

Jak na to napsat jak na to?

aneb Jak přežít dobu dokumentační

Lukáš Růžička

lruzicka@redhat.com

listopad 2019

Žádná dokumentace, dobrá dokumentace?

Dokumentace není potřebná, když je všem hned jasné, jak daný produkt funguje, co je s ním možné provádět, a co si počít, když zase tak dobře nefunguje.

Tedy:

- věc lze použít jediným způsobem
- vždy to dopadne stejně
- nikdy se to nerozbije, nebo alespoň ne tak, aby nebylo jasné, že se to rozbilo a proč.

Takže? Třeba **džbán**? Ale věděli jste, že tak dlouho se chodí se džbánem pro vodu, až se ucho utrhne?

21. století nazýváme stoletím dokumentace

Nevěříte? Zrovna, když reviduju tuto prezentaci, na Googlu probíhá téměř 78000 hledání za vteřinu. Proč?

Jak si v Google statistice stojí "How to" a "jak na to"?

- **how to install linux** chtějí vědět v Barmě (cca 80)
- **how to fry eggs** zajímá občany Jihoafrické republiky (cca 50)
- **how to change oil** hledají v USA (cca 75)
- **how to make tea** je populární na Trinidadu (cca 75)
- **how to write documentation** vládne v Indii (cca 50)

Další údaje naleznete na <https://trends.google.com>.

Dokumentace je prostě všude ...

... protože život je zkrátka příliš složitý.

- YouTube
- diskusní servery a blogy
- infografiky
- vysvětlivky
- školní výuka a učebnice
- rada od přátel
- a mnoho dalších

Dokumentace v teorii

Jakým způsobem můžeme dokumentovat?

Existují čtyři základní přístupy k dokumentaci:

- konceptuální
- procedurální
- referenční
- výukový

Konceptuální dokumentace

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

Například:

- jak funguje volební systém České republiky
- jak se přenáší terestriální signál
- jak funguje RAID
- jak pracuje rekuperace v elektromobilu

Procedurální dokumentace

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

- jak nainstalovat aplikaci **X**.
- jak si uvařit vlastní pivo
- jak vytvořit nového uživatele
- jak si vyjednat slevu u mobilního operátora

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 pomocí tutoriálů vysvětlit širší problematiku, pak volíme několik na sebe navazujících.

Správný styl dokumentace

- 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

- estetické formy jazyka (metafory, přirovnání, nadsázka)
- hovorové výrazy
- ironie a sarkasmus
- humor
- 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.

Technické zpracování

Dokumentaci můžeme psát ve spoustě různých formátů. Záleží, co přesně od dokumentace očekáváme:

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

Dále se budeme zabývat pouze **psanou** dokumentací.

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).

Některé zdrojové formáty

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

Některé cílové formáty

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

Nejčastěji používané formáty

- Markdown (Github)
- AsciiDoc (Red Hat)
- reST (Python)
- Mallard (Gnome)
- TeX, LaTeX (vydavatelské domy)
- DocBook (SuSE, IBM)

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.

Dokumentace v praxi

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 z hlediska organizace

Red Hat v současné době udržuje mnoho dokumentačních projektů. Organizační struktura dokumentačního oddělení má několik **pilířů**:

- obsahový stratég (content strategist)
- technický manažer (pillar lead)
- manager pro lidské zdroje (people manager)
- programový manažer (document program manager)
- dokumentátor (technical writer)

Každý z nich je zodpovědný za jinou oblast vývoje dokumentace.

Dokumentace v Red Hat z hlediska použitých metod

- Většina dokumentačních projektů v Red Hatu používá **AsciiDoc** jako hlavní syntaxi pro vytváření zdrojových textů.
- Totéž se týká dokumentace **Fedory**.
- Mnoho projektů používá tzv. **modulární přístup** a upřednostňuje jej před klasickým přístupem.

Modulární výhody

- výstup lze poskládat pro každého jinak (customizace)
- jednou napsaný modul lze použít na více místech textu (reusability)
- lepší orientace v textu

Šedivá je teorie a zelený je strom života

- naprogramovat mechanismus pro customizaci není jednoduché
- použít modul na více různých místech je složité až nemožné bez řádných metadat
- bez utilit, které dokážou mezi moduly hledat, je ruční hledání zdlouhavé
- lidské ego se nerado dělí

Docs-As-Code

Tohle je poměrně moderní, ale už dost rozšířený, způsob, kdy s dokumentací zacházíme jako s kódem a používáme:

1. popisovací jazyk (Markdown, AsciiDoc, LaTeX, Docbook)
2. sdílený repozitář (Gitlab, Github)
3. správu verzí (Git, SVN)
4. kontinuální integraci (CI)
5. automatické publikování (web)

Praktická ukázka Docs-As-Code

Využití Githubu

GitHub (github.com) dokáže překládat soubor `README.md` příslušného projektu a ten zobrazovat v HTML formátu na adrese `uzivatel.github.io/repositar`.

Tato prezentace je také na Githubu jako repozitář `cvutkurz` a je mimojiné dostupná přímo na <http://dokumentarista.github.io/cvutkurz>.

Opravujeme a vytváříme

1. Klonování Git repozitáře

```
git clone https://github.com/dokumentarista/cvutkurz.git
```

2. Vytvoření lokální větve pro vývoj

```
git checkout -b novy_slide
```

3. Změny v nové větvi

4. Přidat změny do gitu

```
git add .
```

```
git commit -m "Provedene zmeny"
```

5. Nahrát změny na server.

```
git push [--set-upstream origin novy_slide]
```

6. Pokračovat ve změnách a opakovat kroky 4 a 5 dle potřeby.

Správa verzí

1. Verze se uchovávají ve **větvích** (branch)
2. V nich probíhá vývoj.
3. Posléze se změny převedou do `master` větve (merge).
`git merge novy_slide`
4. Nahrání upraveného *masteru* na server
`git push --force`
5. Za chvilku se změny projeví (tzv. *continuous integration*) a v hlavní větvi se objeví obsah vývojové větve.

Hustě nahuštěná pneumatika

Správné použití vzduchového kompresoru:

1. Odpojte hadici od kola a smotejte ji ke kompresoru.
2. Našroubujte kryt ventilku.
3. Stisknutím a přidržením tlačítka **Plus** zvyšte tlak na požadovanou hodnotu, nebo jej pomocí tlačítka **Minus** snižte.
4. Zajed'te s autem co nejbliž'e kompresoru, abyste tlakovou hadici dosáhli na každé kolo.
5. Z manometru na kompresoru odečt'ete současný tlak v pneumatice.
6. Rozmotejte tlakovou hadici a připojte ji na ventilek.
7. Odšroubujte kryt ventilku.

Pište dokumentaci pro Fedoru

Fedora je komunitní distribuce Linuxu, na jejímž vývoji se může podílet kdokoliv. Tak i na její dokumentaci.

- repozitáře na pagure.io (Git)
- využívá projekt Antora (antora.org)
- vydána na docs.fedoraproject.org

Příspěť můžete po přečtení návodu <https://docs.fedoraproject.org/en-US/fedora-docs/contributing/>

Otázky a odpovědi

Proč, když líná pes, tak mu padají chlupy, ale když **líná huba**, tak je **holé neštěstí** a ne ta huba, která líná?