

DOKUMENTOVAŤ?! JA MÁM INÚ PRÁCU

Atrament je lepší ako tá najlepšia pamäť

(čínske príslovie)

Peter Jurčík

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
peter.jurcik@cgb.sk

Abstrakt. *Technická dokumentácia tvorí neoddeliteľnú súčasť dodávaného diela. Napriek tomu sa jej prikladá v porovnaní s vývojom softvéru iba nepatrná dôležitosť. Dôvodov prečo sa tak stáva je viacero. Vytvorená kvalitná dokumentácia nemá formu iba určitého manuálu pre používateľov. Môže tvoriť dôležitý zdroj informácií pre programátorov ale aj pre manažérov vývojového tímu. Absolútnym minimom ako môžu programátori prispieť k procesu tvorby dokumentácie je písanie výstižných komentárov, z ktorých sa dá využitím nástrojov na automatické generovanie komentárov zo zdrojového kódu, získať aspoň základ technickej dokumentácie produktu. Samotné komentáre pritom pomáhajú aj programátorom sa lepšie orientovať vo vytváranom kóde. Napriek faktu, že je veľmi ťažké objektívne merať kvalitu dokumentácie, stanovíme základné pravidlá, ktoré treba dodržiavať pri tvorbe kvalitnej technickej dokumentácie.*

Kľúčové slová: *technická dokumentácia, manažér a dokumentácia, programátor a dokumentácia, prečo dokumentovať, výhody kvalitnej dokumentácie*

Úvod

„Tvorba technickej dokumentácie sa v tímoch zaoberajúcich sa vývojom softvéru veľmi často podceňuje a má charakter iba určitého spätne napísaného manuálu.“[2] Ten slúži prevažne iba ako návod pre koncového používateľa, pre ktorého bol produkt vyvinutý.

Má vôbec zmysel tvoriť kompletnú technickú dokumentáciu aj v prípadoch, keď to nie je explicitne špecifikované v dodávateľských podmienkach? Je to úloha, ktorá nejakým spôsobom priamo ovplyvňuje proces tvorby softvéru alebo len práca navyše? Pozrime sa na problém spoločne očami zainteresovaných protagonistov z vývojového tímu a zaujmime svoj vlastný postoj k problematike.

Prečo sa tvorba technickej dokumentácie podceňuje

Tvorba softvéru ako hlavného predmetu činnosti organizácie, je vlastne výrobná aktivita, ktorej hlavným cieľom je výsledný produkt. Ako uvádza aj [2] funkčnosť programu je prioritou pre všetky zúčastnené strany. Zákazníkovým prvoradým cieľom je funkcionálnosť programu a k nemu dodaná dokumentácia predstavuje len určitú prílohu k realizovanému dielu. Ak je cena za dodané dielo príliš nízka môže sa stať, že už jednoducho neostane rozpočet na ďalšie práce a dodaná dokumentácia sa obmedzí iba na určitý návod na použitie.

„Dokumentácia dôkladne popisujúca vytvorené dielo, však rozhodne nie je len nejaký manuál na použitie, ale predstavuje východiskový bod pre akúkoľvek zmenu vykonávanú v programe.“[3] Všetky zúčastnené strany ocenia jej existenciu a kvalitu najmä z dlhodobého pohľadu.

Kvalitná dokumentácia vytváraného diela je v organizáciách požiadavka smerujúca k programátorom z manažérskych pozícií. Ak je však programátor dlhodobo neúmerne preťažovaný riešením funkčných požiadaviek programu, ktoré sú samozrejme prioritou, prudko klesá jeho chuť spraviť toľko všade spomínaný „krok navyše“.

Pohľad očami vedúceho projektu

Vedúci projektu je prioritne zodpovedný za kvalitu a termíny odovzdania objednaného diela. Jeho prioritnou úlohou je teda efektívne manažovanie ostatných členov v tíme. Na problémy nazerá skôr vo všeobecnejšom rovine a preto ako spôsob udržiavania kontaktu s procesom vývoja vyžaduje pravidelné reporty a technickú dokumentáciu. Vzhľadom na fakt, že ide o vedúceho pracovníka v tíme, uvedomuje si svoje postavenie a chce si ho prinajmenšom z osobných dôvodov udržať.

