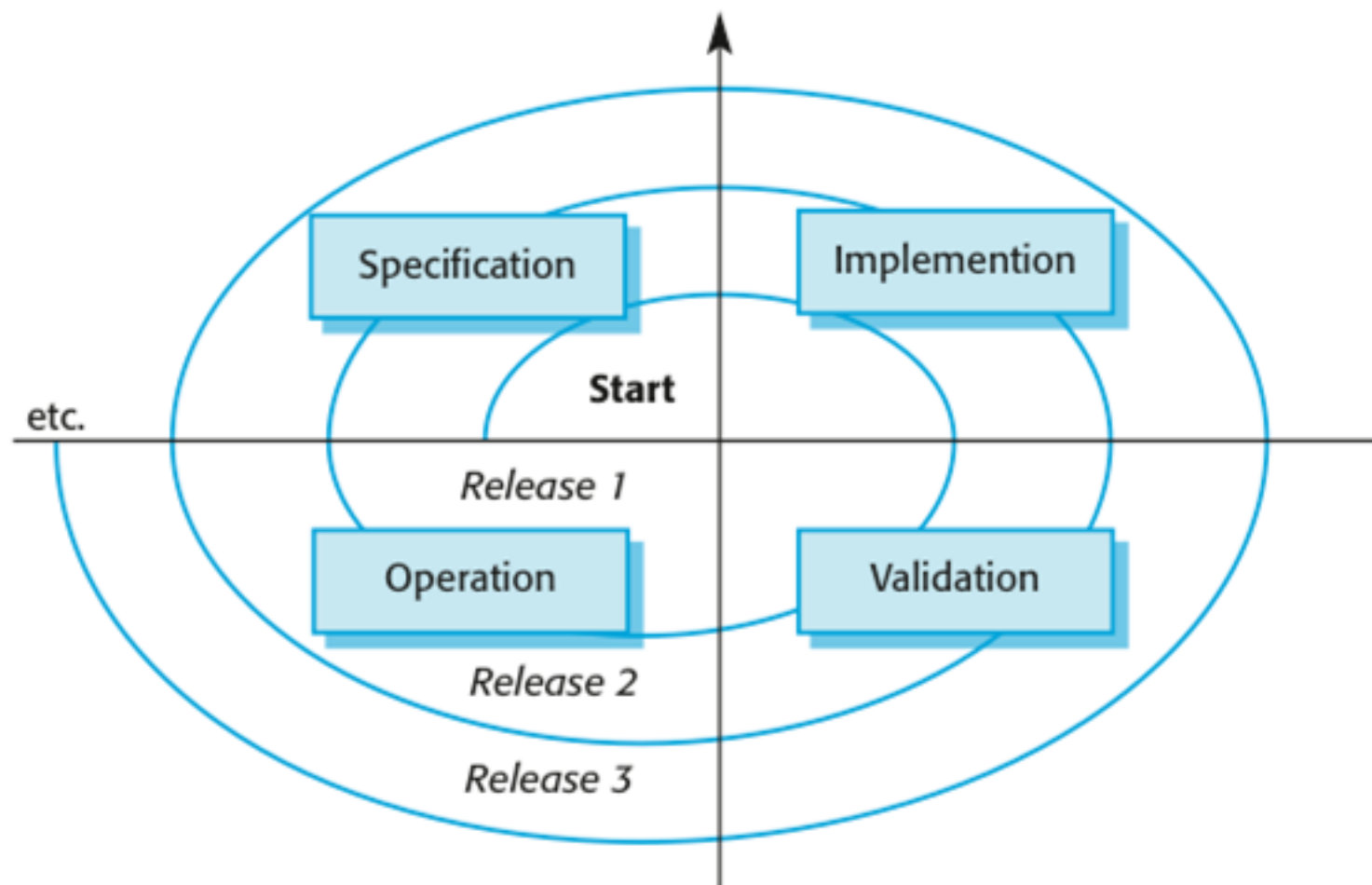
- **Nízká kvalita, nízká obchodní hodnota**
 - Tyto systémy by měly být zrušeny.
- **Nízká kvalita, vysoká obchodní hodnota**
 - Tyto mají obchodní potenciál, ale jsou nákladné na údržbu. Měly by být přepracovány nebo nahrazeny, pokud je k dispozici vhodný systém.
- **Vysoká kvalita, nízká obchodní hodnota**
 - Vyřazení nebo zvýšení obchodní hodnoty - viz pozdější slajdy.
- **Vysoká kvalita, vysoká obchodní hodnota**
 - Pokračujte v provozu pomocí běžné údržby systému.



