

Týmová spolupráce, komunikace, sdílení dat v týmu, základy návrhu a plánování projektu

Jaroslav Dytrych

Fakulta informačních technologií Vysokého učení technického v Brně
Božetěchova 1/2. 612 66 Brno - Královo Pole
dytrych@fit.vutbr.cz



26. února 2025

- Komunikace v týmu
- Prostředky pro sdílení dat v týmu
- Plánování projektu
- Úvod do návrhu software

Komunikace v týmu

- Ve škole i v budoucím zaměstnání se můžete dostat do situace, kdy musíte sestavit tým lidí k řešení projektu.
- Ne vždy si však můžete zvolit členy svého týmu podle svých představ a ne vždy máte možnost ihned zjistit, který člen je zkušený a pracovitý a který nikoliv.
- Tým lidí je často ustaven pomocí elektronických prostředků a členové nikdy nebudou sedět ve společné kanceláři.
- Vhodným přístupem a komunikací můžete omezit negativní dopad činnosti „nepříliš dobrých“ členů týmu.
 - Ihned po sestavení týmu je nutné zjistit informace o jeho neznámých členech.

- Velké firmy mají obvykle pravidla a standardy pro vnitřní komunikaci.
 - v každé firmě jiné,
 - firma každého nového zaměstnance zaškolí.
- Ve škole, malé soukromé firmě či při práci na živnostenský list je situace složitější:
 - každý, s kým máte spolupracovat, může mít jiné zvyklosti,
 - každý může preferovat jiné komunikační kanály,
 - někdo nemusí být ochoten využívat Vámi preferovanou komunikaci,
 - **je třeba dojít ke kompromisu a dohodnout společné komunikační kanály a pravidla pro komunikaci.**

- Máte zadán školní týmový projekt do povinného předmětu.
- Každý tým musí mít 3–4 osoby.
- Máte 1 kamaráda a s nimi tedy můžete sestavit pouze dvoučlenný tým.
- Na zadání se přihlašuje přes IS a do týmu se Vám tedy může přihlásit kdokoliv další.
 - Přihlásili se 2 neznámí studenti.

- Cílem je dohodnutí osobní schůzky před zahájením projektu a ustanovení stabilních komunikačních kanálů.
- Lze využít dostupné prostředky, což je často pouze e-mail či v lepším případě telefon.
- Lze využít i jednoduché webové služby, které mohou urychlit komunikaci:
 - pro dohodu na termínu lze využít specializované nástroje, jako např. Doodle (<http://www.doodle.com/>), Framadate (<https://framadate.org/>) nebo NeedToMeet (<http://www.needtomeet.com/>),
 - pro složitější dohadování lze rozeslat odkaz na zaheslované fórum či chat, kde se všichni dohodnou,

Hlasování - IVS - obhajoby

IVS - obhajoby

 Tisk

 Export do CSV

Creator of the poll

Jaroslav Dytrych

Public link to the poll

<https://framadate.org/FXbfe9Ve3Syc7zyi>

Votes





May 2025

Tue 6

Wed 7

Thu 8

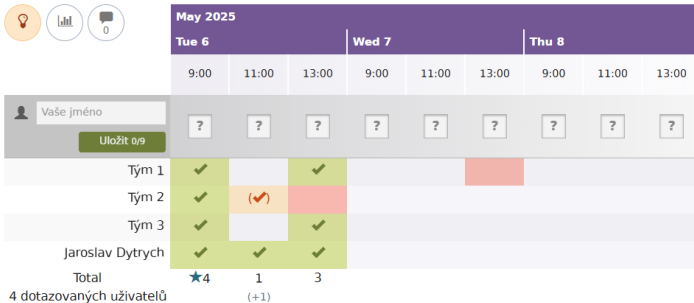
9:00	11:00	13:00	9:00	11:00	13:00	9:00	11:00	13:00
?	?	?	?	?	?	?	?	?



Uložit 

Uložit

Votes



Nejlepší volba

★ The current best choice is:

Tuesday, May 6, 2023 - 9:00



s 4 Hlasy.

- Někdo musí začít komunikovat.
 - Buď začne jeden aktivní jedinec, všichni nebo nikdo.
 - Začít by měli všichni, jinak se může stát, že se dlouho nebude nic dít.
- Každý by měl napsat e-mail všem ostatním, poslat detailnější kontaktní informace, nabídku termínů schůzky či odkaz na Framadate, případně nabídnout lepší komunikační kanál (např. fórum, které spravuje), je-li něco potřeba dohodnout před termínem schůzky.
 - Z členů týmu A, B, C a D byl nejrychlejší C. Všem zaslal svůj Jabber, telefonní číslo a časy, ve které by příští týden mohl přijít na schůzku.
 - Následně odpověděl B, ale poslal jenom Jabber a jeden z termínů od C.
 - A na to reagoval a poslal Jabber, časy, odkaz na Framadate a svůj Facebook.

- Každý by měl odpovědět nejméně na e-mail od pana C (v kopii zaslat všem ostatním – funkce „odpovědět všem“) a reagovat na e-mail od pana A (nejlépe hlasováním na Framadate).
- Následná komunikace by měla pokračovat na nejlepším dostupném komunikačním kanálu, v případě stejných na dříve nabídnutém.
 - Všichni by měli využít Framadate.

- Pokud je to možné, měli by být všichni osobně přítomni.
- Je třeba dohodnout:
 - kdo bude vedoucím týmu (hlídání termínů, komunikace se zákazníkem, ...),
 - komunikační kanály,
 - pravidla pro komunikaci,
 - prostředky pro sdílení dat v týmu,
 - ...

Komunikační kanály

- telefon
- e-mail
- instant messaging
- VoIP / Skype / Google Chat
- diskusní fóra a chaty
- ...

- Proč dávat někomu neznámému číslo?
 - Ostatní členové týmu by měli mít možnost Vám zavolat pokud nedorazíte na domluvenou schůzku, spletete si termín obhajoby (30 min. předem stále nebudete na místě) apod.
 - Nedůvěra v seriózní jednání ostatních vytváří v týmu nepříjemnou atmosféru.
- K čemu jej využívat?
 - pouze pro neodkladnou komunikaci.
 - Ne každý je ochoten tolerovat časté obtěžování telefonem.

- nedostupnější komunikační prostředek.
- Odpověď může určitou dobu trvat:
 - v nejlepším případě je to 2 x (doba předání mezi servery + interval kontroly schránky klientem) + čtení zprávy a psaní odpovědi (desítky minut),
 - v horším případě si osoba vybírá poštu 1x denně či nepravidelně (až několik dní či týdnů do odpovědi).
- U důležitých zpráv je vhodné využívat doručenkou.
- U zpráv, které se týkají celého projektu, je vhodné zasílat kopii všem členům týmu a při odpovídání využívat funkci „Odpovědět všem“.
 - U velkých projektů mailing listy – nástroje jako např. Mailman
<http://www.gnu.org/software/mailman/>
- Vždy zachováváme historii (citaci toho, na co reagujeme)!
- Reakci vždy píšeme nad citovaný text nebo mezi jeho částí!

- různé sítě a různí klienti
 - možné problémy s autorizací (nelze spolehlivě rozpoznat, kdo je online),
 - přenosy souborů často nefungují.
- vhodné pro komunikaci 2 osob, které spolupracují na určitém úkolu, nebo pro soukromé dotazy.
- důležitá rozhodnutí a poznatky je třeba vypisovat a rozesílat e-mailem či zpřístupňovat ostatním přes jiný komunikační kanál.
 - Na čem jsme se to vlastně dohodli?
Historii mám v domácím PC a ne v NB.
 - Jaké slovo a v jakém tvaru mám v těch stovkách zpráv hledat, abych to našel?

- telefonní hovory, videohovory, audiokonference či videokonference.
- lze využít jako náhradu osobní schůzky.
- oproti telefonu menší dostupnost:
 - Některý člen týmu nemusí mít potřebné vybavení, nebo jej nemusí mít mobilní.
 - Omezená doba, po kterou jsou členové týmu online.
- Není-li k dispozici náhlavní souprava, nutno tlumit mikrofon a dávat delší časy na odezvu.
 - Ve větší skupině vždy – redukce šumu a datového toku.
- nutnost děláním poznámek o důležitých rozhodnutích (obvykle není k dispozici záznam a pokud je, jeho procházení je zdlouhavé a nepříjemné).

- vhodná pro řešení problémů, které se týkají celého projektu či týmu.
- nutné zachovat přehlednost (vlákna, moderátor, ...)
 - řešíte-li více problémů současně, bez vláken se ztrácí souvislosti.
- nevhodná k hádkám.
- pozor na veřejná fóra
 - když je konkurence objeví, využije dostupné informace (kdo potom od koho opisoval?).

- Existuje celá řada dalších webových služeb a nástrojů pro komunikaci.
- Některé nástroje jsou i velmi sofistikované a specializované na týmovou spolupráci.
- Nevýhodou je, že na využití takového nástroje se musejí dohodnout všichni členové týmu a musejí se s ním seznámit.

- Kam zařadit Facebook?
 - Kombinace více nástrojů.
 - Především nestrukturované fórum a chat.
 - Přehlednost si musí zajistit uživatelé – není-li využití disciplinované, přehlednost se rychle ztrácí.
 - Pozor na licenci k obsahu!

- Základem je klasický nestrukturovaný chat s více místnostmi, možností privátního chatu a videohovorů.
- Poměrně nově možnost využití vláken a místností s diskusním fórem.
- Platforma pro hráče – pro práci na projektu podobné vlastnosti jako MS Teams apod., i když se stále zlepšuje.
- Pro pokročilejší využití nutno doplňovat funkcionalitu pomocí robotů.
- Zrádné nastavení oprávnění.
 - 2 mechanismy nastavování oprávnění – práva a role.
 - Administrátor s nízkou rolí nemůže nastavovat práva ostatním, protože mají vyšší roli.
 - Uživatel s vysokou rolí nemusí mít administrátorská oprávnění – pak je jako běžný uživatel, ale je problém mu cokoliv nastavit.
 - Méně přehledné – rizika eskalace oprávnění a dalších nežádoucích stavů.
- Problematické vyhledávání v historii.

- Když se v zemi začal masivněji šířit Internet, většina připojení byla vytáčených přes pevné linky.
 - Čtení zpráv většinou dávkově se zpožděním.
 - Webové služby byly jednoduché, specializovaných aplikací pro komunikaci nebylo mnoho.
 - Snaha dobře strukturovat komunikaci (je třeba se vyznat ve všem novém) – vytváření dobře strukturovaných fór s tématy, vlákny, ...
- Dnes jsme téměř pořád online.
 - Očekává se, že příjemce zprávu dostane okamžitě a zprávy čte pořád (zvládá sledovat kontext).
 - Množství informací je velké – snažíme se optimalizovat jejich zpracování (čtení 1. věty z každého příspěvku, přeskakování nezajímavého, ...).
 - Strukturování komunikace přináší režie – snaha se jím nezabývat.
- Při práci na SW projektu potřebujeme i informace z historie, protože ne vše si zapamatujeme a ne vše, co se dohodne, uděláme ihned.
 - Ale jsme zvyklí na nestrukturovanou komunikaci, kde se k historii špatně vrací.

Projektová wiki a web projektu

- vhodná pro shromáždění veškerých informací o projektu.
- hlavní stránka by měla obsahovat základní informace o projektu:
 - název,
 - řešitelé,
 - datum zahájení a důležité termíny,
 - základní informace o zadání projektu,
 - základní informace o průběhu projektu (cíle, výsledky, dostupné verze, ...),
 - ...
- na dalších stránkách by měly být popisy dílčích problémů a jejich řešení.
- Není příliš vhodná jako diskusní fórum.



navigation

- Main page
- Community portal
- Current events
- Recent changes
- Random page
- Help

search

Go Search

toolbox

- What links here
- Related changes
- Upload file
- Special pages

[page](#)
[discussion](#)
[edit](#)
[history](#)
[move](#)
[watch](#)

[Idytrych](#)
[my talk](#)
[my preferences](#)
[my watchlist](#)
[my contributions](#)

Gadgets v extjs

Bakalářská práce 2009/2010, ukončeno (květen 2010)

Řešitel: Petr Dvořák, xdvora64@stud.fit.vutbr.cz, <http://www.stud.fit.vutbr.cz/~xdvora64/>

Vedoucí: Ing. Jaroslav Dytrych, <http://www.fit.vutbr.cz/~idytrych/>

Contents [\[hide\]](#)

- 1 Cil práce
- 2 Google Gadgets
- 3 Implementace
- 4 Dokumentace správce rozložení stránky projektu KiWi
 - 4.1 Seznam souborů se stručnými popisy
 - 4.2 Podporované prohlížeče
 - 4.3 Observer
 - 4.3.1 update(updatedBy, variableName, newValue)
 - 4.3.2 getStateVariable(variableName)
 - 4.3.3 getState()
 - 4.4 Layout manager
 - 4.5 Metody v layout manageru
 - 4.5.1 loadPage - metoda pro načtení stránky
 - 4.5.2 getComponent(componentId)
 - 4.5.3 getComponentById(componentId)

- vyplatí se u větších projektů řešených více lidmi či u projektů, kde je třeba informovat veřejnost.
- Lze vytvořit i celý portál, který agreguje veškeré webové služby využívané při práci na projektu:
 - seznam kontaktů na členy týmu,
 - diskusní fórum,
 - chat,
 - úložiště dokumentů,
 - pracovní výkazy,
 - ...
- Je třeba zvážit, které informace dát do veřejné a které do neveřejné části.

