

HLAVNÉ HRIECHY TVORCOV DOKUMENTÁCIE

*Kvalitná dokumentácia znamená zníženie nákladov
a zvýšenie spokojnosti zákazníkov.*

Juraj Kostolanský

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
juraj[.]kostolansky[zavináč]gmail[.]com

Abstrakt. *Technická dokumentácia, používateľská príručka, komentáre zdrojového kódu. Hoci sa dnes dokumentácia považuje za neoddeliteľnú súčasť softvérového produktu, často sa jej tvorbe venuje oveľa menšie úsilie, než by si zaslúžila. Napriek tomu, že kvalitná dokumentácia vedie nielen k spokojnejším zákazníkom, jednoduchšiemu zorientovaniu sa v zdrojovom kóde pre programátora, ale aj k úspore času a nemalých finančných prostriedkov, stále je považovaná skôr za nevyhnutné zlo než za spôsob zjednodušenia si práce a skvalitnenia výsledného produktu. V tejto eseji sa zameriavam na potrebu dokumentovania softvéru z viacerých pohľadov, zamýšlam sa nad pojmom kvalita v kontexte dokumentovania a opisujem základné chyby, ktorých sa tvorcovia dokumentácie dopúšťajú.*

Kľúčové slová: *dokumentácia, manuál, príručka, kvalita dokumentácie, chyby dokumentácie, manažment dokumentácie*

Úvod

Každý sa s ňou už stretol. Ak si kúpime novú práčku alebo autorádio, siahneme najprv po používateľskej príručke. Dozvieme sa v nej, ako výrobok rozbaľiť, zložiť, nainštalovať, ako s ním správne zaobchádzať, ako sa oň starať a na čo si dávať pozor. Druhá časť používateľov, ktorá uprednostňuje metódu pokus–omyl, siahne po dokumentácii až v prípade, že si s niečím nevie rady.

No je to tak aj s dokumentáciou k softvéru? Koľkí z nás ju bežne čítajú, ak narazia na problém s aplikáciou? Z vlastnej skúsenosti viem, že z pohľadu používateľa je

jednoduchšie hľadať riešenie svojho problému na rôznych internetových fórach a diskusiách. Prečo je to tak? Sú na vine zvyky používateľov? Alebo za to môžu nekvalitné a nezrozumiteľné dokumentácie, v ktorých sa čitateľ cíti stratený? Myslím, že oboje. Sú to aj nekvalitné dokumentácie, ktoré viedli používateľov k vytvoreniu určitých návykov pri hľadaní informácií.

Potreba dokumentovať

Pre tvorcú softvéru by mala byť na prvom mieste funkčnosť výrobku, ktorý bude odovzdaný zákazníkovi. Mal by sa sústrediť na zdrojový kód, ktorý produkuje, na optimálnosť použitých algoritmov, snažiť sa pracovať efektívne a svoju prácu ukončiť v čo možno najkratšom čase.

Možno preto sa mnohým zdá, že písanie dokumentácie – či už používateľskej príručky, technickej dokumentácie alebo komentárov zdrojového kódu – je len strata času. Ich názor môže podporiť aj prvý pohľad na moderné metódy softvérového inžinierstva, napr. v manifeste agilného vývoja sa môžu dočítať o zásade uprednostnenia fungujúceho softvéru pred obsiahlou dokumentáciou (z angl. *working software over comprehensive documentation*).

Avšak každý programátor, ktorý sa počas tvorby softvérového produktu pripojil k zabehnutému tímu, už iste zistil úskalia zorientovania sa v cudzom kóde a tým aj nevyhnutnosti tvorby dokumentácie. Pochopenie cudzích myšlienok a návykov býva často zložitá a časovo náročná. Takýto „nováčik“ väčšinou radšej siahne po softvérovej dokumentácii. Čím je kvalitnejšia, tým skôr pochopí zaužívané zvyklosti pri tvorbe softvéru v preňho novej spoločnosti.

V prvom rade môže siahnuť po používateľskej príručke. V nej sa môže dočítať o základných informáciách o vyvíjanom produkte, o implementovaných funkciách, naučí sa s programom pracovať z pohľadu používateľa. Druhým krokom sú interné metodiky, v ktorých môže nájsť informácie o zaužívaných praktikách v spoločnosti, spôsobe riadenia, časových harmonogramoch, postupoch a pod. Keď už pochopí, ako sa veci majú, pomôže mu so samotnou prácou technická dokumentácia. Či už to sú komentáre k zdrojovému kódu, ktoré uľahčia pochopenie (aj) zložitých algoritmov, alebo technický opis jednotlivých častí softvéru, napr. tried, rozhraní, metód a pod.