KONTINUÁLNÍ ROZVOJ

Služby k používání širokou skupinou

Kontinuální rozvoj systému

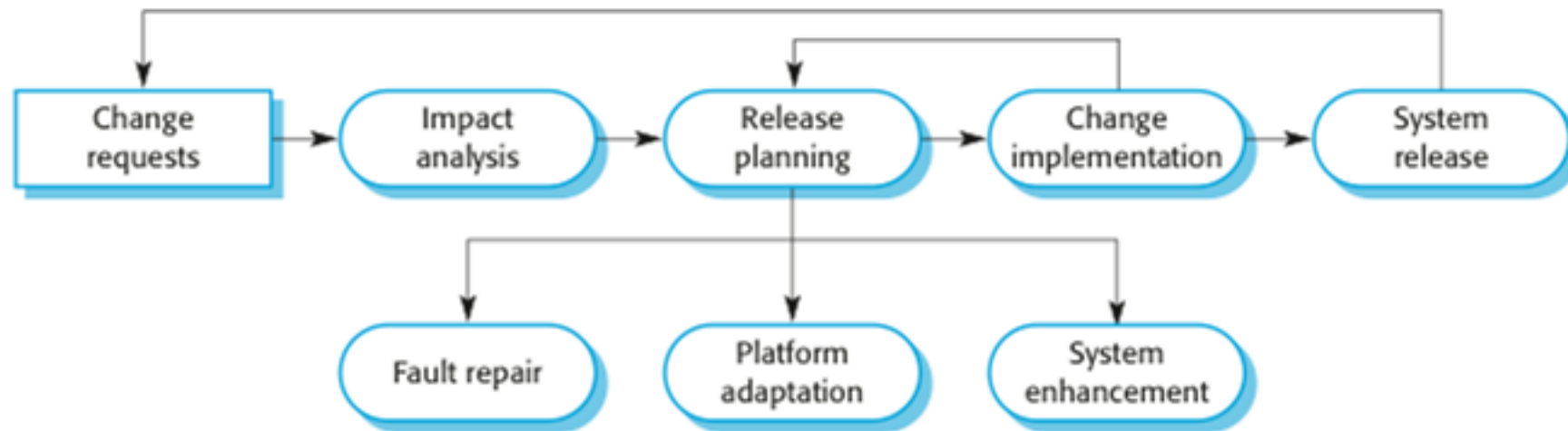


Řízení změn

- Změna softwaru je **nevyhnutelná**
 - Při používání softwaru vznikají nové požadavky;
 - Obchodní prostředí se mění;
 - Chyby je třeba opravit;
 - Do systému jsou přidávány nové počítače a zařízení;
 - Může být nutné zlepšit výkonnost nebo spolehlivost systému.
- U vlastních systémů **náklady na údržbu** softwaru obvykle **převyšují náklady na vývoj softwaru**.
- Klíčovým problémem všech organizací je zavádění a **řízení změn** ve stávajících softwarových systémech.

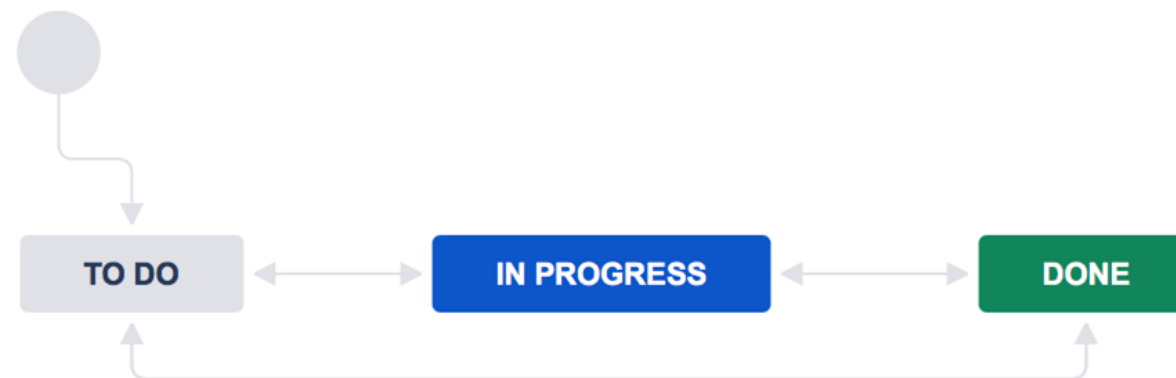
Řízení změn

- Návrhy na změny jsou hnacím motorem vývoje SW.
 - Měly by být spojeny se složkami, které jsou změnou ovlivněny, což umožní odhadnout náklady a dopad změny.
- Identifikace a vývoj změn pokračuje po celou dobu životnosti systému.