- Web projektu nemusíme psát manuálně pro každý projekt.
- Existují aplikace, které tvoří web projektu s integrovanou wiki, systémem pro správu verzí, plánováním apod.
- Příklady:
 - Redmine <http://www.redmine.org/>
 - Trac <http://trac.edgewall.org/>

Stahování, zpracování a indexování rozsáhlých textových korpusů

Hledat:

Stahování, zpracování a indexování rozsáhlých textových korpusů

Přehled Aktivita **Úkoly** Nový úkol Ganttův diagram Kalendář Novinky Dokumenty Wiki Soubory Repozitář Nastavení

Úkoly

▼ Filtry

☒ Stav

otevřený

Přidat filtr

► Nastavení

✓ Použít 🗑 Smazat 💾 Uložit

✓ # ▼	Fronta	Stav	Priorita	Předmět	Přiřazeno	% Hotovo	Aktualizováno
☐ 475	Požadavek	Nový	Normální	Napojenie deduplikacnej kniznice na vertikalizator	David Endrych		17.02.2017 20:37
☐ 474	Požadavek	Nový	Normální	Přebalení warc do xz			14.10.2016 18:32
☐ 473	Požadavek	Ve vývoji	Normální	Logování deduplikující vertikalizace	David Endrych		17.02.2017 20:38
☐ 472	Požadavek	Ve vývoji	Nízká	Změny rozložení hashů	Eduard Cuba		19.01.2017 22:18
☐ 471	Požadavek	Nový	Normální	Přebalení warc podle vert	David Prexta		03.11.2016 14:56
☐ 469	Požadavek	Nový	Nízká	Zotavení z pádu deduplikujícího vertikalizátoru	David Endrych		17.02.2017 20:38
☐ 450	Chyba	Nový	Nízká	Vertikalizácia: [FAILURE] Exited with error code 125			01.08.2016 09:32
☐ 444	Podpora	Vyřešený	Vysoká	Doplnit detailnější dokumentaci	Eduard Cuba		09.08.2016 15:35

Úkoly

[Všechny úkoly](#)
[Přehled](#)
[Kalendář](#)
[Ganttův diagram](#)

Pravidla pro komunikaci

- V týmu je třeba stanovit určitá pravidla pro komunikaci:
 - k čemu se budou využívat které komunikační kanály,
 - minimální frekvence kontroly pošty,
 - minimální frekvence prohlížení diskusního fóra,
 - minimální rozsah dostupnosti online,
 - ...
- Bez stanovení pravidel nelze říci, jak dlouho čekat na něčí reakci, než prohlásíme, že nespolupracuje.

- Jako vedoucí byl zvolen pan A.
- Byly dohodnuty tyto komunikační kanály a pravidla:
 - Hlavním komunikačním kanálem bude fórum na webu projektu. Každý jej navštíví nejméně 1x za každý pracovní den, přičemž se stejnou frekvencí se bude vybírat i pošta.
 - Pro elektronické diskuse mezi více členy současně bude využíván chat na webu projektu, termíny diskusí budou domlouvány na fóru.
 - Pro komunikaci 2 osob pracujících na podúkolech bude využit Jabber. Každý člen týmu, který zrovna pracuje na nějakém úkolu, by měl být při práci online.
 - Telefon se bude využívat pouze výjimečně.
 - Pokud někdo nebude moci pravidla dodržet, omluvu a zdůvodnění uvede na fóru.

Prostředky pro sdílení dat v týmu

- FTP
- web
- sdílený disk
- systémy pro správu verzí
- ...

- nejjednodušší a nejdostupnější prostředek.
- nevhodné pro dokumenty či zdrojové texty, které se mění
 - ztráta předchozích verzí,
 - konflikty (přepis novější verze upravenou starší, kterou měl někdo u sebe),
 - souběžný přístup pro zápis?
- vhodné pro sdílení dat, která vzniknou a dále se nemění
 - možné stahování přes HTTP (snadné sdílení odkazu),
 - efektivní i pro velké objemy dat.
- pozor na oprávnění!
 - je-li FTP na hostingu zdarma, ve výchozím nastavení bude vše přístupné přes web,
 - neznalost odkazu nezabrání nalezení konkurencí.
- pozor na bezpečnost!
 - preferujte zabezpečenou variantu (SFTP).

- Je dostupná řada služeb pro sdílení souborů a informací.
- Obvykle obdobné vlastnosti jako FTP, ale často omezení velikosti souboru
 - může být výhodné pouze pro zveřejňované soubory.
- Datové úložiště lze vytvořit i ve webu projektu.
- Vždy pozor na bezpečnost dat.



NONA PROJECT

Projekt Aktuality Organizace Schůzky Chat Nastavení Diskuse Ankety Práce Soubory Plán Odhlásit

Přenos souborů přes protokol http (max. 2MB)

Přidávaný soubor:

 Procházet...

[Přidat](#)

Název	Velikost [B]	Přidal	Přidáno	
sesnaz_dokumentu.pdf	132831		30.04.2009 v 00:34:07	Smazat
prace_export.ods	50702		30.04.2009 v 00:32:12	Smazat
vyhodnoceni_casu.pdf	252322		30.04.2009 v 00:31:10	Smazat
r488-release.tar.gz	112842		29.04.2009 v 01:19:41	Smazat
uzavreni_projektu.pdf	239638		29.04.2009 v 00:41:27	Smazat
zaznamenani_zkusenosti.pdf	204602		29.04.2009 v 00:37:38	Smazat
schuzka5-28-04-2009.pdf	138339		28.04.2009 v 23:51:15	Smazat
schvaleni_bodoveho_hodnoceni.pdf	148782		28.04.2009 v 23:49:30	Smazat
prezentace.pdf	1473282		28.04.2009 v 21:32:41	Smazat
vyhodnoceni_metrik.pdf	244255		28.04.2009 v 05:12:11	Smazat
vyhodnoceni_dotazniku.pdf	261460		28.04.2009 v 02:41:00	Smazat
dotazniky.zip	105611		27.04.2009 v 14:34:40	Smazat
schuzka5-24-04-2009.pdf	135400		27.04.2009 v 13:19:32	Smazat
predvedeni_verze_produkту.pdf	132085		27.04.2009 v 13:18:38	Smazat
prezentace_navrh.pdf	54028	xdytrv00	27.04.2009 v 06:39:45	Smazat

- Dropbox
 - Google Drive
 - TeamDrive
 - Microsoft OneDrive
 - Cesnet ownCloud
-
- Pozor na omezení a placené služby.
 - Ne vždy naplní očekávání (možnost návratu k předchozímu stavu nemusí stačit – nedetekuje a neřeší konflikt).

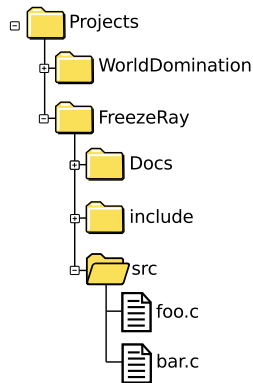
Systemy pro správu verzí

- Anglické označení:
 - RCS – Revision Control System (i název konkrétního nástroje)
 - VCS – Version Control System (systém pro správu verzí)
- určeny pro správu a sdílení zdrojových textů programu a dokumentů, které se v průběhu vývoje projektu mění nebo jsou vytvářeny více lidmi.
- umožňují dohledání starších verzí souborů.
- umožňují řešit konflikty (přepsání novější verze upravenou starší).
- sledují historii změn, takže se lze kdykoliv vrátit do libovolného předchozího bodu, kde to ještě fungovalo.
- umožňují mít více větví vývoje (např. stabilní větev a experimentální větev, nebo speciální verzi pro určitého zákazníka), přepínat mezi nimi a spojovat je.
- nevhodné pro velké soubory s testovacími či jinými daty, která není třeba verzovat (nutí ostatní členy týmu si je stáhnout).

- Zastaralé:
 - 1972 SCCS – pouze 1 vývojář v 1 adresáři
 - 1980 RCS – výkonnější, lepší UI
 - 1986 CVS – více vývojářů, centrální server
 - 1999 Subversion (SVN) – atomické commity
- Starší, méně využívané (složitější použití, horší škálovatelnost), distribuované:
 - 2001 GNU arch, Monotone
 - 2002 Darcs
- Komerční:
 - 1995 Perforce Helix
- Aktuální, distribuované (a podporované v IDE):
 - 2005 Git – vytvořil Linus Torvalds pro jádro Linuxu
 - 2005 Mercurial (hg) – Matt Mackall, alternativa Gitu
 - 2005 GNU Bazaar (bzd) – Canonical, evoluce GNU arch, více možností, ale pomalejší

- **Pracovní strom** (working tree)
 - adresářová struktura se zdrojovými soubory (máme i bez VCS)
- **Vyčkávací prostor**
 - prostor pro to, co se bude commitovat (ukládat do repozitáře), tedy přidané, odstraněné, přejmenované a změněné soubory
 - nemusí obsahovat všechny změny v pracovním stromě
- **Commit**
 - sada změn v repozitáři (jednotka práce vývojáře)
 - definuje určitý stav projektu (verzi)
- **Větev**
 - uživatelem pojmenovaná sada commitů
 - větve umožňují paralelní vývoj (rozdílné linie vývoje)
- **Repozitář**
 - databáze, která obsahuje historii projektu
- **Klon repozitáře**
 - kopie repozitáře s pracovním stromem
- **Holý (bare) repozitář**
 - kopie repozitáře bez pracovního stromu

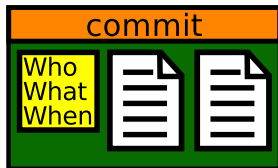
- adresářová struktura
- zdrojové soubory



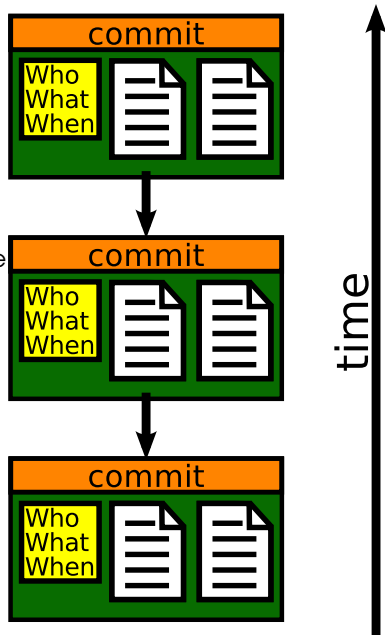
- Obsahy souborů



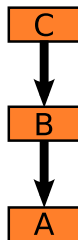
- Obsahy souborů
- „Commits“



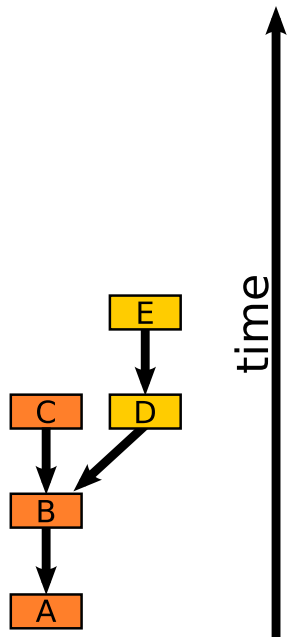
- Obsahy souborů
- „Commits“
- Rodová historie
 - Odkaz na přímého předchůdce



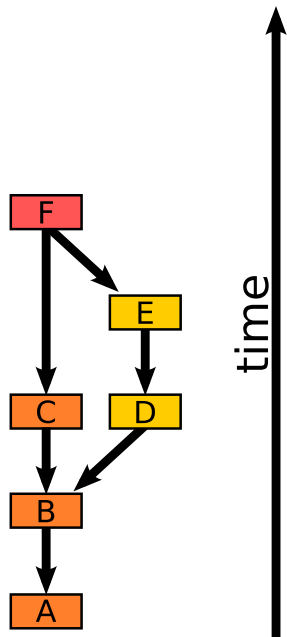
- Obsahy souborů
- „Commits“
- Rodová historie
 - Odkaz na přímého předchůdce



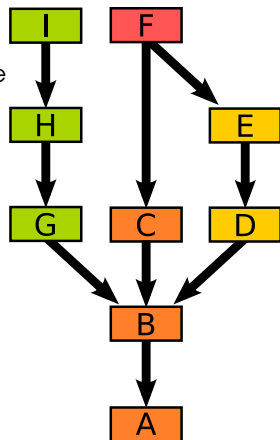
- Obsahy souborů
- „Commits“
- Rodová historie
 - Odkaz na přímého předchůdce
 - větve (branches)



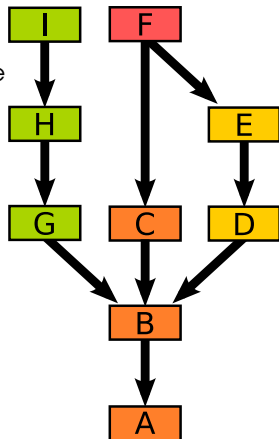
- Obsahy souborů
- „Commits“
- Rodová historie
 - Odkaz na přímého předchůdce
 - větve (branches)
 - spojení (merges)



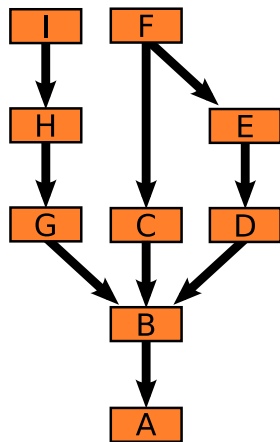
- Obsahy souborů
- „Commits“
- Rodová historie
 - Odkaz na přímého předchůdce
 - větve (branches)
 - spojení (merges)
- Hlava (head / tip)
 - nejmladší commit ve větvi