V prípade, že ide o tzv. softvér s otvoreným zdrojovým kódom (z angl. *open source software*), majú tieto dokumenty možno ešte väčší význam. Používateľská príručka slúži na nalákание používateľa. V prípade, že je týmto používateľom vývojár-dobrovoľník, ktorý by sa rád podieľal na zlepšení produktu, ozrejmi mu ďalší potrebný postup dokument, v ktorom sú opísané praktiky spoločnosti. Uvidí, ako sú vývojári riadení, aké prípadné benefity získavajú spoluúčasťou na tvorbe tohto programu, aké sú vyžadované znalosti. Nedostupnosť týchto informácií môže odradiť veľké množstvo dobrovoľníkov, ktorí by radi pomáhali, no nevedia ako.

Pre zákazníka má najväčší význam používateľská príručka. Väčšinou po nej siahne v prípade, že si s niečím nevie rady - môže to byť napríklad zložitejší postup pri používaní softvéru alebo zisťovanie príčiny nečakaného pádu aplikácie.

Používateľská príručka nemusí mať význam iba pre zákazníka – ak je (aspoň sčasti) vytvorená skôr, než samotný produkt. Účelom takto „predčasne“ vytvorenej

dokumentácie je overenie požiadaviek klienta. Môže si dokumentáciu prečítať, môže ju odovzdať ľuďom, ktorí s vyvíjanou aplikáciou budú neskôr pracovať. Takto tvorcovia programu získajú akúsi spätnú väzbu na nedostatky návrhu skôr, než je napísaný jediný riadok zdrojového kódu, čo môže v konečnom dôsledku ušetriť čas aj nemalé peniaze.

Hriechy tvorcov dokumentácie

Ako spoznáme, že softvérová dokumentácia je kvalitná? Rozhodnutie o kvalite produktu by malo vychádzať z merateľných faktov. Kvalitným monitorom nebude ten, ktorý dobre vyzerá, ale ten, ktorý verne zobrazuje farby, má správne rozlíšenie, dlhú životnosť a nízku spotrebu. No kedy môžeme povedať o dokumentácii, že je kvalitná? V tejto otázke sa stotožňujem s názorom uvedeným v [1]: "Keď stlačíte vo výfahu tlačidlo s číslom 3 a ten vás vezme na štvrté poschodie, viete, že sú tu problémy s kvalitou, ktoré treba riešiť." Pri posudzovaní kvality nemusíme vždy vychádzať z merateľných faktov. Kvalitnú dokumentáciu poznáme, keď ju čítame. Nepotrebujeme k tomu špeciálne vzdelanie. Vystačíme si s intuíciou. Napriek tomu však existuje niekoľko základných pravidiel, ktorých by sa mal pridržiavať každý tvorca dokumentácie.

Chybovosť

Chybám sa treba vyhýbať v každom písanom texte. Preklepy a gramatické chyby v knihách a článkoch nepomáhajú pozdvihnúť reputáciu autora. No chyby v dokumentácii nie sú len o ypsilonoch, čiarkach a preklepoch. Chybne podané informácie môžu mať fatálne následky. Čo ak by sa v príručke pre používateľa namiesto príkazu pre kopírovanie súboru nechtiac objavil príkaz pre zmazanie obsahu pevného disku? Následky si vieme predstaviť všetci - obzvlášť tí, ktorí svoje dáta nezálohujú. Preto je dôležité dbať na správnosť všetkých údajov uvedených v dokumentácii. Na rozdiel od iných textov, v softvérovej dokumentácii závisí na každom detaile [1].

Nekompletnosť

Mnoho technických dokumentácií je nekompletných. Tento fakt súvisí najčastejšie s tým, kto je ich autorom. V prípade softvérových produktov sú to najmä ich tvorcovia. Tí produkt navrhli a implementovali, takže často poznajú každý kúsok aplikácie. Z tohto dôvodu sa nedokážu na softvér pozrieť očami používateľa, ktorý často nemá technické vzdelanie a s programom prichádza do styku po prvýkrát [1].