Sledování změn a problémů

- Používá se také jako nástroj pro sledování a správu softwarových projektů.
- Proces zaznamenávání a sledování průběhu softwarového problému, požadavku na změnu až do vyřešení problému.
- Nejčastěji se používá k hlášení problémů zákazníků
- Problémy jsou transformovány do tiketů, které obsahují zprávu o problému a jeho stavu.



Tabule pro sledování problémů

The image shows a Kanban board with four columns representing different stages of task completion. Each column contains several task cards. Each card includes a title, a label, status icons (checkmark, up arrow, down arrow, and a count), a due date (TIS-), and an assignee's profile picture.

Column	Task Title	Label	Status	Count	Due Date	Assignee
TO DO 5	Engage Jupiter Express for outer solar system travel	SPACE TRAVEL PARTNERS	✓ ↑	5	TIS-25	[Profile]
	Create 90 day plans for all departments in the Mars Office	LOCAL MARS OFFICE	↑ ↘	9	TIS-12	
	Engage Saturn's Rings Resort as a preferred provider	SPACE TRAVEL PARTNERS	↑ ↑	3	TIS-17	[Profile]
	Enable Speedy SpaceCraft as the preferred	SEESPACEEZ PLUS				
IN PROGRESS 5	Requesting available flights is now taking > 5 seconds	SEESPACEEZ PLUS	↑ ↑	3	TIS-8	[Profile]
	Engage Saturn Shuttle Lines for group tours	SPACE TRAVEL PARTNERS	✓ ↑	4	TIS-15	[Profile]
	Establish a catering vendor to provide meal service	LOCAL MARS OFFICE	🔧 ↑	4	TIS-15	[Profile]
	Engage Saturn Shuttle Lines for group tours	SPACE TRAVEL PARTNERS				
CODE REVIEW 2	Register with the Mars Ministry of Revenue	LOCAL MARS OFFICE	↑ ↑	3	TIS-11	
	Draft network plan for Mars Office	LOCAL MARS OFFICE	✓ ↑	3	TIS-15	[Profile]
DONE 8	Homepage footer uses an inline style - should use a class	LARGE TEAM SUPPORT	↑ ↑		TIS-68	[Profile]
	Engage JetShuttle SpaceWays for travel	SPACE TRAVEL PARTNERS	↑ ↑	5	TIS-23	[Profile]
	Engage Saturn Shuttle Lines for group tours	SPACE TRAVEL PARTNERS	✓ ↑		TIS-15	[Profile]
	Establish a catering vendor to provide meal service	LOCAL MARS OFFICE				