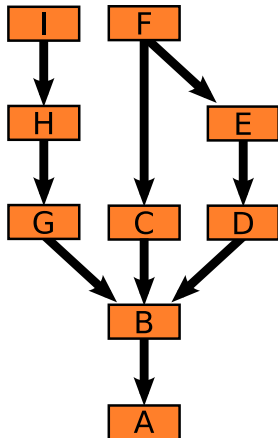
- Obsahy souborů
- „Commits“
- Rodová historie
 - Odkaz na přímého předchůdce
 - větve (branches)
 - spojení (merges)
- Hlava (head / tip)
 - nejmladší commit ve větvi
- Vícehlavový repozitář (multi-head / multi-tip)
 - více větví, které nejsou spojené



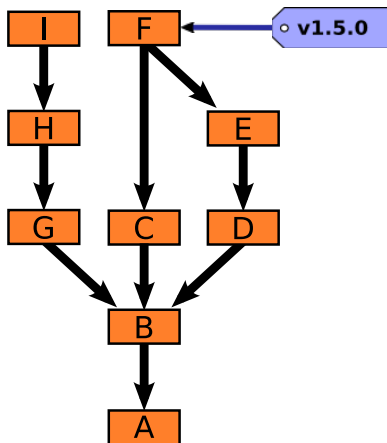
Orientovaný acyklický graf



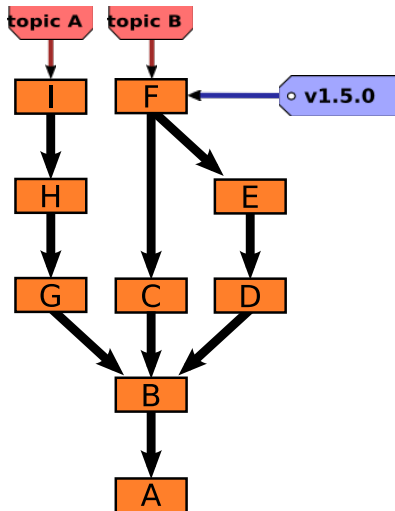
Ukazatele do DAG



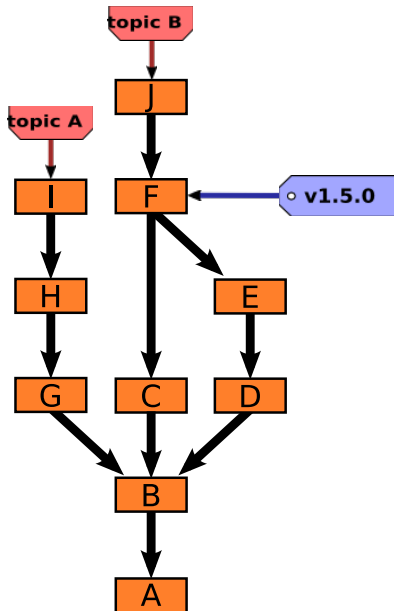
- značky (tags)
 - obvykle se nemění



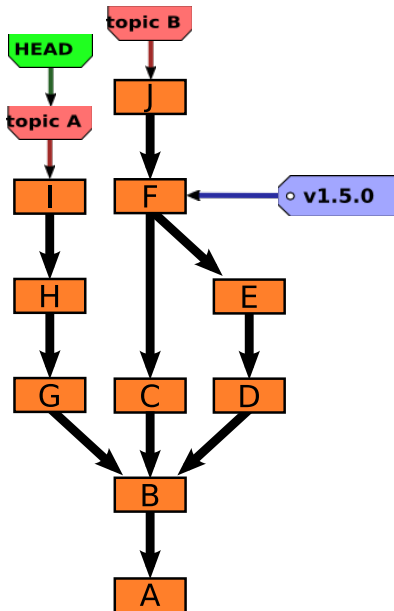
- značky (tags)
 - obvykle se nemění
- větve (branches)



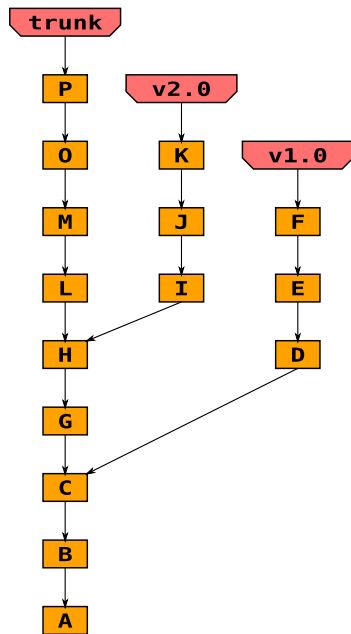
- značky (tags)
 - obvykle se nemění
- větve (branches)



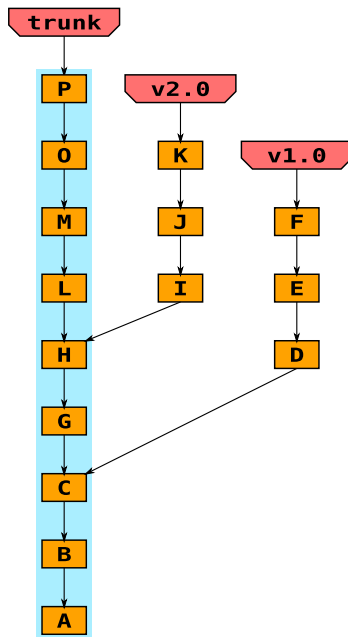
- značky (tags)
 - obvykle se nemění
- větve (branches)
- aktuální checkout
 - Hlava (HEAD)
 - odkaz na větev, na které právě pracujeme



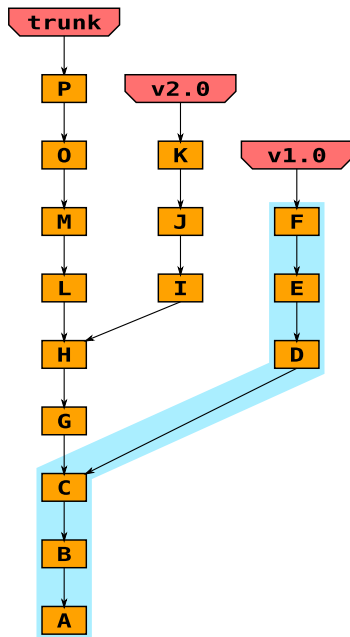
- Workflow (pracovní tok) označuje postup práce s repozitářem. Činnosti se typicky opakují.
- Základní tok je udělat změny v pracovním adresáři, přidat tyto změny do vyčkávacího prostoru (sloveso stageovat) a commitnout. Toto děláme opakovaně.
- Obvyklé:
 - Tradiční workflow
 - Tématické větve



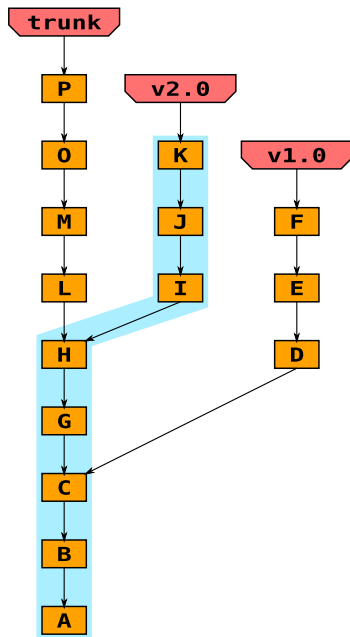
Hlavní vývojová větev



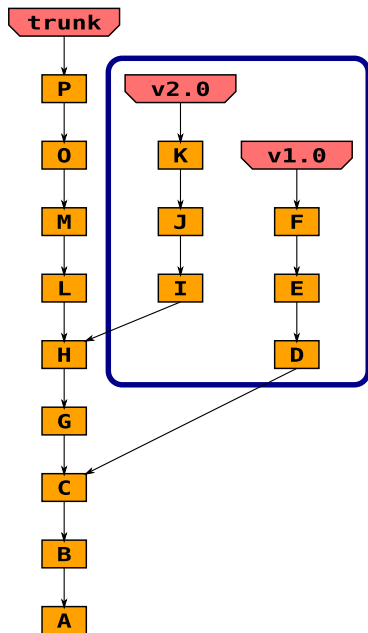
Historické verze



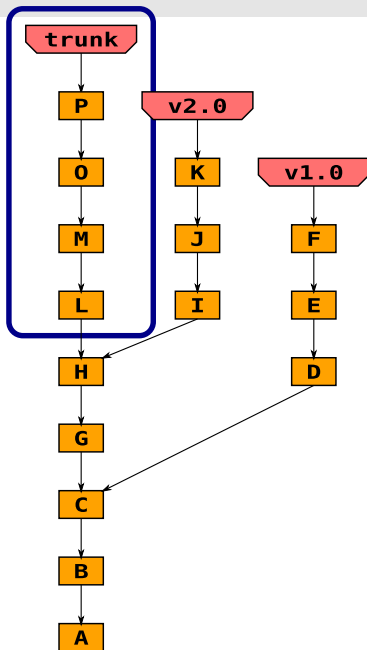
Historické verze



Pouze opravy chyb

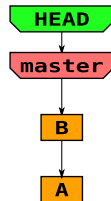


Nové funkce a opravy chyb

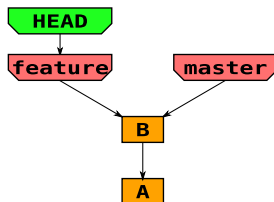


- Jedna větev pro každé téma
- Časté commity
- Krátká životnost
- Dokončené téma spojeno do **master**

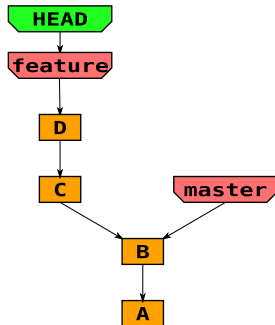
Stávající repozitář



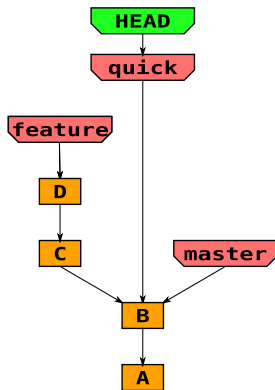
Vytvoříme větev pro **feature**.



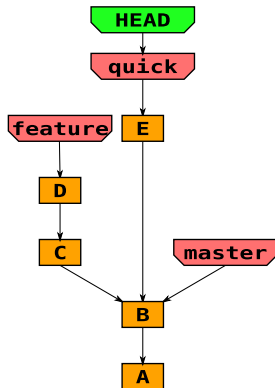
Vyvíjíme **feature**.



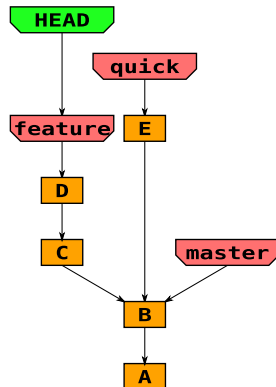
Při tom objevíme chybu,
která nám brání ve vývoji,
tu raději hned opravíme.



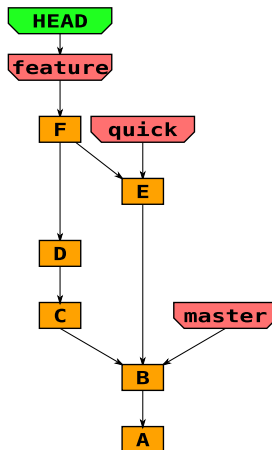
Rychle opravíme chybu.



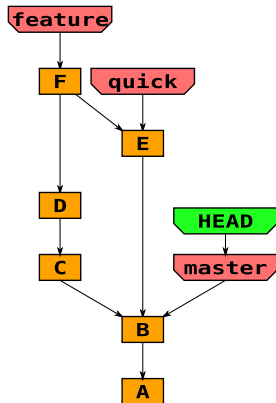
Zpět k vývoji **feature**.



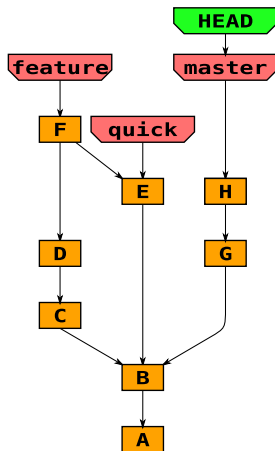
Připojíme (merge) opravu chyby.



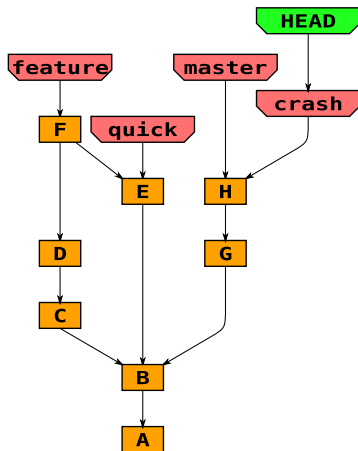
Upstream padá!