Spolupráca tvorcov dokumentácie s technickou podporou spoločnosti môže prispieť k výraznému zlepšeniu kvality obsahu dokumentácie. Personál technickej podpory má na rozdiel od tvorcov dokumentácie (ktorými sú často technicky orientovaní vývojári) lepšie vedomosti nielen o používateľoch, ale aj o najčastejších problémoch pri využívaní softvérového produktu. Na základe častej a priamej komunikácie s používateľmi, ktorí potrebujú pomoc, sa vedú lepšie vcítiť do ich potrieb a zmýšľania. Vedú informácie podať zrozumiteľnejším spôsobom, pričom využívajú termíny, ktoré sú zákazníkom bližšie (čo je vlastne ich práca). Podľa [3] je potrebná nielen komunikácia týchto dvoch oddelení spoločnosti, ale aj jednotný a konzistentný spôsob ich kooperácie. Autor na základe skúseností navrhuje (ako jedno z možných riešení) využitie akejsi databázy

znalostí. Tá obsahuje otázku zákazníka a poznámky zamestnanca podpory, ktoré vedú k jej vyriešeniu. Bežne sa takáto databáza využíva ako interný systém technickej podpory, avšak jej využitie pri tvorbe dokumentácie už tak bežné nie je. Základný scenár vybavovania otázok zákazníkov je v prvom rade hľadanie informácie v takejto databáze. Tým pracovník podpory zistí, či sa už podobná otázka v minulosti riešila. V prípade, že nie, využíva pracovník k hľadaniu riešenia dokumentáciu a po vyriešení požiadavky pridá záznam do databázy riešení. "Personál technickej podpory je jedným z najdôležitejších čitateľov dokumentácie." [3]

Páči sa mi myšlienka spolupráce pisateľov dokumentácie s oddelením technickej podpory prostredníctvom jednotného systému – databázy znalostí. Takýto prístup umožňuje zistiť, ktoré časti dokumentácie sú dôležité, ktorým častiam používateľ najmenej rozumie (t.j. najčastejšie využíva technickú podporu), ktoré sú, naopak, menej zaujímavé a môžu byť redukované, prípadne vypustené. V prípade, že databáza znalostí obsahuje aj referenciu na konkrétnu časť dokumentácie, ktorá sa zaoberá otázkou zákazníka, môže sa zamestnanec podpory na túto časť odvolať bez pracovného hľadania a rýchlejšie reagovať na potreby používateľa (najmä v prípade, že zákazník uprednostňuje kontakt s podporou pred čítaním dokumentácie). V takomto prípade záznam bez odkazu na časť dokumentácie môže tiež znamenať chýbajúcu informáciu, ktorú je potrebné doplniť. Možnosti takéhoto systému sú veľké a vedú nielen k rýchlejšej odozve zákazníckej podpory, ale aj ku kvalitnej dokumentácii „na mieru“, spokojnosti zákazníka a celkovému zníženiu nákladov.

Bezhlavé zavádzanie takéhoto spôsobu spolupráce prostredníctvom jednotného systému však môže byť aj kontraproduktívne. Aj tu platí, že sa treba prispôbiť konkrétnemu problému. Ak ide o veľkú spoločnosť, toto rozhodnutie je na mieste. Ak však ide o „garážovú firmu“ troch či štyroch kamarátov, je na zváženie, či rozsiahly systém spolupráce medzi tvorcom dokumentácie a technickou podporou prinesie výhody voči priamej interakcii jednotlivých členov tímu.

Nejasnosť

Hoci vytvoriť správnu a kvalitnú dokumentáciu je umenie, nejde o umelecký štýl písania. Úlohou tvorcu dokumentácie nie je zabaviť čitateľa napínavým príbehom, ale zrozumiteľne a vecne mu podať potrebné informácie. "Úlohou dokumentácie nie je zaujať, ale jasne informovať. Najväčšia elegancia v technickom písaní vyžaduje jednoduchosť a zrozumiteľnosť." [1]

Ako som už spomenul, najčastejší spôsob čítania používateľskej príručky je vyhľadanie potrebnej kapitoly v prípade, že zákazník „má problém“ – aplikácia sa nespráva tak, ako by sa mala. Preto je potrebné, aby sa čitateľ pri otvorení dokumentácie na náhodnom mieste necítil stratený a nepotreboval ju čítať od začiatku až do konca. S tým súvisí nielen jej vhodná štruktúra (čitateľ sa vie zorientovať, kde v texte sa práve nachádza a kde má hľadať požadovanú informáciu), ale aj akási vecnosť textu (napr. pri hľadaní určitého postupu krokov nájde jednoduchý postup v bodoch, nie rozsiahly slovný opis).

Nejasná dokumentácia však nemusí ovplyvniť iba zákazníka. Udržiavanie takejto dokumentácie v prípade zmien v produkte je drahé a časovo náročné (viď časť Nekonzistentnosť).