Další informace o provádění změn

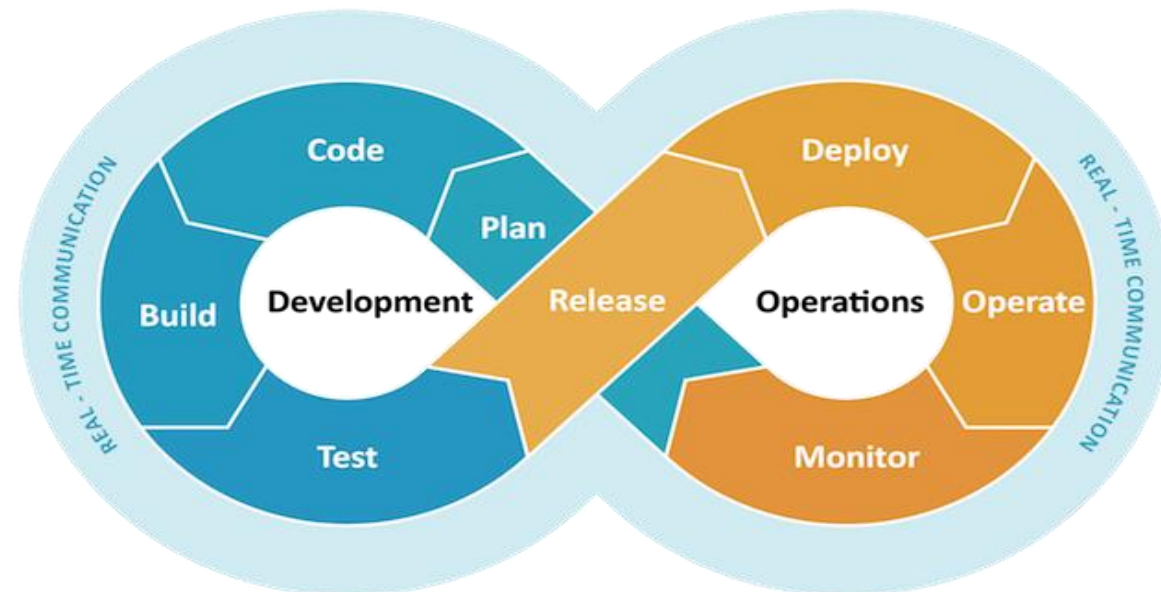
- Iterace vývojového procesu, kdy se navrhují, implementují a testují revize systému.
 - Může být provedeno **jiným týmem**, nikoli původními vývojáři.
 - Zahrnuje **porozumění programu**, zejména pokud se na něm nepodílejí původní vývojáři.
 - Ve fázi porozumění programu je třeba pochopit, **jak by navrhovaná změna mohla program ovlivnit**.
- Může se stát, že bude nutné provést **naléhavé změny**, aniž by prošly všemi fázemi vývojového procesu.
- **Technický dluh** může vzniknout a musí být řízen.

KONTINUÁLNÍ INTEGRACE/NASAZENÍ

Continuous Integration/Delivery CI/CD

DevOps

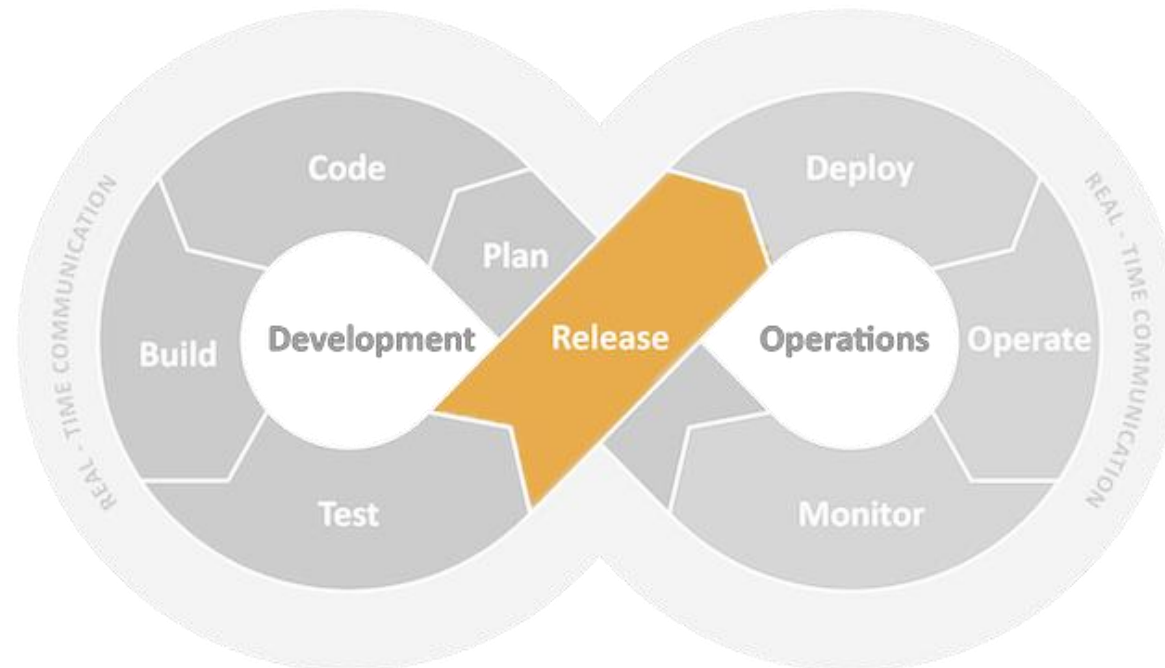
- Soubor postupů, který kombinuje vývoj softwaru (Dev) a provoz (Ops) a jehož cílem je zkrátit životní cyklus vývoje a zajistit kontinuální dodávku s vysokou kvalitou softwaru.
- Kultura podporující spolupráci mezi vývojovým a provozním týmem, která zvyšuje efektivitu týmu.