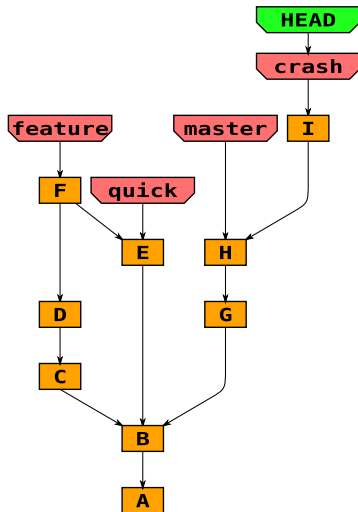
Nejprve aktualizujeme
master verzí z upstreamu.



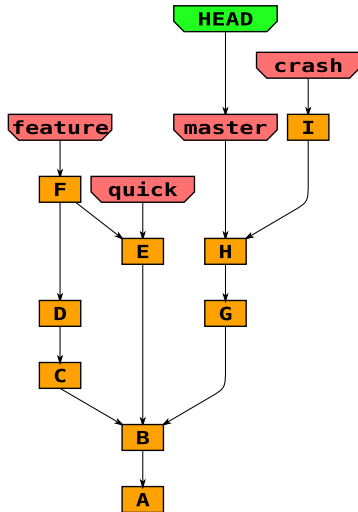
Vytvoříme větev pro opravu této chyby **crash**.



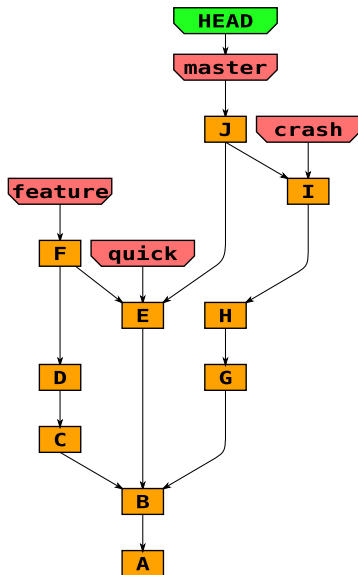
Opravíme tuto chybu **crash**.



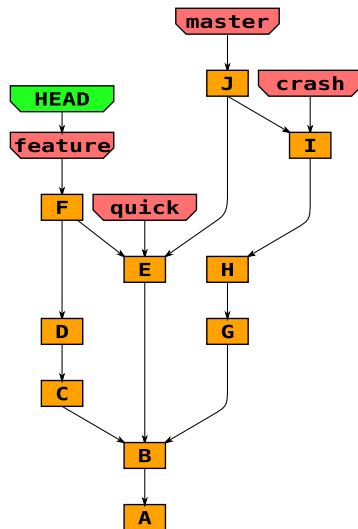
Budeme integrovat.



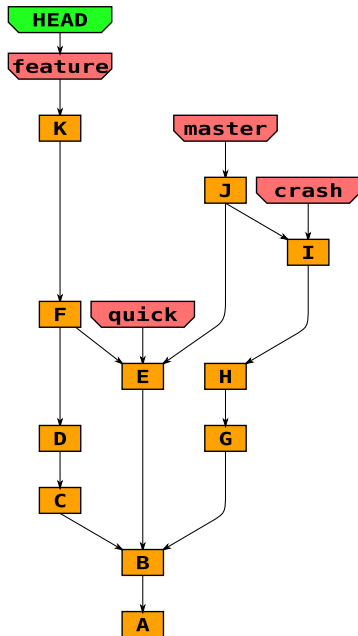
Integrujeme opravy obou
chyb **quick** i **crash** do **master**.



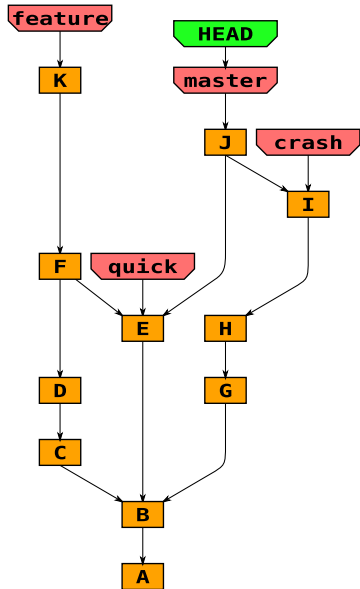
Zpět k vývoji naší **feature**.



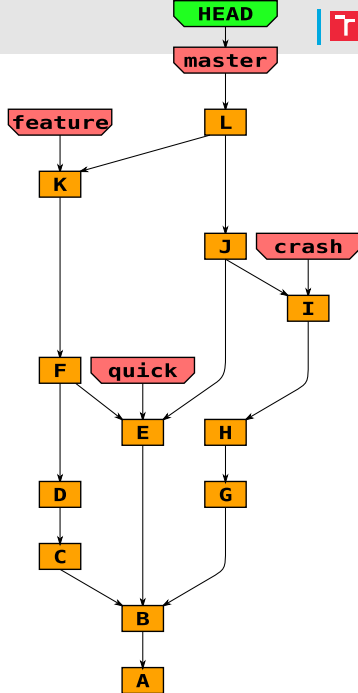
Dokončíme **feature**.



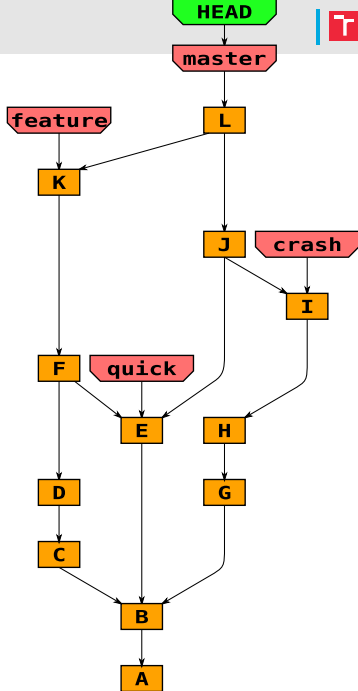
Přepneme do **master**.



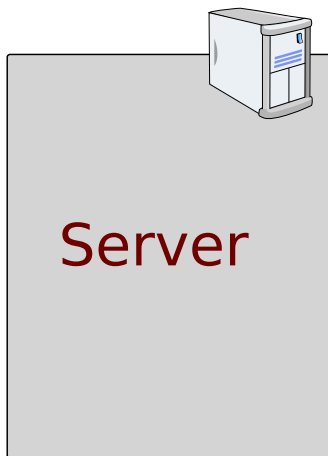
Připojíme **feature** do **master**.

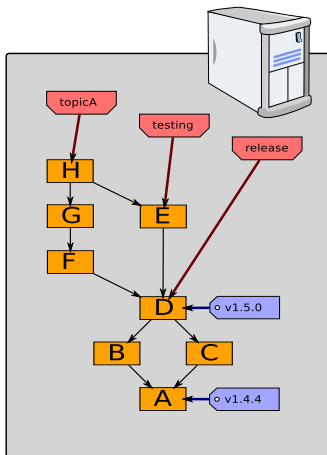


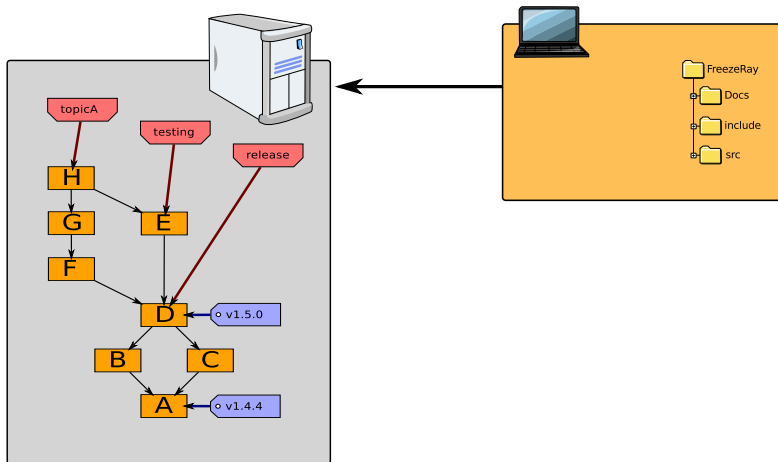
- Přehlednější
- Revize (kontrola před spojením)
- Spojení lze odkládat



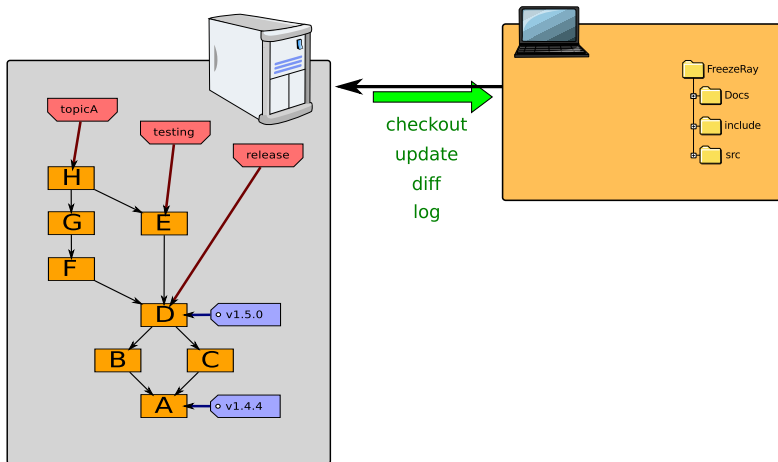
- Centralizované
- Decentralizované (distribuované)



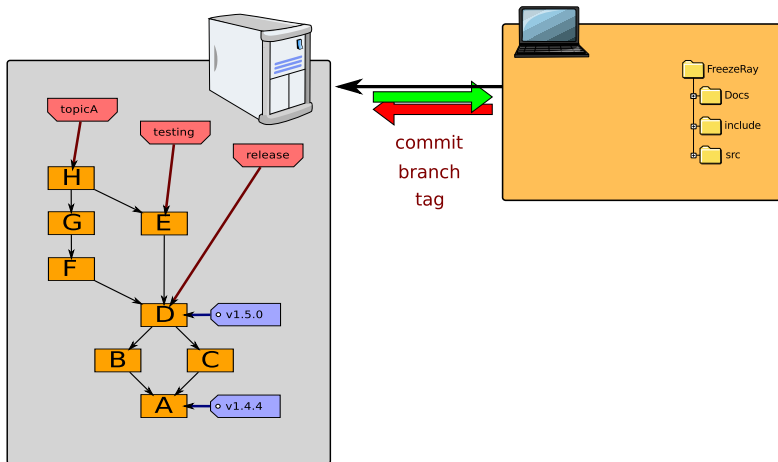




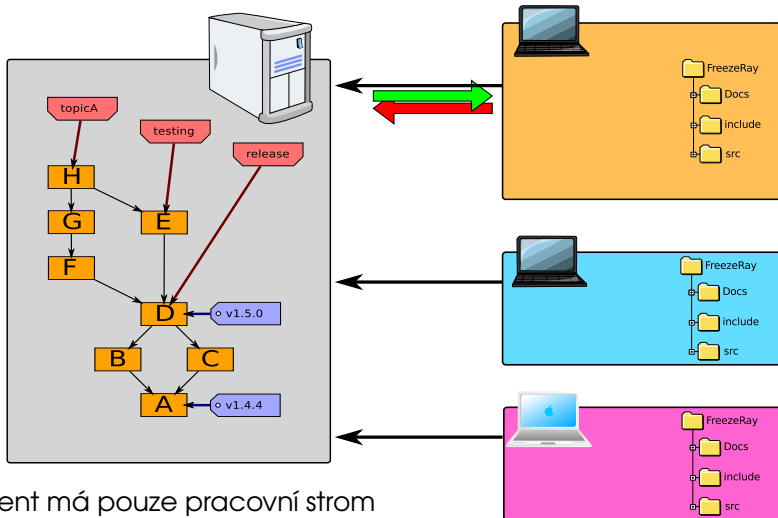
- Klient má pouze pracovní strom



- Klient má pouze pracovní strom
- Všechny operace vyžadují server

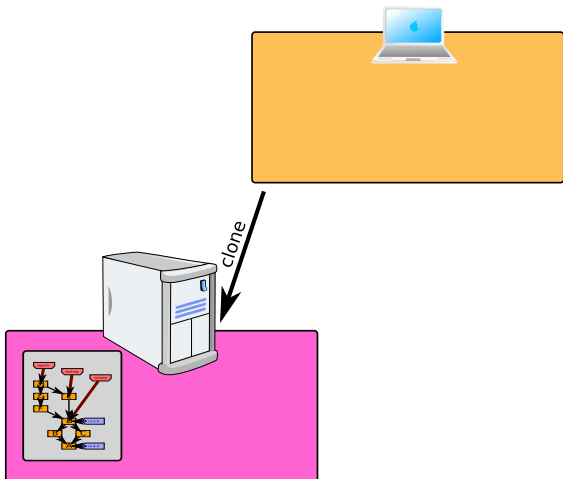


- Klient má pouze pracovní strom
- Všechny operace vyžadují server

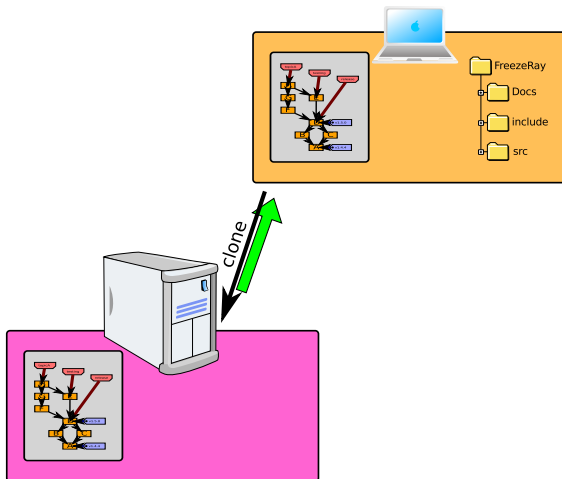


- Klient má pouze pracovní strom
- Všechny operace vyžadují server
 - Jediný bod selhání (Single point of failure)
 - Úzké místo

