

# Červená nitka vede od Mínotaura

aneb Dokumentace jako vodítko po labyrintu života.

Lukáš Růžička (lruzicka@redhat.com)

16. dubna 2018, FIT ČVUT Praha

# Žádná dokumentace, dobrá dokumentace?

Dokumentace není potřebná, když je každému hned jasné, co a jak má s danou věcí dělat. Tedy danou věc lze použít

- pouze jedním způsobem
- s pevně daným výstupem
- a nemožností selhání

# Tak raději přece?

O dokumentaci byste měli uvažovat vždy, když:

- existuje více možností, jak danou věc použít
- věc nabízí více než jednu funkci
- je možné, že výstup se může lišit v závislosti na podmínkách použití
- věc nemusí vždy fungovat správně

# Úkol pro vás

Na základě zmíněných kritérií jmenujte jednu věc, pro kterou nepotřebujete dokumentaci.



Bohové musí být šílení (1980)

# K přemýšlení

- Když lidé nebudou o vašem projektu vědět, nebudou ho používat.
- Když lidé nebudou umět nainstalovat váš program, nebudou ho používat.
- Když lidé nepoznají, jak použít váš kód, nepoužijí ho.
- Když váš program nebude mít očekávané výsledky, lidé ho nebudou používat.

# Jaký máme výběr?

Mezi dokumentaci můžeme zahrnout několik různých forem:

- konceptuální
- procedurální
- referenční
- výukovou

# Konceptuální dokumentace

**Konceptuální** dokumentace se snaží vysvětlit uživatelům danou problematiku (**koncept**) tak, aby tito pochopili princip fungování a potřebné souvislosti. Například:

- Co je počítačová síť a jak se po ní posílají datagramy?
- Co jsou optické vlastnosti objektivu a jak ovlivňují výsledné foto?
- Jak funguje útok *hrubou silou* a proč je důležité volit si komplikovaná přístupová hesla?
- Jak působí bakterie zubního plaku a jak zvolit správný kartáček?



# Procedurální dokumentace

**Procedurální** dokumentace poskytuje jasný návod, většinou rozdělený na jednotlivé kroky (**procedura**), ke splnění uživatelského záměru (**user case**). Mimo jiné:

- Jak nainstalovat aplikaci XY.
- Jak napsat a odeslat email pro úplné začátečníky.
- Jak vytvořit nového uživatele a přidělit mu patřičná systémová práva.
- Jak provést předletovou kontrolu osobního letadla Boeing 747.

# Referenční dokumentace

**Referenční** dokumentace nabízí ucelený přehled (**referenci**) vlastností, voleb, nastavení a způsobů použití, aby si uživatel mohl sám objevit vlastní přístup a sestavit si své vlastní postupy:

- manuálové stránky v Linuxu
- přehled typů žárovek pro osvětlení vozidla
- přehled ingrediencí potřebných pro upečení dortu
- přehled voleb příkazu `dnf` ve Fedoře.

# Výuková dokumentace

**Výuková** dokumentace, tzv. **tutoriály**, je spojení konceptuální a procedurální stránky dokumentace, takže výsledkem není jenom splněný uživatelský záměr, ale také částečné pochopení problematiky a kontextu.

Je velmi důležité, abychom z konceptuálního hlediska vysvětlili pouze tolik, kolik je nezbytně nutné. Chceme-li pomoci tutoriálů vysvětlit širší problematiku, pak volíme několik na sebe navazujících.

# Úkol pro vás

Navrhněte dokumentaci pro nějakou věc denní spotřeby.  
Kterou formu zvolíte? Proč?

# Správný styl dokumentace

V dokumentaci bychom měli zachovávat několik stylistických pravidel:

- jednoznačné výrazy
- jednoduché a krátké formulace
- neutrální výrazy
- genderově neutrální prvky
- v případě pochybností je lepší více informací než méně
- rozkazovací způsob (procedury)

# Nesprávný styl dokumentace

V dokumentaci bychom se raději měli vyhnout:

- estetickým formám jazyka (metafory, přirovnání, nadsázka)
- ironii a sarkasmu
- humoru
- zlehčování problémů
- utěšování uživatelů

# Porušování pravidel

Někdy můžeme uznat za vhodné pravidla porušit a získáme tak jinou formu dokumentace, jež je

- zajímavá
- neotřelá
- vtipná
- parodická

Je však nutné si uvědomit, kdo budou čtenáři naší dokumentace a jsou-li tito ochotni takový přístup snášet.