<https://www.ecloudvalley.com/wp-content/uploads/2018/08/What-is-DevOps-1.png>

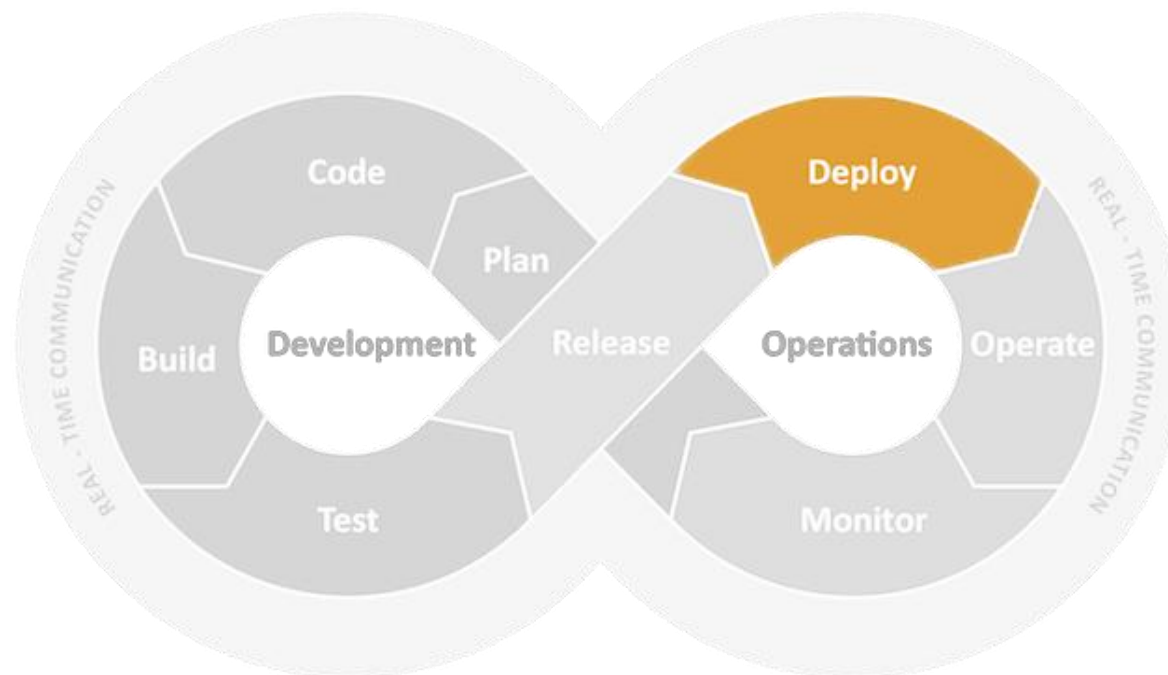
Vydání

- Distribuce verze softwaru
- Na konci každé iterace v agilních metodikách
- Software musí být naplánován, implementován, otestován a následně dodán zákazníkovi