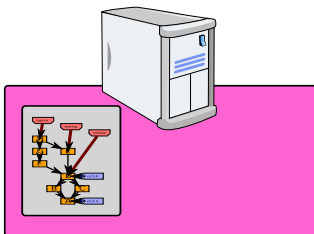
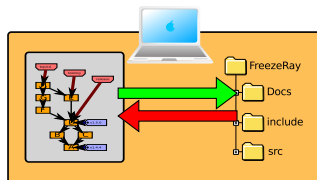
- Klient má klon repozitáře



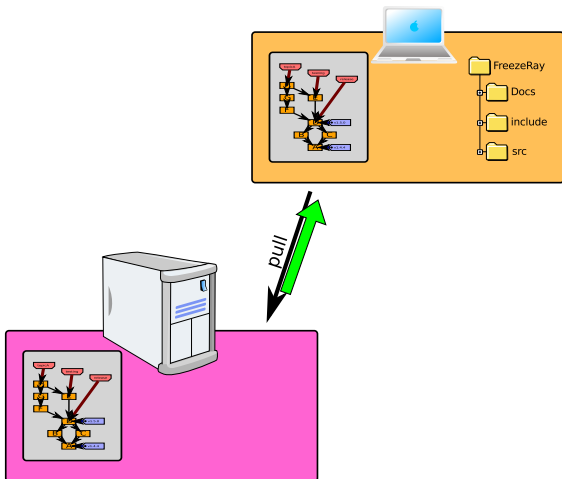
- Klient má klon repozitáře



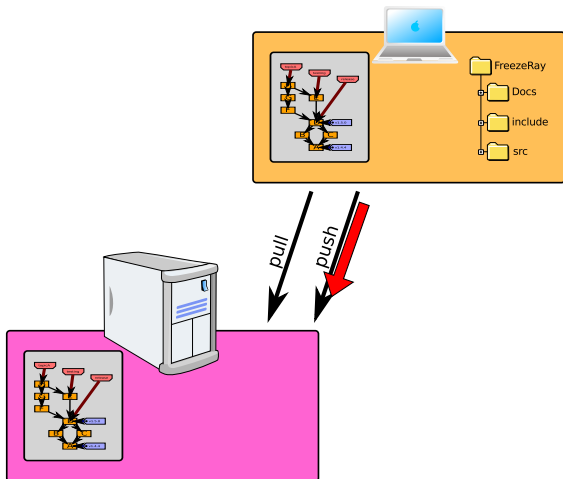
- Klient má klon repozitáře (operace dělá lokálně)



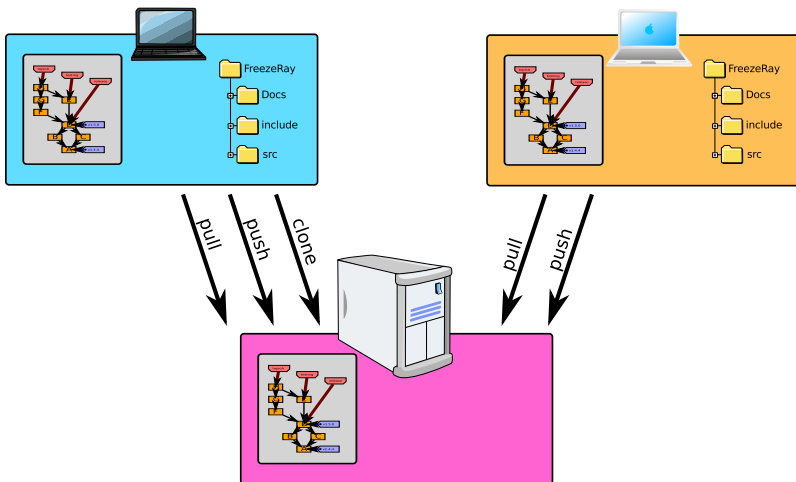
- Klient má klon repozitáře (operace dělá lokálně)
- Operace pull (stáhnout)



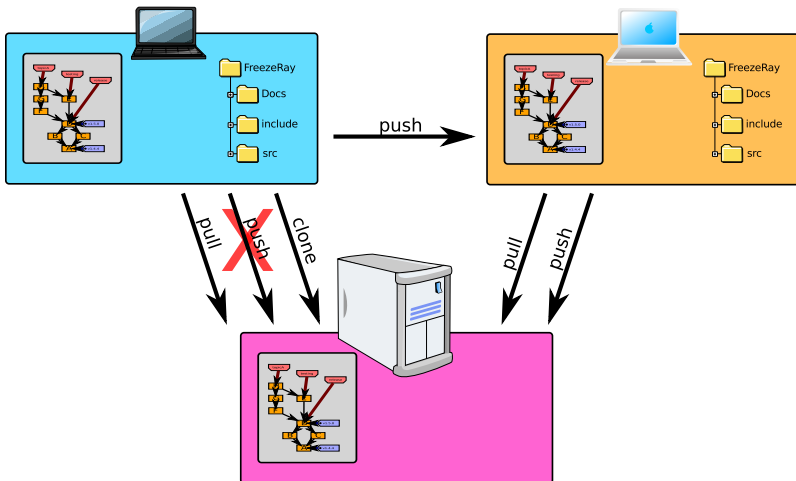
- Klient má klon repozitáře (operace dělá lokálně)
- Operace pull (stáhnout) a push (natlačit na server)



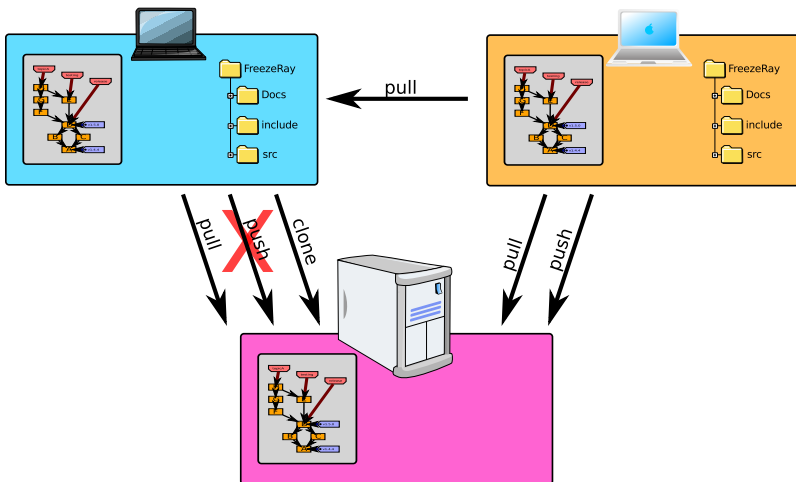
- Klient má klon repozitáře (operace dělá lokálně)
- Operace pull (stáhnout) a push (natlačit na server)



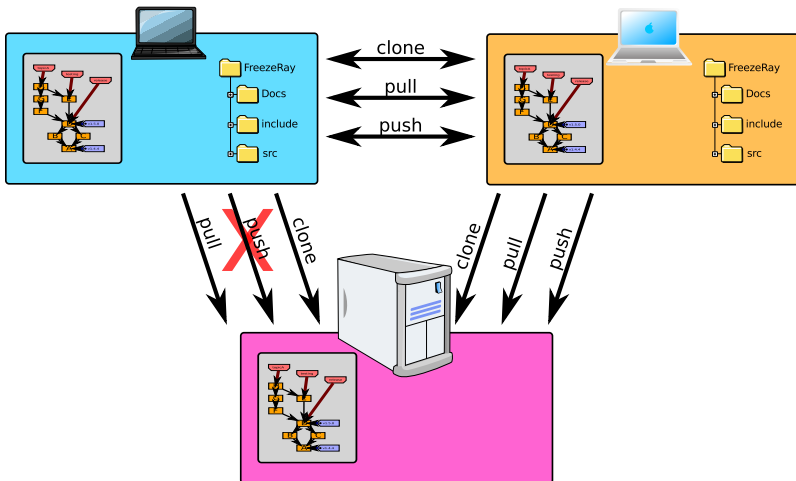
- Klient má klon repozitáře (operace dělá lokálně)
- Operace pull (stáhnout) a push (natlačit na server)
- Kdokoliv může být server (ale nemusí povolit zápis)



- Klient má klon repozitáře (operace dělá lokálně)
- Operace pull (stáhnout) a push (natlačit na server)
- Kdokoliv může být server (ale nemusí povolit zápis)



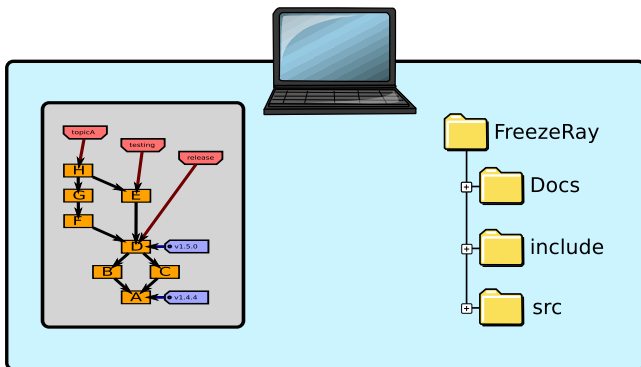
- Klient má klon repozitáře (operace dělá lokálně)
- Operace pull (stáhnout) a push (natlačit na server)
- Kdokoliv může být server (ale nemusí povolit zápis)



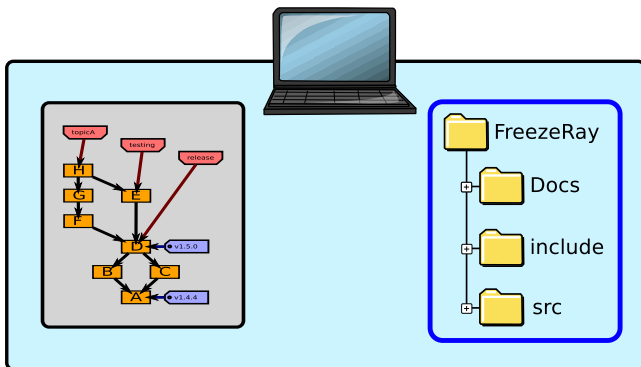
- Rychlost (lokální operace)
- Neinvazivní „mikro“ commity
 - Commitovat často
 - Commitovat malé změny (v naší větvi nemusí být funkční kód)
 - Nerozbijeme sestavení
- Práce offline
- Absence jediného bodu selhání (single point of failure)
- Triviální zálohování
 - každý vývojář je záloha

- Jeden vývojář
- Centralizovaná
- Integrátor
- Diktátor a poručíci

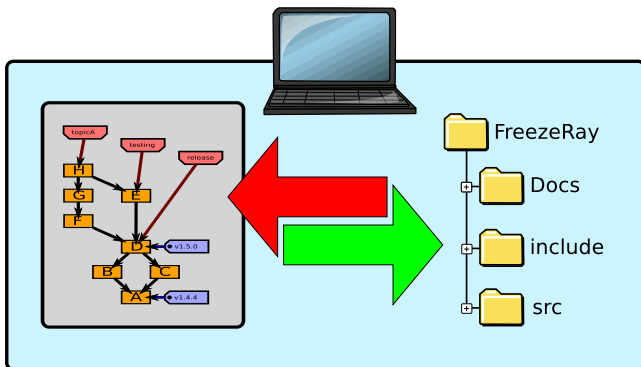
Vývojář lokálně vytváří svůj nový projekt

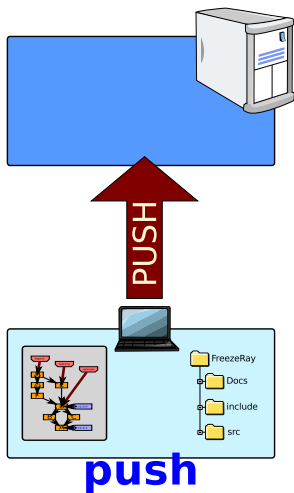


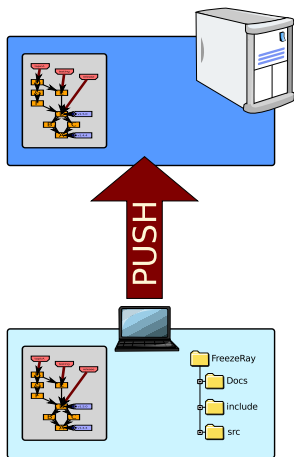
Vývojář lokálně vytváří svůj nový projekt

