

Petr Krčmář



*Uzavřený a otevřený vývoj:  
katedrála a tržiště*

# Stručná osnova

- Historie otevřeného vývoje
- Otevřený versus uzavřený vývoj
- Jak probíhá otevřený vývoj
- Specifika otevřeného vývoje
- Proč a jak to vlastně funguje
- Kdo se do vývoje zapojuje



# Otevřený vývoj – žhavá novinka?

- V 50. a 60. letech akademická půda
- Není samostatná komodita
- Otevřený vývoj – spolupráce
- Systémy i software volné šíření
- Včetně zdrojových kódů (modifikace, vylepšení)
- Vývojáři = dnešní „komunita“
- Software z principu svobodný – žádné cílené úsilí svobodných vývojářů jako dnes

# Rozštěpení software

- Změna na konci 60. let
- Přibalování software k hardware; roste cena
- V 70. letech AT&T UNIX zdarma, ale nesvobodný (freeware)
- 70. a 80. léta vznik samostatných SW spol.
- Použití autorského práva, licencí, ochr. známek
- 1976 Bill Gates „Otevřený dopis kutilům“
  - Sdílení podle něj = krádež

# Uzavřený vývoj +

- Vývoj v malých skupinách (3 až 4 lidé)
- Efektivní, kontrolovatelný, plánovatelný
- Struktura: management → vývojáři
- Vývoj řídí management – výhoda i nevýhoda
  - Dobrá reakce na zákazníky, podřízení penězům (IE6)
- Formální podpora a školení
- (relativní) rychlost vývoje

# Uzavřený vývoj -

- Složitá komunikace týmů přes management (bez přímého zapojení nemají „mapu“)
- Pro uživatele problém s úpravami
- Přimknutí k dodavateli (vendor lock-in)
- Vylepšení za každou cenu (samokonkurence)
- Špatná interoperabilita (každý vlastní písek)
- Pomalejší oprava chyb (omezené HR)

# Poznámka: přeplánovanost



*„V dubnu Microsoft  
vydá osm záplat“*

*(Zdroj: Lupa.cz)*

# Otevřený vývoj +



- Nové verze s novými funkcemi
- Možnost spolupráce mnoha firem i jednotlivců
- Volné konkurenční prostředí
- Možnost lepší integrace do stávajících systémů
- Velmi dobrá interoperabilita
- Kontrola nad funkcí
- Výrazně rychlejší oprava chyb (FF 1, Safari 5, MSIE 9 dní)



# Otevřený vývoj -

- Chybějí dlouhodobé plány
  - Ale ty často nezveřejňují ani cs vývojáři
- (relativně) nižší rychlost vývoje
- Nové uživatelské funkce (vývojáři sobě)
- Chybí centrální školení a podpora
- Virálnost licencí (?)

# Specifika otevřeného vývoje



- Rozsáhlá komunikace vývojářů
- Vše je dostupné a zpětně dohledatelné
- Neexistuje pevně stanovená hierarchie
  - Vývojářská struktura se neustále mění
- Vedoucí osobnosti nepořádají meetingy
  - Přímo komunikují s vývojáři
- Mnoho podnětů přichází zvenčí (včetně kódu)
  - Ono totiž žádné pravé „uvnitř“ není

# Otevřený vývoj v praxi

- Také v malých skupinkách (malé projekty)
- Větší projekty rozdrobené na menší (KDE, Debian)
  - Skupiny spolu stále komunikují
- Existuje přirozená hierarchie (žádní umělí manažeři)
- Otevřenost = jasně viditelná zodpovědnost
- Přímá komunikace s vývojářem (ne přes manažery)
- Kód je na prvním místě

# Procesy v projektu

- Kód je na prvním místě
  - Nejdřív programuj, pak řeš ostatní
- Marketing je podstatný (webové stránky)
  - Ale jiný smysl: ne shánět placící uživatele
  - Shánět aktivní uživatele = vývojáře
- Vypustit software do světa
  - Vyhledávat a opravovat chyby
  - Napsat dokumentaci (možná)

# Proč je otevřený vývoj možný

- Internet; kdokoliv se může přidat
- Stačí jen znalosti (JXD: Boeing ve stodole)
- Vývojáři jsou zároveň uživateli
- Vysoká modularita (patch jedné funkce neovlivní zbytek)
- Veřejná diskuse (kód nikomu nepatří)
- Svobodné licence (možnost forku)
- Nelíbí se? Napiš si svoje.

# Kdo se zapojuje?



- Jádro Linux: Red Hat, Novell, IBM, Intel (> 1/3)
- Firmy iniciující otevřený vývoj
  - Red Hat, Sun, Novell, Canonical, MoCo...
- Velké zastoupení vnějších vývojářů
  - RH už by sám projekt neutáhl
- Dále se zapojují: IBM, Oracle, Google, HP, Cisco...
- Ale i mnoho dalších firem (Seznam.cz)

Máte otázky?

*Petr Krčmář*

*www.root.cz*

*www.debian-linux.cz*