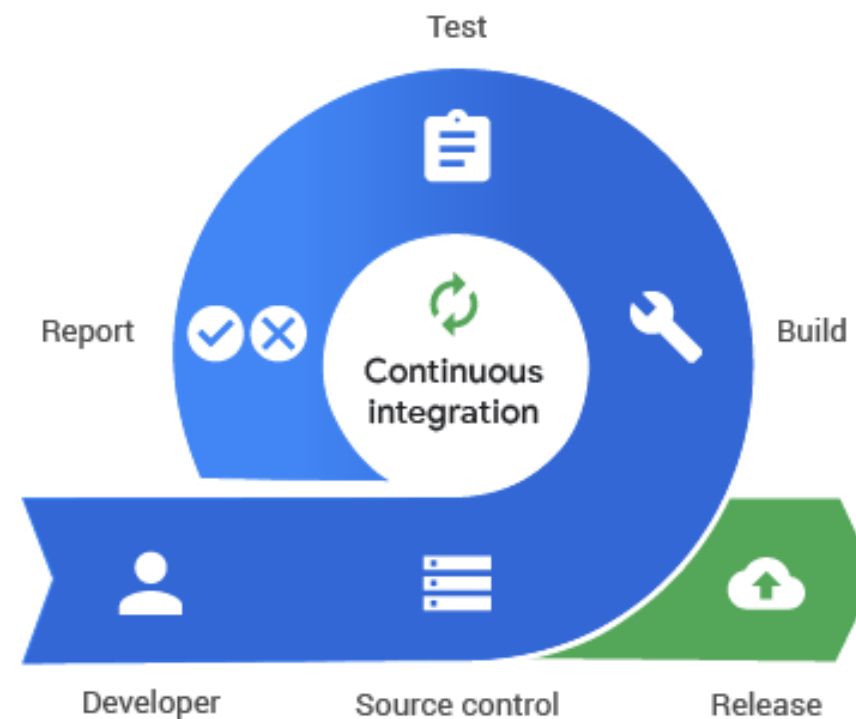
Nasazení

- Proces přípravy softwarové aplikace na běh a provoz v určitém prostředí.
- Plán nasazení, který zajistí, že změny budou pokaždé provedeny stejným způsobem.
- Různá prostředí:
 - Místní
 - Vývoj
 - Inscenace
 - Výroba



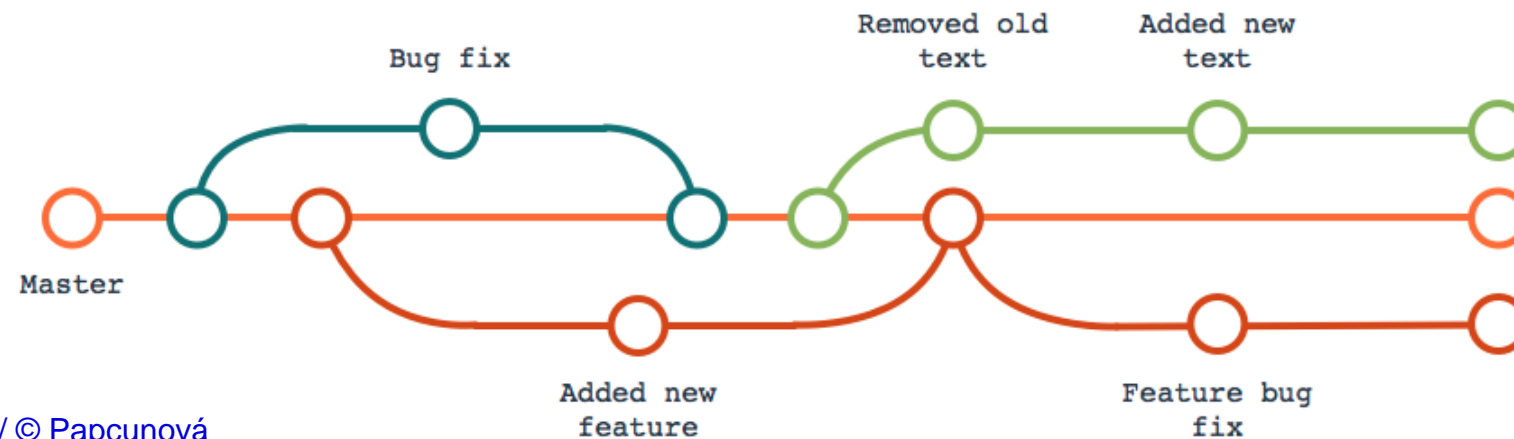
Kontinuální integrace

- Vývojová praxe, která vyžaduje, aby vývojáři pravidelně integrovali kód do sdíleného úložiště.
- Proces automatizace sestavování a testování kódu pokaždé, když člen týmu odevzdá změny do správy verzí
- Proces může pomoci rychle odhalit chyby a snadno je odstranit.



Řízení verzí

- Systém, který zaznamenává změny v souborech v průběhu času, aby bylo možné později vyvolat konkrétní verze.
- Umožňuje porovnávat změny a vracet soubory do předchozího stavu.
- Kód je uspořádán ve stromové struktuře - vývojáři mohou pracovat na některých částech tohoto stromu a později mohou své změny sloučit dohromady



CO NÁS ČEKÁ PŘÍŠTĚ

11. Trendy a budoucnost IT

- Trendy a budoucnost, příležitosti a hrozby
- Automatizace
- Robotika a IoT
- Big data and BI
- AI and bots

Domácí práce a příprava na příští přednášku

- Najděte a nasdílejte inspirativní článek k tématu do diskuzního fóra

https://is.muni.cz/auth/discussion/predmetove/fi/podzim2022/CORE013/trendy_a_budoucnost_it/