Neaktuálnosť

Pre vyvíjaný program je ohlásený termín jeho vydania. Ten sa nezadržateľne blíži. Upozorňujú na to aj masívne reklamy v najrôznejších médiách. Ešte posledné testovania a môže sa uzatvoriť prvá kapitola vzniku nového softvérového produktu. No v poslednej chvíli sa zisťuje zásadná chyba. Čo teraz? Odložiť jeho vydanie? Nejde o tak závažnú chybu, aby sa nestihla opraviť. No zostáva čas aj na revidovanie dokumentácie? Mnohokrát nie. Aj to môže byť jedným z mnohých dôvodov, prečo je dokumentácia k produktu neaktuálna.

Neaktuálnosť dokumentácie môže viesť v konečnom dôsledku k zmätenosti používateľa. Ten v prípade nejasnosti otvorí dokumentáciu a nájde príslušný postup, ktorý opisuje postupnosť potrebných krokov. Ak však natrafí na prípad použitia programu, kedy postupuje podľa príručky, avšak produkt sa správa inak, ako je v nej deklarované, môže si mylne pomyslieť, že ide nekvalitný softvér. No nemusí tomu tak byť. Nekvalitnou môže byť práve dokumentácia z dôvodu jej neaktuálnosti.

Podľa [2] existujú aj okolnosti, pri ktorých je rozhodnutie ponechať dokumentáciu neaktuálnu opodstatnené. Niektoré typy dokumentácie sú napríklad potrebné iba v špecifickom čase vývoja softvéru a ich udržiavanie by neprinieslo žiadne výhody. Nazdávam sa však, že v tomto prípade ide zväčša o špecifické prípady dokumentácie (tzv. prípad *výnimka potvrdzuje pravidlo*), napr. interné dokumenty, prípadne dokumenty týkajúce sa prototypov na zahodenie, ktoré v určitom momente prestanú byť potrebné. Bežná používateľská príručka do tejto kategórie dokumentov však určite nespadá.

Nekonzistentnosť

Jednou z najdrahších a kľúčových činností pri tvorbe dokumentácie je udržiavanie jej konzistentnosti. Celková dokumentácia k softvérovému produktu často zahŕňa množstvo dokumentov rôzneho druhu určených pre rôzne použitie - technickú dokumentáciu, používateľskú príručku, zoznam najčastejšie kladených otázok a pod. V jednotlivých dokumentoch sa môže z rôznych dôvodov vyskytovať ten istý údaj niekoľkokrát na rôznych miestach, prípadne môžu byť jeho fragmenty rozptýlené v rámci jedného (alebo viacerých) dokumentov. V prípade, že v produkte nastane zmena, je potrebné aktualizovať všetky súvisiace dokumenty. Vzniká tak otázka, ako udržiavať dokumentáciu konzistentnú s vynaložením minimálneho úsilia.

Riešenie tohto všeobecného problému sa skladá z viacerých parciálnych prístupov k tvorbe dokumentácie, napr. maximalizácia separácie relevantných informácií, prepojenie súvisiacich informácií prostredníctvom explicitných odkazov, využitie konceptu jedného zdroja, kedy sa fragmenty súvisiacich informácií nachádzajú na jednom mieste (napr. zdrojový kód + dokumentácia API), tzv. pohľad - virtuálny dokument, ktorý kombinuje obsah z viacerých rôznych dokumentov a pod. [2].

Problém s nekonzistentnosťou dokumentácie môže byť tiež spôsobený nekoordinovanou spolupracou viacerých ľudí pri jej tvorbe. Kým jeden človek (najčastejšie programátor) sa stará o dokumentáciu ku zdrojovému kódu, ďalší môže písať príručku používateľa. Ak však programátor zmení časť programu a upraví príslušnú technickú dokumentáciu, je zaručené, že na túto zmenu zareaguje aj tvorca príručky?

Existujú dva hlavné spôsoby zachovania celkovej aktuálnosti dokumentácie: synchronný (všetky potrebné zmeny sa vykonajú okamžite) a asynchronný (vytvorí sa požiadavka na zmenu dokumentácie, ktorá sa spracuje neskôr) [2]. Hoci za výhodu prvého spôsobu riešenia stanoveného problému považujem zaručenie konzistentnosti dokumentácie v každom momente, jeho realizácia pri väčších tímoch je viac ako problémová. Nevieť si predstaviť spôsob, akým by sa všetci tí, ktorí majú na starosti obsah dokumentácie, dohodli na zmene a v istom momente by ju aktualizovali (pričom by, samozrejme, museli na čas odložiť svoju aktuálnu prácu). Využitie tohto spôsobu zachovania konzistencie preto vidím len tam, kde má tvorbu celej dokumentácie na starosti len jeden až dvaja ľudia.

