

AJ NEAKTUÁLNA DOKUMENTÁCIA JE UŽITOČNÁ

Modli sa, pracuj a hlavne dokumentuj.

Filip Ogurčák

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
filip.ogurcak@gmail.com

Abstrakt. *Technická dokumentácia tvorí podstatnú, no častokrát zanedbávanú časť výsledného softvérového produktu. Zanedbávanú v tom zmysle, že pozornosť je kladená prevažne na funkcionality výsledného produktu, a na pravidelné aktualizovanie a dopĺňanie dokumentácie sa často zabúda. Dokumentácia však neslúži iba zákazníkom, ale je dôležitá aj pre programátorov, vedúcich vývojového tímu a manažment organizácie. V tejto eseji by som sa preto rád zamerlal na fakt, že aj keď dokumentácia nie je práve najaktuálnejšia, a nie sú v nej zapracované najnovšie zmeny, stále má dostatočnú prídavnú hodnotu pre zákazníka, programátora a manažéra kontrolujúceho nasledovanie projektu. Podstatný je v tomto prípade pohľad na vyvíjaný produkt ako celok, ale aj spôsob prepojenia jednotlivých modulov a znázornenie celkovej funkcionality. V konečnom dôsledku sa však do popredia dostáva neprítomnosť určitého štandardu, ktorý by proces tvorby technickej dokumentácie zjednotil a zjednodušil.*

Kľúčové slová: *dokumentovanie, aktuálnosť, programátor, zákazník, manuál, príručka, informačný systém, manažment dokumentovania*

Načo je dokumentácia a komu je určená?

Dokumentácia je vlastne určitý typ manuálu, ktorý popisuje všetky vlastnosti vytvoreného produktu. Zákazníkovi je dodávaná spoločne s týmto produktom, a preto je jej prítomnosť veľmi podstatná, ba až nevyhnutná.

Komu je však dokumentácia určená? V prvom rade to je zákazník, ktorý z viacerých druhov dokumentácií využíva najmä používateľskú príručku. Často je však dobré, ak má k dispozícii aj technickú dokumentáciu, vďaka ktorej vie sledovať napredovanie projektu a priebežne upravovať svoje požiadavky. Za rovnakým účelom využívajú technickú dokumentáciu aj vedúci vývojových tímov, vďaka čomu majú prehľad o práci na projekte a dodržiavaní plánu.

V poslednom rade je dokumentácia určená samotným vývojárom, ktorí si po jej prečítaní dokážu odpovedať na väčšinu svojich otázok [2] a získajú prehľad o produkte ako celku, ale aj o architektúre a jednotlivých častiach tohto produktu.

Obsah dokumentácie

Dokumentácia je prínosom pre menší projekt, a nevyhnutnou súčasťou pre projekt veľký. Jej písanie a dopĺňanie by malo prebiehať počas všetkých fáz vývoja produktu. Z vlastnej skúsenosti však môžem povedať, že dokumentácia je považovaná iba za nutné zlo, a preto sa väčšinou píše tesne pred termínom odovzdania, alebo až po dokončení produktu.

Čo však musí dobrá dokumentácia obsahovať? A ako vieme, že dokumentácia je dobrá? Pre začiatok stačí, ak si po jej prečítaní človek dokáže odpovedať na všetky otázky aj bez potreby konzultácie s autorom [2]. To však ešte stále neznamená, že dokumentácia je dobrá. Kvalitu dokumentácie spoznáte až po jej prečítaní [4]. Pritom však neexistuje žiadne konkrétne pravidlo, podľa ktorého sa dokumentácia delí na dobrú a zlú. Toto delenie je čisto subjektívne a závisí od každého čitateľa. Existujú však určité pravidlá, ktorých dodržaním sa vieme ku kvalitnej dokumentácii dopracovať. Najdôležitejšie je však uvedomiť si, že dokumentáciu budú čítať aj ľudia, ktorí do danej oblasti nevidia dostatočne [4]. To je obzvlášť dôležité v prípade používateľského manuálu.

Predstavte si, že vy ako vývojár, by ste si prečítali odborný manuál určený chirurgovi. Tiež by ste sa z lekárskej terminológie nič nedozvedeli.

Dôležité je preto používať úplné a jasné definície, a všetky informácie udržiavať správne a aktuálne [4]. Zastavím sa však pri poslednej zo spomínaných vlastností: „informácie udržiavať aktuálne.“ Táto vlastnosť sa ukazuje ako kameň úrazu pri udržiavaní dokumentácie. V tejto eseji by som preto rád tento problém rozobral a poukázal na fakt, že to nie je až taký veľký problém, aký sa na prvý pohľad javí. Všetko má však svoje „ale?“. Poďme však pekne po poriadku.

Ako často teda dokumentáciu aktualizovať?