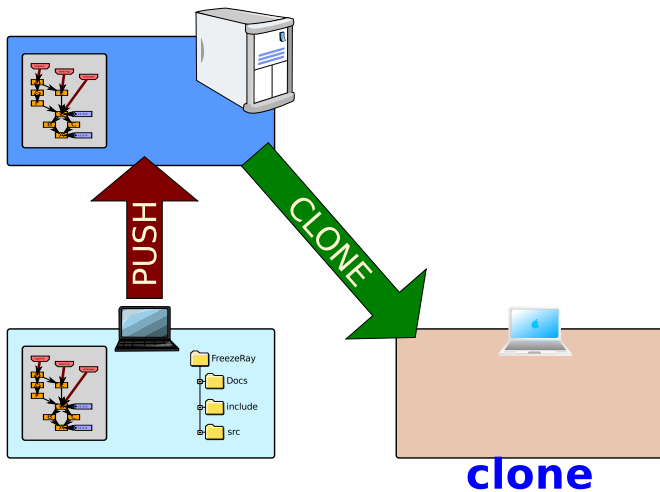


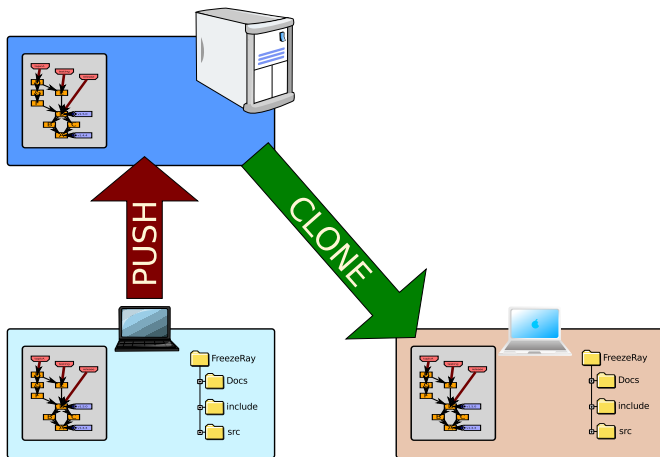
Vývojář lokálně vytváří svůj nový projekt

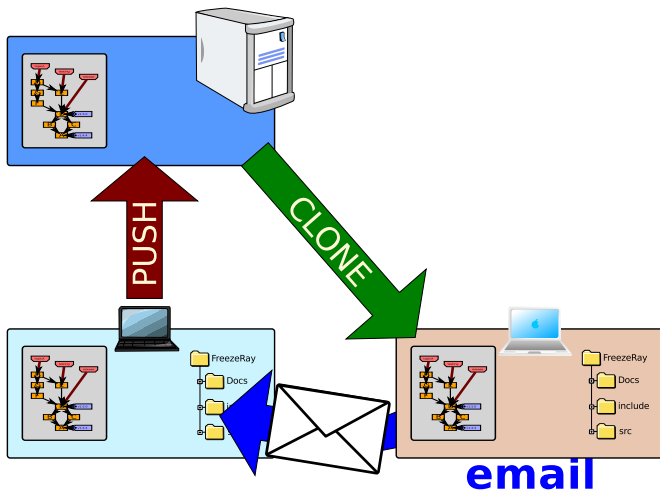




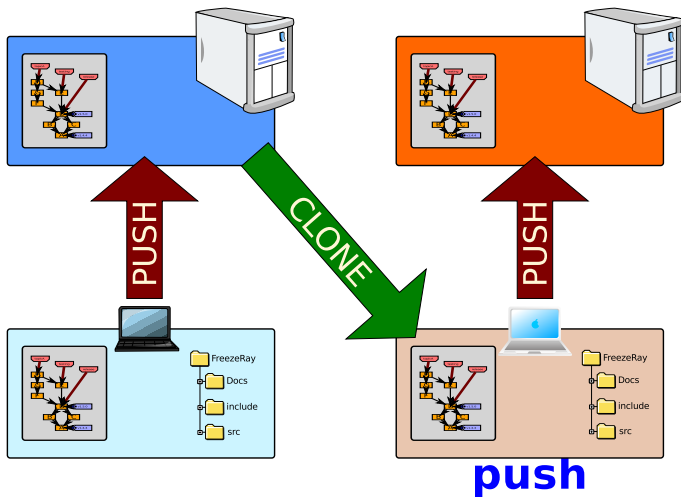




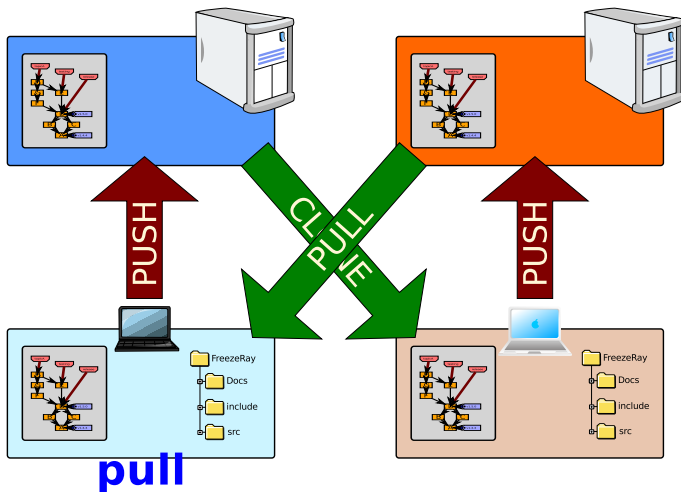




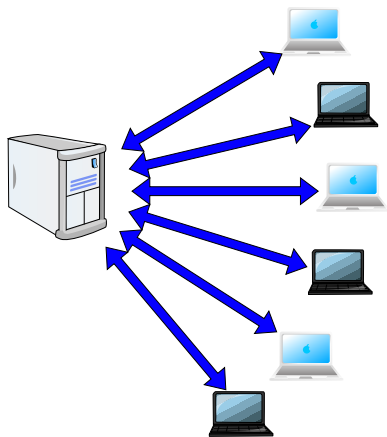
Zaslání patche e-mailem

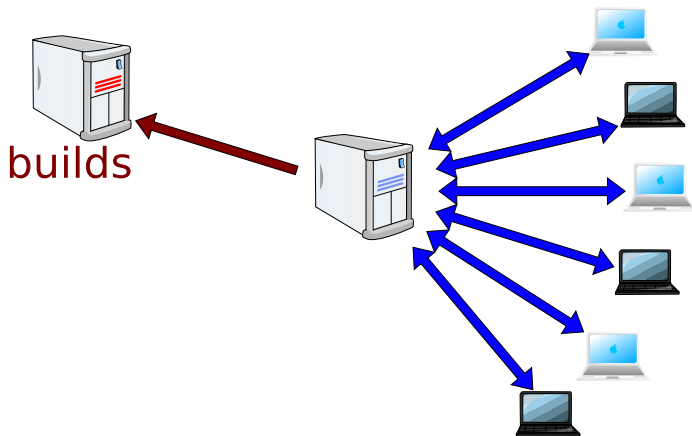


Pull request

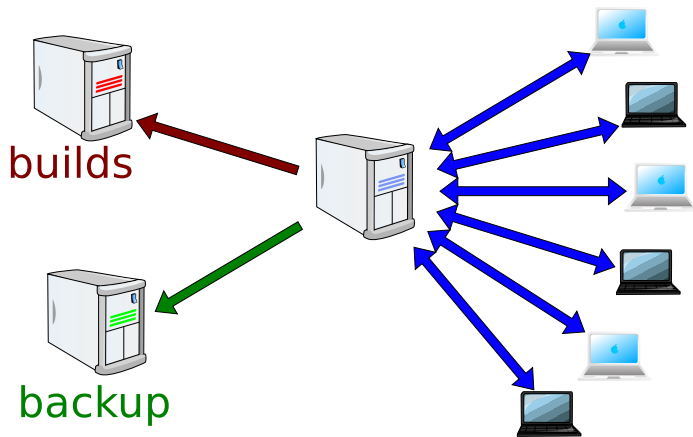


Pull request

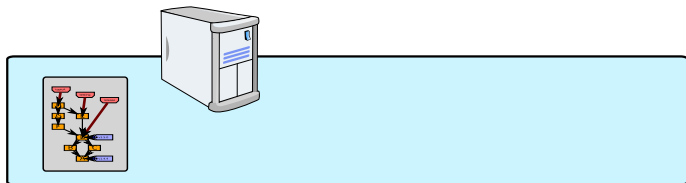


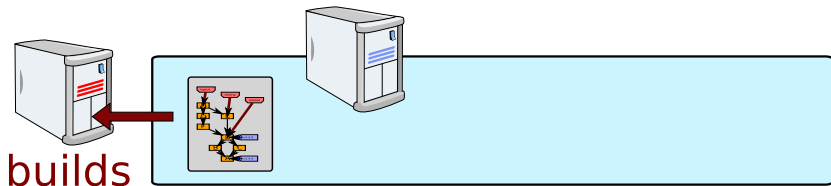


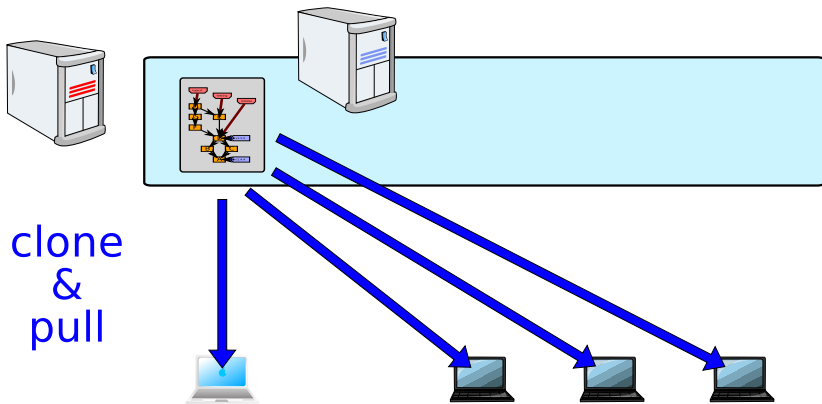
Specializované servery pro sestavení



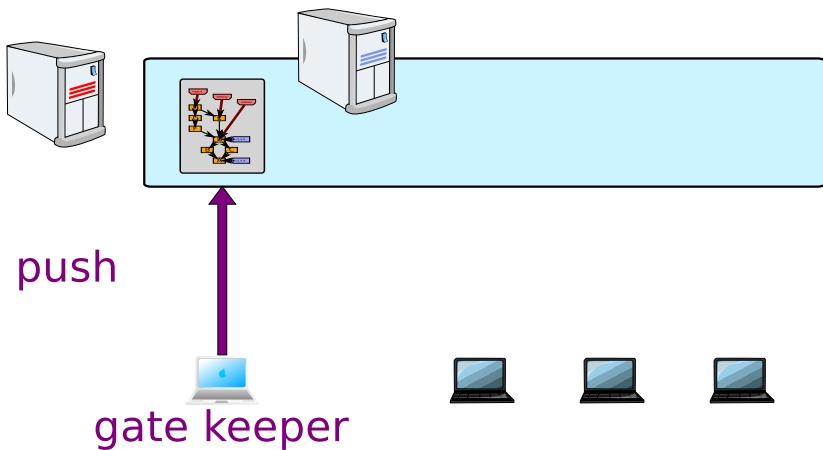
Specializované servery pro sestavení a zálohování