Vo väčších tímoch je preto výhodnejšie strpieť dočasnú nekonzistenciu dokumentácie s tým, že všetky požiadavky na potrebné úpravy sú v istej podobe zaznamenané a čakajú na svoje vyriešenie. Tým, že sa takéto požiadavky zaznamenávajú, sa naskytá možnosť jednoduchšieho monitorovania aktuálnosti dokumentácie (vidíme, koľko požiadaviek na zmenu čaká na vybavenie, ktorých častí sa týkajú a pod.).

Vhodné využitie týchto prístupov môže viesť ku konzistentnej dokumentácii, čím sa ušetrí čas jej tvorcov potrebný na jej aktualizáciu a udržiavanie a tým aj nemalé finančné prostriedky.

Záver

Výhody kvalitnej dokumentácie k softvérovému produktu vidí asi každý, kto už prišiel do styku s rozsiahlejším programom – či už z pohľadu zákazníka, programátora alebo technickej podpory zákazníka. Napriek tomu sa jej tvorbe stále nevenuje dostatok úsilia, takže výsledkom sú dokumenty, v ktorých sa čitateľ cíti stratený, neporozumie problematike alebo nenájde, čo hľadá.

Hoci je ťažké nájsť konkrétne merateľné kritériá na určenie kvality dokumentácie, intuitívne identifikovanie kvalitnej dokumentácie už takým problémom nie je. V tejto eseji som sa venoval niektorým základným chybám, ktorým by sa mal vyhýbať každý tvorca dokumentácie. Na prvý pohľad by sa mohlo zdať, že dodržiavaním týchto zásad bude jednoduché a privedie nás ku kvalite. Podrobnejším skúmaním jednotlivých bodov ako celku však môžeme dospieť k záveru, že mnohé z nich si navzájom odporujú. Na odstránenie problému nejasnosti dokumentácie je potrebná istá redundancia dát. V protiklade však stojí zásada maximálneho možného štruktúrovania príbuzných informácií pre potreby jednoduchého udržiavania konzistencie. Pri využívaní systému na zaznamenávanie požiadaviek na zmenu častí dokumentácie zápasíme s problémom neaktuálnosti. Z týchto a mnohých ďalších konfliktov je preto zrejmá potreba nájsť akýsi kompromis. Zvoliť si hranicu, do akej strpíme prítomnosť istého problému za cenu vyriešenia iného (alebo skupiny iných). Tieto hranice sú však silne závislé od mnohých faktorov, ako je veľkosť tímu, spôsob jeho riadenia a organizácie, typ softvérového produktu, cieľová skupina zákazníkov a pod. A práve stanovenie týchto hraníc má na starosti člen tímu zastávajúci úlohu manažéra dokumentácie.

V neposlednom rade je tiež potrebná disciplína pisateľov, aby stanovený koncept rešpektovali a dodržiavali. Je totiž zbytočný dokonalý systém tvorby dokumentácie, keď zlyhá ľudský faktor. Tvorba a udržiavanie kvalitnej dokumentácie nie je jednoduché. No

pri troche úsilia nám nakoniec bude odmenou kvalitná dokumentácia, spokojní zákazníci, ušetrený čas pracovníkov technickej podpory a tiež zníženie vynaložených finančných prostriedkov.

Použitá literatúra

1. Rivera, T., Tate, A., Will, S.: Deadly Sins of Technical Documentation. *IEEE SoutheastCon Proceedings*, 2004, pp. 297-301.
2. Correia, F., Aguiar, A., Ferreira, H., Flores, N.: Patterns for Consistent Software Documentation. *Proceedings of the 16th Conference on Pattern Languages of Programs*, ACM, 2012.
3. Pierce, R.: Optimizing Your Documentation with the Help of Technical Support. *Proceedings of the 21st annual international conference on Documentation*, ACM, 2003, pp. 6-11.

Annotation

Major sins of documentation writers

Technical documentation, user's guide, source code comments. These documents are considered an essential part of a software product. However, writers often spent much less energy to create them than they deserve. Although good documentation implies more satisfied customers, saved significant time and money, it is still considered as a necessary evil more than as a way to simplify own work and to improve the quality of the final product. In this essay I focus on the need for documenting software from multiple perspectives, I think about the concept of quality in the context of documenting and I describe the basic mistakes made by the documentation writers.