Zamerajme sa na väčšie projekty, na ktorých pracuje viacero programátorov a sú manažované určitým druhom manažmentu. V takomto prípade je logické, že dokumentácia sa píše priebežne v každej etape vývoja produktu a jej obsah má byť aktualizovaný po každej zmene. Tento spôsob práce je opodstatnený z toho dôvodu, aby si všetci zainteresovaní z dokumentácie vedeli získavať pre seba dôležité informácie.

Z vlastnej skúsenosti však môžem povedať, že aj keď sa dokumentácia píše priebežne, aktualizácia jednotlivých údajov prebieha skôr sporadicky, a teda väčšinu času je dokumentácia neaktuálna. Myslím si, že za túto skutočnosť môžu z veľkej časti vývojári, ktorí sú zaťažení prácou na produkte a dokumentáciu posúvajú nabok. Na druhej strane

však za tým môže byť aj obyčajná lenivosť. Je totiž jednoduchšie a hlavne rýchlejšie počkať, kým nebude zmien viacero, ako po každej otvárať dokument, vyhľadať potrebné kapitoly, a nakoniec túto zmenu zakomponovať.

Môžeme sa pozrieť na výsledky štúdie [1], ktorá zistila, že text dokumentácie priamo spojený so zdrojovým kódom (teda akceptačné testy a z časti aj dokumentácia kvality) je aktualizovaný častejšie ako ten zvyšný. Dá sa síce predpokladať, že celkový návrh a architektúra systému sa z finančných a časových dôvodov meniť nebudú. Ale prečo sa tak často aktualizujú akceptačné testy? Z môjho pohľadu, teda toho vývojárskeho, za to môže uvedomovanie si ďalekosiahlych následkov pri neaktualizovaní tejto časti dokumentácie. Tým mám namysli, že produkt, ktorý neprešiel akceptačnými testami, je zlý produkt. To ma za následok hľadanie vinníka a následné vyvodzovanie a znášanie následkov.

Môžeme sa však na to pozrieť aj z pohľadu návrhára, ktorý produkt priebežne upravuje požiadavkám zákazníka a možnostiam programátorov. V takom prípade sa aj pri minimálnych zmenách určitým spôsobom menia vlastnosti produktu. Tieto zmeny je potrebné do dokumentácie zakomponovať, a taktiež upraviť výsledky jednotlivých akceptačných testov. V konečnom dôsledku má však najväčšiu zásluhu na aktuálnosti akceptačných testov práve ich jednoduchá štruktúra [1], ktorá má presne definované pravidlá tvorby, a vecný a prehľadný obsah. Práve preto sa ako najvhodnejšia javí dokumentácia s jednoduchou štruktúrou, a vecným a prehľadným obsahom.

Čo s neaktuálnou dokumentáciou?

Pozrime sa opäť na výsledky prieskumu [1]. Z neho vyplýva, že až 81% inžinierov si myslí, že aj neaktuálna dokumentácia môže byť užitočná. Poďme si to teda rozanalyzovať.

Je logické, že dokumentácia sa číta, aby sme sa dozvedeli niečo nové. Väčšinou nám však stačia iba základné informácie o produkte [1], ktoré postačujú na objasnenie základnej funkcionality. Pri získavaní týchto „základných“ vedomostí pritom nie sú podstatné detaily a presné hodnoty, ale podľa mňa je dôležitejšia architektúra a samotné správanie sa produktu. Táto časť dokumentácie je pritom v podstate aktuálna, nakoľko sa v priebehu vývoja produktu nemení, alebo sa mení iba minimálne.

Z výsledkov prieskumu [1] je viditeľné aj to, že iba tretina používateľov využíva dokumentáciu na hľadanie odpovedí na otázky, alebo na hľadanie hlbšej podstaty fungovania systému. Týmto používateľom už používateľská príručka nemusí stačiť nakoľko presné a korektné informácie sú pre nich kľúčové. Pre takýchto používateľov bude neaktuálna dokumentácia zbytočná a zavádzajúca. Na druhej strane sa však jedná hlavne o skúsených koncových používateľov, a keďže dokumentácia sa pred odovzdaním produktu dopĺňa a aktualizuje, k týmto používateľom sa dostane už len finálna, a teda aktuálna verzia dokumentácie.

Čo sa týka samotných vývojárov, tí využívajú prevažne technickú dokumentáciu, a to na testovanie funkcionality produktu. Pre toto testovanie sú podstatné hlavne akceptačné testy, ktorých aktuálnosť sme si vysvetlili skôr.

Z predošlých riadkov by sa mohlo zdať, že dokumentáciu aktualizovať ani netreba. To je však omyl. Čo zákazník, ktorý sa podieľa na tvorbe produktu? A čo manažér kontrolujúci pokrok vývoja? Tí využívajú dokumentáciu na získavanie informácií

o aktuálnom stave produktu, a jej aktuálnosť je pre nich kľúčová. Z informácií získaných z dokumentácie vedia totiž manažéri správne plánovať ďalšie etapy projektu, a zákazníci upravovať požiadavky na výsledný produkt. Ako teda motivovať vývojárov k častejšiemu písaniu dokumentácie a pravidelnému aktualizovaniu údajov?