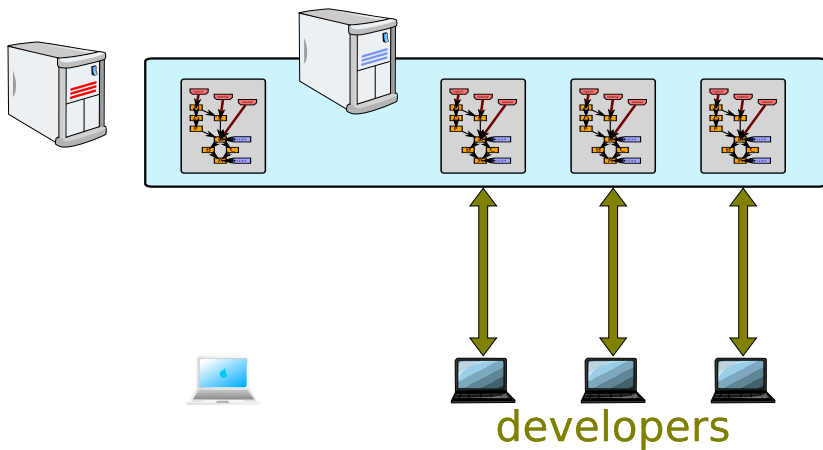




Kdokoliv může číst

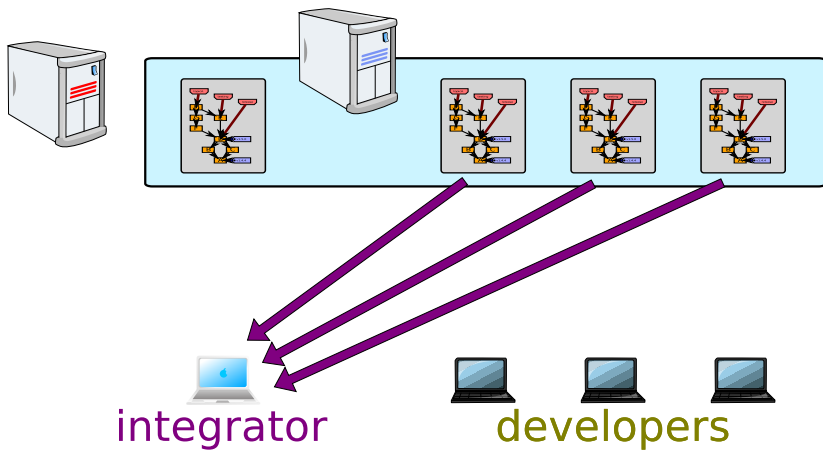


Pouze 1 vývojář může zapisovat

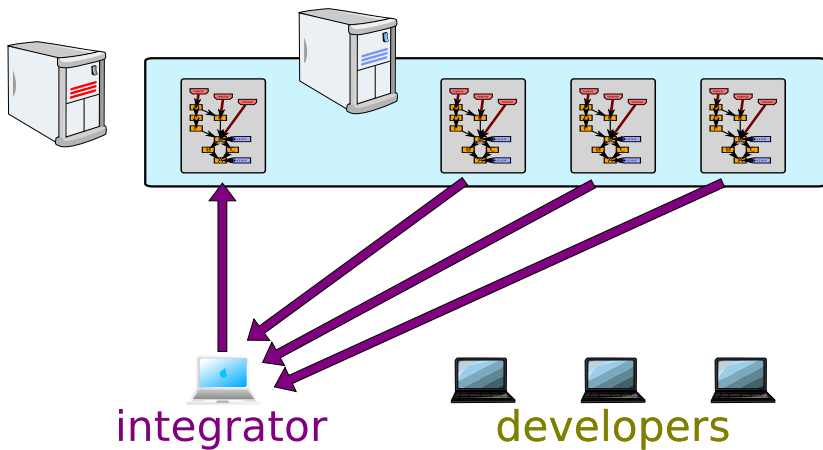


Ostatní vývojáři udržují svoje klony repozitářů

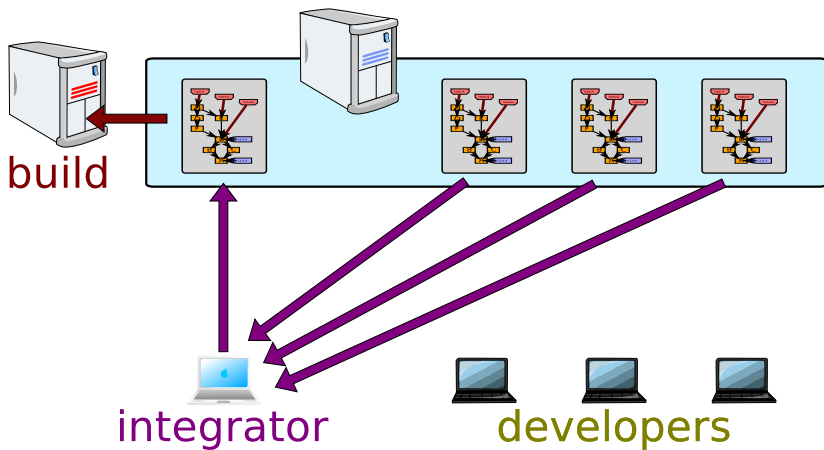




Integrátor stahuje a integruje tématické větve



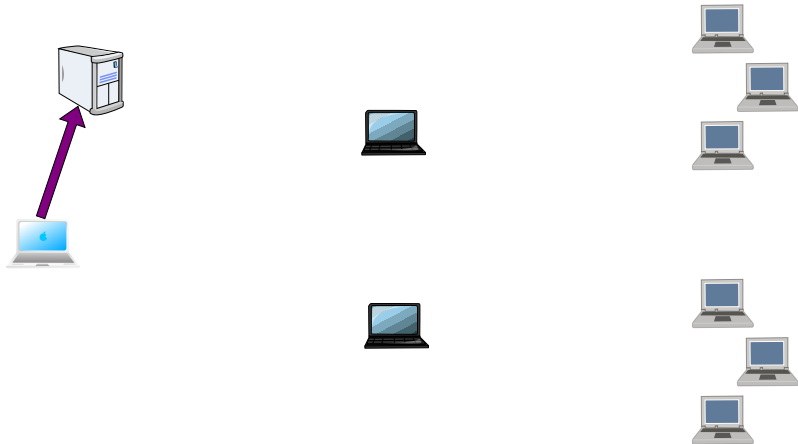
Výsledek integrátor otestuje a natlačí do repozitáře



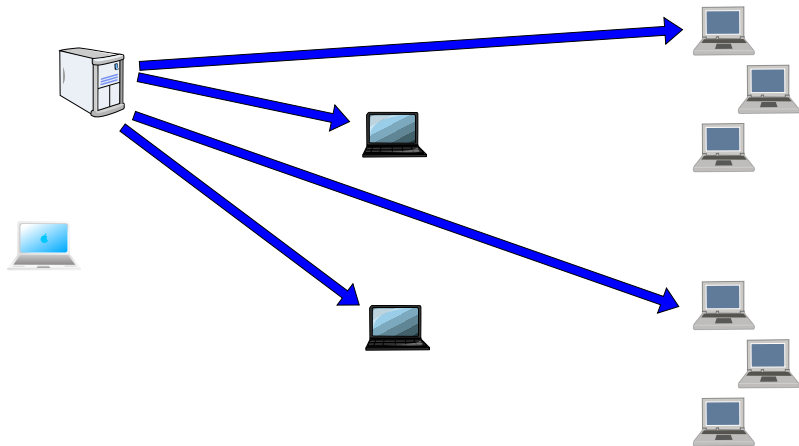
Obsah repozitáře je automaticky sestaven



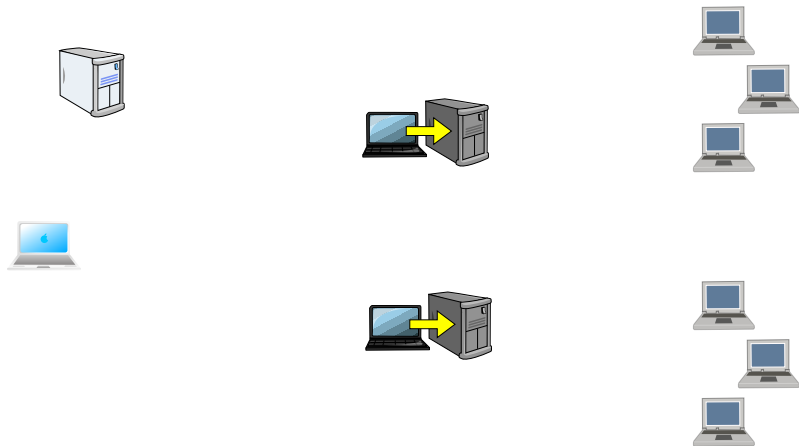
Využívá se např. pro vývoj jádra Linuxu



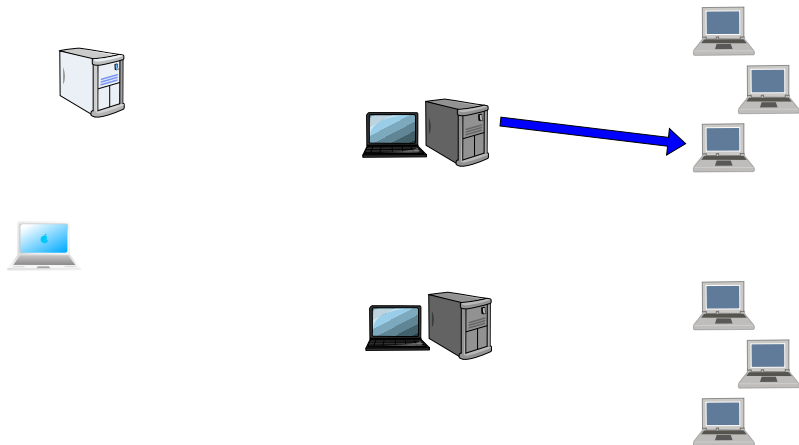
Linus Torvalds vytvořil 1. verzi jádra



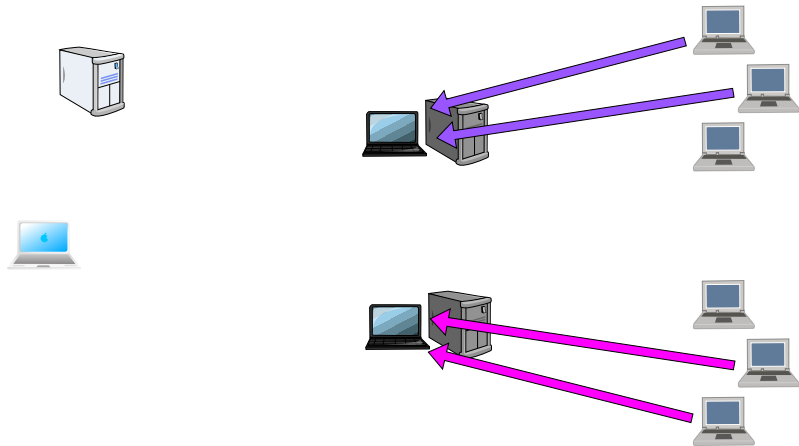
Všichni si naklonují repozitář



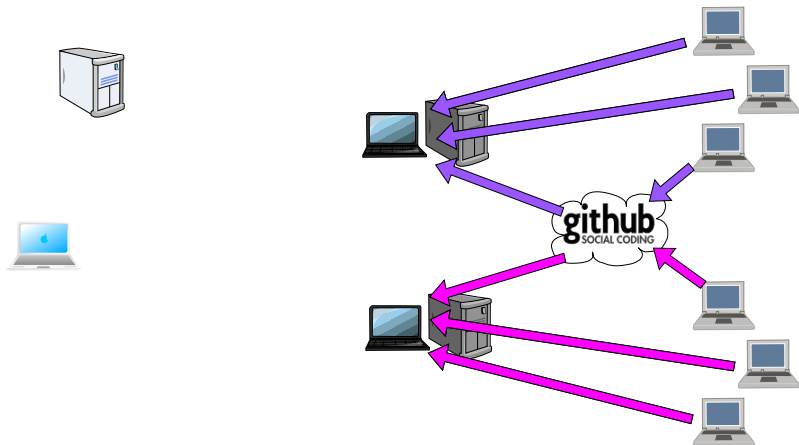
Poručíci (lieutenant) vytvoří repozitáře pro podsystémy



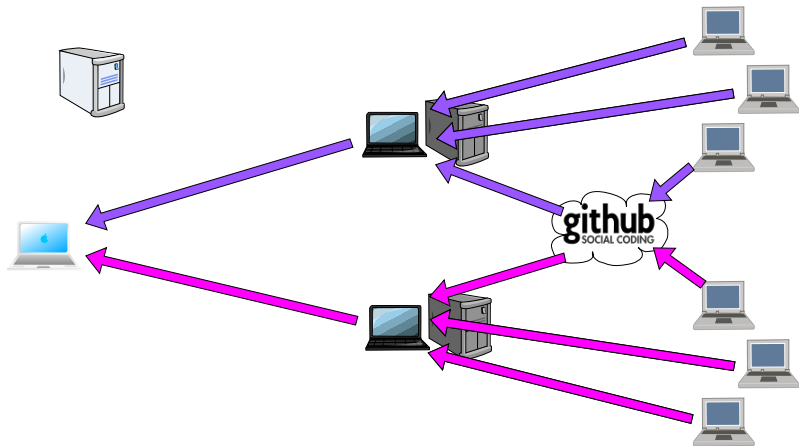
Poručíci (lieutenant) jsou zodpovědní za jednotlivé podsystémy



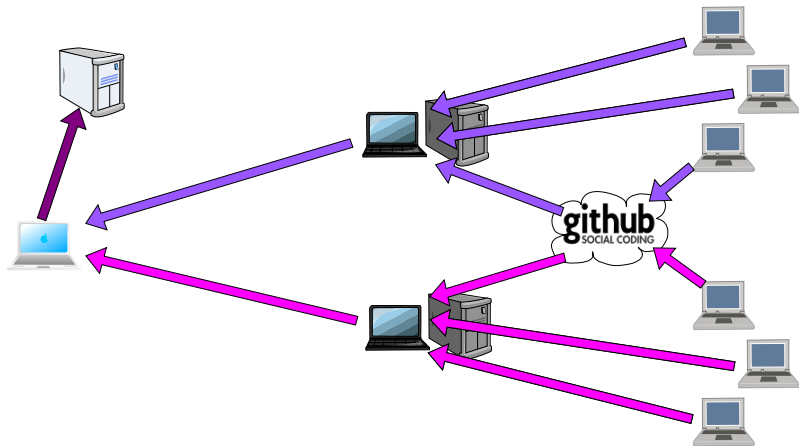
Poručíci (lieutenant) jsou zodpovědní za jednotlivé podsystémy



Poručíci (lieutenant) jsou zodpovědní za jednotlivé podsystémy



Poručíci posílají diktátorovi pull request

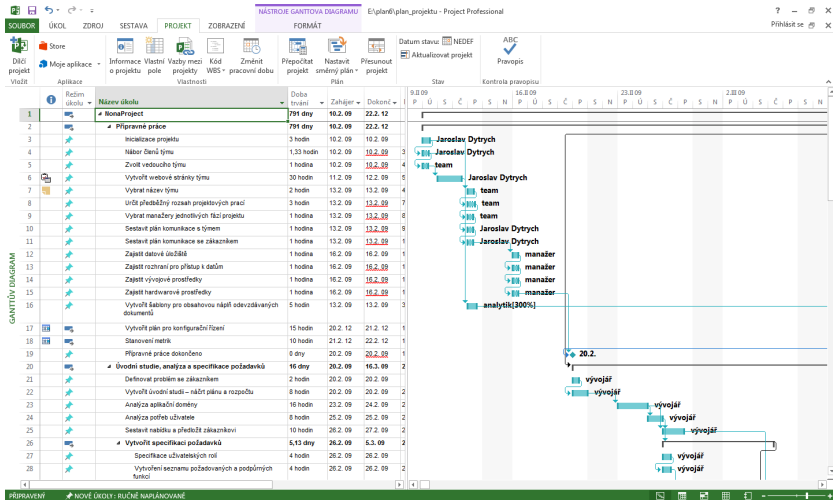


Diktátor spravuje centrální repozitář

Úvod do plánování projektu

- Práce na projektu je třeba předběžně rozvrhnout.
- První fáze projektu se vždy zpozdí a na poslední nebude dostatek času.
- Při plánování je třeba uvažovat rezervy.
- Plán musí být ve všech částech dostatečně podrobný.
- Lze využít i nějaký SW pro podporu plánování, např. MS Project.
- Plán je nutné sledovat a aktualizovat.