# Úkol pro vás

Jakou věc by bylo možné dokumentovat s porušením předchozích pravidel? Jakou věc byste nikdy nechtěli takto dokumentovat?



# Technické zpracování

Dokumentaci můžeme psát ve spoustě různých formátů. Záleží, co přesně od dokumentace očekáváme. Můžeme si vybrat na základě následujících kritérií:

- obtížnosti tvorby dokumentace (WYSIWYG, markdown, markup, ...)
- metod zobrazení (web, čtečka, tištěné médium, audiovizuální metody)
- možností spolupráce (žádná, cvs, git, Google)
- možností publikace (ručně, automaticky (CI))
- dostupnosti metadat textu (sémantický markup)

V podstatě platí, že čím více muziky čekáme, tím více do dokumentace budeme muset investovat.

# Formáty

Pokud se v souvislosti s dokumentací mluví o formátech, pak máme na mysli v podstatě dva typy a to:

- **zdrojové** formáty, tedy ty, ze kterých se dokumentace překládá
- **cílové** formáty, tedy ty, do kterých se překládá

Některé souborové formáty mohou být jak zdrojové, tak cílové (HTML)

# Stručný přehled zdrojových formátů

- čistý text
- markdown (Markdown, AsciiDoc)
- markup (HTML, XML, DocBook, Mallard, reST,  $\text{\LaTeX}$ )
- nativní formáty WYSIWYG aplikací (doc, docx, fm, odt, indd)

# Stručný přehled cílových formátů

- HTML (web)
- epub, mobi a další (ebooky)
- pdf (tiskárna)

# Nejčastěji používané formáty

- Markdown (Github)
- AsciiDoc (Red Hat)
- reST (Python)
- Mallard (Gnome)
- T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X
- DocBook (SuSE, Red Hat, IBM)

# Markdown

## Výhody:

- jednoduchý
- čitelný

## Nevýhody:

- omezené typografické možnosti
- minimální sémantika
- nejasné hranice mezi prvky
- významy některých značek kolidují s reálnými znaky (\*)

Markdown je vhodný především pro malé projekty, například README na Githubu.

# AsciiDoc

## Výhody:

- jednoduchý a čitelný
- částečně programovatelný
- lze psát sémantický kód

## Nevýhody:

- nejasné hranice mezi prvky
- vlastní úpravy porušují standard (kompatibilita)
- sémantika zvyšuje složitost a omezuje čtivost
- kolidující značky

AsciiDoc je vhodný pro malé i velké projekty, ale v základu postrádá některé funkce, které jiné formáty nabízejí.

# reST a Sphinx

## Výhody:

- jednoduchý a čitelný
- programovatelný
- sémantický
- jasné hranice mezi prvky

## Nevýhody:

- nutnost přesně dodržovat strukturu kódu

reST, a jeho rozšíření Sphinx, jsou velmi vhodné pro veškeré typy projektů. Jsou oblíbené především mezi programátory Pythonu, protože struktura formátu připomíná Python.



# Mallard

## Výhody:

- sémantický
- jasné hranice mezi prvky
- modulární

## Nevýhody:

- méně přehledný a čtivý
- poměrně neznámý
- modulární

Mallard byl původně určen pro psaní **Gnome** help. Je navržen pro *topic based authoring* (tématicky zaměřené psaní), takže zdrojový kód se spíše skládá z fragmentovaných prvků než velkých celků.

# T<sub>E</sub>X, L<sub>A</sub>T<sub>E</sub>X

## Výhody:

- sémantický
- jasné hranice mezi prvky
- poměrně přehledný a čtivý
- dlouhodobě vyvíjený

## Nevýhody:

- nehezký výstup do HTML
- mnoho odnoží s různými výstupy a funkcemi

Sázecí systém T<sub>E</sub>X a sada maker zjednodušujících práci s ním, L<sub>A</sub>T<sub>E</sub>X, byl vyvinut pro sazbu matematických publikací. Nejlepší výsledky poskytuje pro vytváření tištěné dokumentace.