Riešením môže byť použitie štandardu

Najväčším problémom pri tvorbe technickej dokumentácie je neprítomnosť určitého štandardu, ktorý by tento proces zjednotil a zjednodušil [3].

Existujú síce určité štandardy, tie sú však určené iba pre špecializovaný a výskumný softvér, a navyše ani tie nemajú presne definované pravidlá svojej tvorby. Je to iba určitý návod, ktorý môže pomôcť pri tvorbe väčšiny softvérových projektov. Tento návod je navyše tak všeobecný, že je užitočný aj pre vývojárov, ktorí majú svoju vlastnú metódu písania dokumentácie.

Technická dokumentácia sa podľa tohto návodu skladá zo 4 častí [3]:

1. Abstrakt
2. Všeobecné informácie o produkte (popis funkcionality, rozsah vstupných hodnôt)
3. Definícia problému (podrobný popis matematických vzorov, modelov, limity produktu, spôsob testovania, ...)
4. Programátorské informácie (popis premenných, dát, platformy, ...)

Už na prvý pohľad je viditeľné, že jednotlivé časti dokumentácie sú zadefinované veľmi všeobecne a nijakým spôsobom neurčujú svoj obsah a rozsah. Ak by sme však zadefinovali presnú štruktúru a obsah jednotlivých častí technickej dokumentácie, mohli by sme určité časti opakovane využívať aj pri ďalších produktoch, a tým uľahčovať a urýchľovať prácu vývojárom. Navyše, vývojári sú pri tvorbe produktov zvyknutí na opätovné využívanie modulov, a myslím si, že rovnaký spôsob tvorby by uvítali aj pri tvorbe dokumentácie. Ak by sme navyše využili jednoduchú a prehľadnú štruktúru založenú na tabuľkách, možno by sme dosiahli rovnaký účinok na aktuálnosť dokumentácie ako majú akceptačné testy.

V konečnom dôsledku by sa nám možno podarilo dosiahnuť častejšie aktualizovanie dokumentácie v rámci organizácie. Aby sme to však dosiahli, vývojári by museli byť k takémuto spôsobu tvorby dostatočne motivovaní, a samotná kontrola dokumentácie by si vyžadovala viac úsilia ako doteraz.

Myslím si však, že týmto problémom sa zaoberá väčšina organizácií, a teda riešenie by malo byť globálne. Preto navrhujem tvorbu štandardu, ktorý by mal presne definovaný obsah a rozsah, a na jeho dodržiavanie by sa podieľala celá organizácia. Tento krok by navyše zjednotil formu všetkých dokumentácií, a teda zákazníci a všetci zainteresovaní by sa v texte dokumentácie zorientovali rýchlejšie, a rýchlejšie by v ňom vedeli vyhľadávať pre seba dôležité informácie.

Zhrnutie

V rámci tejto eseje sme načrtli problém písania kvalitnej dokumentácie, pričom sme sa bližšie zamerali na problematiku neaktuálnosti dokumentácie. Znázornili sme viacero

uhlov pohľadu na tento problém, pričom sme poukázali aj na fakt, že neaktuálna dokumentácia nemusí byť taký veľký problém, akým sa môže spočiatku javiť.

V závere eseje sme sa zamerali na hľadanie riešenia a zadefinovanie určitých postupov, ktoré by mohli riešiť viacero problémov spojených s dokumentovaním ako takým.

V konečnom dôsledku sa však ukazuje ako najväčšia nevýhoda práve neprítomnosť určitého štandardu, ktorý by technické dokumentácie zjednotil a prispel k ich častejšiemu dopĺňaniu a aktualizovaniu.

Použitá literatúra

1. Lethbridge, T., Singer, J., Forward, A.: How Software Engineers Use Documentation: The State of the Practice. In: *IEEE Software*, vol. 20 issue 6, 2003, pp. 35-39.
2. Simmons, D.: The Art Of Writing large Programs. In: *Computer*, vol. 5 issue 2, 1972, pp. 47-48.
3. Phoha, V.: A Standard for Software Documentation. In: *Computer*, vol. 30 issue 10, 1997, pp. 97- 98.
4. Rivera, T., Tate, A., Will, S.: Deadly Sins of Technical Documentation. In: IEEE SoutheastCon, 2004. Proceedings, 2004, pp. 297-301.

Annotation

Out-of-date documentation is useful, too

Technical documentation is an essential but often ignored part of the final software product. Ignored in the sense, that attention is given mainly to the functionality of the final product and for regular updates of documentation no time remains. Documentation, however, is not only for final customer, but is also important for programmers, development team leaders and company management. In this essay I would like to focus on the fact, that out-of-date documentation has still enough additional value for customer, programmer and manager controlling the progress of the project. In this case is important developed product as a whole, connection between modules and representation of general functionality. Finally is showed an absence of a standard, which would make technical documentation unified and simplified.