GANNTTUV DIAGRAM



plan_projektu - Project Professional

NÁSTROJE KALENDÁŘE

Soubor Úkol Zdroj Sestava Projekt Zobrazení FORMÁT

Ganttův diagram Vložit Vymout Kopírovat Kopírovat formát

Obrazovka Schránka

Písmo

Plán

Označit plnění podle plánu Respektovat propojení Deaktivovat

Naplánovat ručně Naplánovat automaticky Úkoly

Zkontrolovat Přesunout Režim

Úkol Mílek

Informace

Poznámky Podrobnosti

Přidat na časovou osu Vlastnosti

Přijít na akci Přijít na akci

Přihlásit se

Kalendář

leden 2009

únor 2009

březen 2009

11

pondělí úterý středa čtvrtek pátek sobota neděle

9. 10. 11. 12. 13. 14. 15.

Inicializace
Nábor čl.
Zvolit ve

Vytvořit webové stránky týmu, 30 hodin

Vybrat ná.
Uřít před.
Vybrat m.
Sestavit j.
Sestavit j.

16. 17. 18. 19. 20. 21. 22.

Zajistit d.
Zajistit rc.
Zajistit vs.
Zajistit h.

Přípravné práce dokonč.
Definovat.
Vytvořit úvodní studii -

23. 24. 25. 26. 27. 28. 1 III

Analýza aplikační domény, 16 hodin

Analýza potřeb uživatele

Sestavit nabídku a předložit zákazníkovi

Specifikace i.
Vytvoření se

Specifikace funkčnosti aplikace, 24 hodin

Specifikace funkčnosti aplikace, 24 hodin

Specifikace uživ

Analýza

Zpracovat náměty a připomínky od zákazníka, 30 hodin

2. 3. 4. 5. 6. 7. 8.

PRIPRAVENÍ NOVÉ ÚKOLY: RUČNĚ NAPLÁNĚVÁNÉ

NÁSTROJE SEZNAMU ZDROJŮ plan_projektu - Project Professional

ÚKOL ZDROJ SESTAVA PROJEKT ZOBRAZENÍ FORMÁT

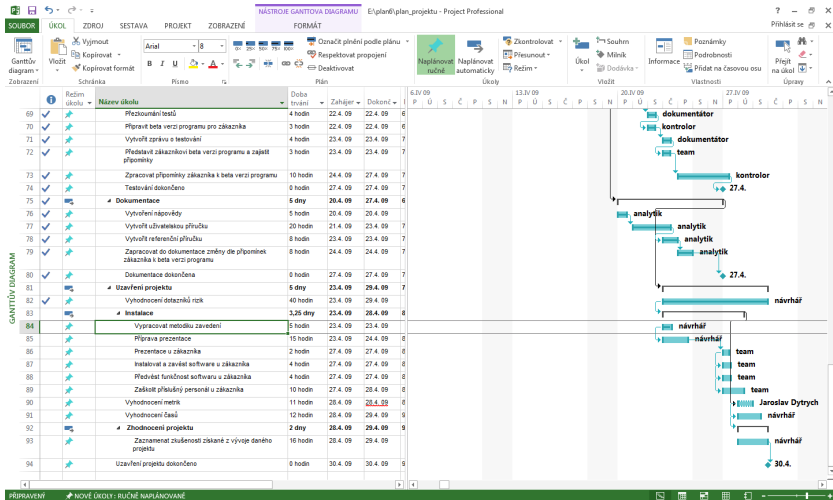
Úkoly

Seznam zdrojů

	Název zdroje	Typ	Popisek materiálu	Iniciály	Skupina	Maximální počet jednotek	Standardní sazba	Přesčasová sazba	Náklady na použití	Nabíhnutí nákladů	Základní kalendář	Kód	Přidat nový sloupec
1	Jaroslav Dytrych	Práce		JD		100%	300,00 Kč/hodina	450,00 Kč/hodina	0,00 Kč	Průběžně	24 hodin		
2	team	Práce		team		900%	700,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
3	manažer	Práce		m		900%	800,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
4	analytik	Práce		a		600%	800,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
5	návěhář	Práce		n		400%	800,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
6	vývojář	Práce		v		400%	400,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
7	tester	Práce		t		200%	500,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
8	kontrolér	Práce		k		100%	400,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
9	dokumentátor	Práce		d		180%	250,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
10	technik	Práce		h		100%	300,00 Kč/hodina	0,00 Kč/hodina	0,00 Kč	Průběžně	Standardní		
11	dokumentace	Materiál		dok			10 000,00 Kč		0,00 Kč	Průběžně			
12	server	Materiál		s			0,00 Kč		0,00 Kč	Průběžně			

PŘÍPRAVY NOVÉ ÚKOLY: RUČNĚ NÁPLŇOVANÉ





SLEDOVACÍ GANTTŮV DIAGRAM

- Tým má obvykle určité „jádro“ – členy, kteří se vzájemně znají, jsou spolehliví a mají aktivní přístup.
- Práce na 1. fázích projektu je obvykle pomalejší (nejtěžší je vždy začít) a měla by tedy být rozdělena mezi členy z jádra týmu, aby projekt začal co nejrychleji (i když se může jednat o jednodušší práci).
- Čím méně známý a potenciálně méně spolehlivý člen týmu, tím pozdější fázi projektu by měl mít na starosti.

- Mohou nastat různé problémy:
 - některý člen v průběhu práce opustí tým ,
 - některý člen týmu bude zdržovat („Zítra to dokončím a pošlu Vám to.“),
 - zdravotní problémy,
 - technické problémy (poruchy HW, sítě, ztráty dat, ...),
 - ...
- Tyto problémy je třeba zvažovat již při prvotním plánování prací na projektu.
 - Protože problém zatím nenastal, jedná se zatím pouze o riziko.
 - Detailně je řeší **řízení rizik**, které probíhá v průběhu celé práce na projektu (více viz předmět MPR).

- Jednoduché řešení pro malý tým a projekt:
 - Vždy je třeba mít rezervy alespoň pro výpadek 1/5 týmu
 - rezervu lze realizovat např. tak, že některý „spolehlivý“ člen týmu na začátku nemusí dostat žádnou práci a následně bude pomáhat ostatním nebo dostane práci, kterou někdo jiný neudělal.
 - Je výhodné mít alespoň 10 % času rezervu.

Úvod do návrhu software

- Začít ihned „živelně“ programovat není vhodné. Program je pak nutné často předělávat, jednotlivé části nemusí být možné spojit a kvalita výsledku může být nevyhovující.
- Libovolně malý program je vhodné navrhnout.
- U velkého programu je návrh nutností.

- analýza požadavků a příprava specifikace
 - Nevíme-li, co máme vytvořit, nemůžeme to správně navrhnout. Proto musíme vždy začít u požadavků zákazníka.
 - Zákazník nemusí mít dostatečnou představu o výsledku.
 - Požadavky mohou být nekompletní, protichůdné, chybné, ...
 - Na základě požadavků je potřeba vytvořit specifikaci, kterou by měl zákazník schválit.
 - Specifikace obsahuje
 - role uživatelů,
 - popis funkčnosti,
 - popis uživatelského rozhraní,
 - výkonnostní požadavky,
 - ...
- analýza existujících řešení
 - Nemá smysl znovu objevovat kolo.
 - Často lze pouze rozšířit existující řešení.
 - Zákazník bude vždy srovnávat.
 - Vyhledání vhodných knihoven vede k velké úspoře času (a financí).

- Vhodným jazykem pro návrh je UML („Obrázek vydá za tisíc slov“).
- Model obchodního (business) systému
 - má smysl zejména pro velké projekty (informační systém firmy apod.),
 - popisuje organizační model firmy a procesy ve firmě,
 - zahrnuje pouze ty aspekty, které budeme později potřebovat pro návrh IT systému.
- Model IT systému
 - popisuje vytvářený software a jeho nasazení na daném hardware,
 - se zabývá pouze činnostmi výpočetních prostředků (ne tím, co dělá uživatel bez jejich využití – např. ne „prodavač vezme zboží z pásu, zadá do pokladny a vloží do košíku“, ale „prodavač zadá množství a EAN kód“).

- Externí pohled
 - Diagram případů použití – kdo co dělá.
 - Diagram aktivit – postupy činností.
 - Diagram sekvence – interakce.
- Interní pohled
 - Diagram balíčků – součásti a zařazení do org. jednotek.
 - Diagram tříd – vztahy mezi zaměstnanci a objekty.
 - Diagram aktivit – interní procesy.

- Externí pohled (pohled zákazníka, black-box)
 - Diagram případů použití
 - Diagram sekvence případu použití
- Strukturální pohled (vnitřní struktura)
 - Diagram balíčků
 - Diagram tříd
- Behaviorální pohled (vnitřní chování)
 - Stavový diagram
- Interakční pohled (interakce uvnitř systému)
 - Diagram sekvence
 - Diagram komunikace
 - Přehledový diagram interakce
 - Diagram časování
- Pohled nasazení (který SW bude na kterém HW)
 - Diagram nasazení
- Další diagramy
 - Diagram objektů
 - Diagram složené struktury
 - Diagram kolaborace
 - Diagram komponent
 - ER diagram (není v základní sadě UML)

- Baumann, H.; Gräassle, P.; Baumann, P.: UML 2.0 in Action: A project-based tutorial. Birmingham: Packt Publishing, Prosinec 2009, ISBN 1-904811-55-8.
- http://ivs.fit.vutbr.cz/SID_jdytrych.pdf

- Ne vždy tvoříme velký SW systém, ale vždy bychom měli vytvořit návrh.
- Je nutné určit, které části návrhu budeme potřebovat a jak detailně je budeme zpracovávat.

- Diagram případů použití je vhodné vytvořit vždy.
 - Nemá smysl vytvářet něco, co nelze použít.
- Pracuje program se složitými datovými strukturami?
 - Strukturální pohled
 - Diagram balíčků, tříd, objektů, ...
- Provádí program složité činnosti?
 - Behaviorální pohled
 - Stavový diagram
 - Pro popis složitých procesů lze využít i **diagram aktivit**.
- Je v procesech zapojeno více SW komponent?
 - Interakční pohled
 - Diagram sekvence, komunikace, ...
- Je program složený z více částí, které budou vyvíjeny odděleně?
 - Diagram komponent, diagram složené struktury, ...
 - Všechna rozhraní musí být předem dobře definovaná.
- Bude složité nasazení (různý HW, SW dle umístění, ...)?
 - Pohled a diagram nasazení

- Nejprve vytváříme přehledové diagramy, abychom získali celkový pohled na systém:
 - Diagram případů použití
 - Diagram balíčků
 - Diagram komponent
 - ...
- Tam, kde diagram na nejvyšší úrovni nestačí k pochopení, z čeho se SW bude skládat, jak má fungovat a jak mají jednotlivé části spolupracovat, vytváříme diagramy popisující detaily:
 - Diagram tříd
 - Diagram objektů
 - Diagram sekvence
 - ...

- Nejprve vytvoříme přehledové diagramy:
 - Diagram případů použití
 - Diagram komponent
 - ...
- Následně se zaměříme na menší části, které lze brzy implementovat a otestovat.
- Začneme diagramy na nižší úrovni:
 - Diagram tříd
 - Diagram komunikace
 - Diagram složené struktury
 - ...
- Poté příbuznou funkcionalitu seskupíme do balíčků a doplníme abstraktnější diagramy.

- Návrh musí být dostatečně detailní, aby pokryl vše důležité, co není jednoznačné.
 - Vývojář obvykle pracuje tak, aby se vydal cestou nejmenšího odporu z hlediska psaní kódu. Z hlediska uživatele je pak funkcionální neologická.
 - Návrhář se musí zaměřit na použití a nesmí při tom příliš uvažovat o implementaci.
 - Návrh by měl být dostatečně detailní, aby implementace nezměnila zamýšlený průběh případů použití.
 - Nelze očekávat, že se dva vývojáři dohodnou na vhodném rozhraní.
 - Nelze očekávat, že vývojář správně navrhne část databáze tak, aby ji bylo možné efektivně využít i mimo jím vytvářené komponenty.
 - ...
 - Nemá smysl popisovat, že se připojíme k databázovému serveru, vybereme databázi, sestavíme dotaz, provedeme dotaz, ...
 - Je-li jasně dané vnější chování komponenty, vývojář aplikuje běžné postupy a dané chování implementuje i bez detailního popisu na úrovni jednotlivých řádků či bloků kódu.

- Dobrá a efektivní komunikace v týmu je základem týmové spolupráce.
- Pro sdílení dat v týmu je třeba zvolit vhodné prostředky a využívat je k vhodným účelům.
- Vhodné naplánování projektu umožní předejít problémům s jeho dokončením v daném termínu.
- Návrh je důležitou součástí procesu vývoje SW. Pro vytvoření kvalitního a použitelného SW je důležité tuto fázi nepodcenit.