# Docbook

## Výhody:

- sémantický
- jasné hranice mezi prvky
- dlouhodobě vyvíjený, tedy robustní
- systém validace nedovolí chyby

## Nevýhody:

- nepřehledný
- mnoho různých prvků a vztahů mezi nimi
- vyžaduje čas na naučení
- neodpouští chyby

Docbook je pravděpodobně nejlepší volbou pro psaní korporátní dokumentace.

# Výběr vhodného formátu

Před výběrem formátu pořádně zamyslet, co od něj přesně chceme, neboť špatný výběr formátu může celou tvorbu dokumentace zkomplikovat.

Ptejme se na:

- formu spolupráce
- obtížnost psaní
- automatizované testování a publikování
- formy výstupu
- přenositelnost do jiných formátů
- budoucnost projektu

Obecně platí, že **čím složitější** formát, tím **více možností použití**.

# Úkol pro vás

Jaký formát byste vyzkoušeli jako první, kdybyste si měli vybrat na základě poskytnutých informací? Proč?

# Obvyklý postup při vytváření dokumentace

Dokumentace obvykle vzniká následujícím postupem:

- 1 zjišťování potřeb uživatelů
- 2 plánování a alokace zdrojů
- 3 stanovení formy spolupráce a formátu
- 4 tvorba zdrojových dokumentů
- 5 překlad do cílových dokumentů
- 6 publikování
- 7 vyhledání a opravení chyb
- 8 znovu přeložení a publikování (nová subverze)

# Dokumentace v Red Hatu obsahově

Red Hat v současné době udržuje několik dokumentačních projektů. Organizační struktura dokumentačního oddělení používá tři oddělené pozice:

- obsahový stratég (content strategist)
- programový manažer (document program manager)
- dokumentátor (technical writer)

Každý z nich je zodpovědný za jinou oblast vývoje dokumentace, leč jejich funkce se mohou v Red Hatu překrývat, protože Red Hat pracuje na principu *meritokracie* a platí, že nejlepší názor se počítá.

# Dokumentace v Red Hatu technicky

Z hlediska technického zpracování se používají dva postupy, které jsou založené buď na:

- Docbooku
- AsciiDocu

Docbook je v současné době na ústupu a projekty přecházejí na AsciiDoc.



# Pracovní postup v Docbooku

Při vytváření dokumentace psané v Docbooku se postupuje takto:

- klonování Git repozitáře
- ruční úprava dokumentů v textových editorech
- sledování změn v paralelní Git větvi (branch)
- validace a překlad na lokálním stroji (publican)
- peer review → kontrola kvality
- automatizované testy obsahu (překlepy, názvy)
- merge do hlavní větve
- překlad hlavní větve (publican) a vystavení na portálu ([www.redhat.com](http://www.redhat.com))

# Pracovní postup v AsciiDocu

Při vytváření dokumentace psané v AsciiDocu se postupuje takto:

- klonování Git repozitáře
- ruční úprava dokumentů v textových editorech
- sledování změn v paralelní Git větvi (branch)
- validace a překlad na lokálním stroji (asciidocctor, ccutil)
- peer review → kontrola kvality
- merge do hlavní větve
- překlad hlavní větve (ccutil → publican) a vystavení na portále

# Probíhající diskuse o dokumentaci

V dokumentačním týmu v současné době probíhá několik diskusí a iniciativ o tom, jak dělat věci lépe, aby zákazník dostal za svoje peníze co nejlepší službu.

- spolupráce upstream  $\longleftrightarrow$  downstream
- konzistentní terminologie (IBM Style Guide)
- Docbook versus AsciiDoc
- správný postup u peer review
- jednotný způsob zápisu značek u AsciiDocu
- sémantický zápis značek v AsciiDocu
- modulární dokumentace

# Otázky

Kdo se moc ptá, moc se doví.

Takže?

Děkuji za pozornost.