„Kvalitná dokumentácia mu poskytuje cenný zdroj informácií o jednotlivých vykonaných krokoch v procese realizácie projektu od tej najnižšej úrovne.“[2] Rovnako tak aj jej udržiavanie zmierni závislosť organizácie od kľúčových vývojových pracovníkov, ktorí tvoria potrebné zložité konštrukcie v zdrojovom kóde programu. V prípade odchodu týchto pracovníkov, môžu noví vývojári omnoho jednoduchšie a rýchlejšie nadviazať na už vykonanú prácu. V prípade absencie akejkoľvek dokumentácie, by vlastne museli prechádzať vytvorený zdrojový kód riadok po riadku, čo v prípade zložitých operácií nie je jednoduchou úlohou ani pre skúsených senior vývojárov.

Minimálne z týchto uvedených dôvodov by mal vedúci projektu trvať na tvorbe kvalitnej dokumentácie všade tam, kde to má opodstatnenie. Pri tvorbe jednoduchšej statickej web stránky s dvojúrovňovým menu si určite vystačíme aj s manuálom na použitie určeného pre koncového zákazníka. Avšak pri tvorbe napríklad zložitejšieho účtovníckeho modulu, si bez udržiavania podrobnejšej kvalitnej dokumentácie, organizácia

samotná komplikuje prácu. Vďaka častým legislatívnym úpravám sú totiž zmeny v takomto type softvéru veľmi časté. V prípade absencie zdokumentovaných procesov a všetkých vzťahov voči ostatným softvérovým súčastiam produktu, sú následné zmeny zbytočne časovo náročné.

Pohľad očami vývojára

Vývojár alebo ak chceme programátor vo svojej práci prioritne rieši funkčnosť vytváraného produktu. V procese tvorby softvérového riešenie je to práca na tej najnižšej úrovni tvorby. Z toho plynie, že programátor musí poznať činnosť produktu a jednotlivých častí do tých najmenších detailov. Všetky súčasti funkčných náležitosti programu sú vytvárané práve programátormi.

„Dokumentovanie zdrojového kódu programátorom použitím vecných a výstižných komentárov v kóde, však môže slúžiť práve aj samotnému programátorovi.“[3] Veď si každý odpovedzme za seba. Koľkokrát sa nám podarilo úspešne zapísať zložitejší algoritmus, premysleli sme jeho každý detail do najmenších maličkostí a plný radosti sme neokomentovali ani čiarku z jeho logiky fungovania? A ak sme sa následne po dlhšom období museli k tomuto algoritmu vracáť, len sme sa škrabali po hlave a rozmýšľali, čo sa to tam vlastne odohráva. A pritom išlo o náš vlastný kód! Stačilo pritom napísať iba pár riadkov textu popisujúcich to, čo sa nám aj tak pri tvorbe kódu dookola premietalo hlavou. Pár riadkov a deň by bol naozaj jednoduchší, kratší. Nevieť ako vám, ale mne to za tých pár riadkov textu naozaj stojí a snažím sa v záujme zjednodušenia si vlastnej práce komentovať jednotlivé pasáže zdrojového kódu. A radšej viac ako keby malo chybať. Keď treba následne revidovať zdrojový kód, vždy príde výstižný komentár vhod a zabráni tomu, aby sme objavovali niečo, čo sme už dokonca predtým sami vynašli.

Ako by teda kvalitná dokumentácia mala vyzeráť

„Je veľmi ťažké určiť presné pravidlá, ako by mala kvalitná technická dokumentácia vyzeráť. Ak by sme napríklad chceli zvoliť nejaké metriky, ktoré metriky by zabezpečili, že predkladaný dokument má vysokú kvalitu?“[3] Namiesto zbytočnej diskusie o tom, ako merať efektívne použitie ľudského jazyka pri tvorbe dokumentácie, si uvedme niekoľko odporúčaní, ktorých je potrebné sa pridržiavať pri tvorbe kvalitnej technickej dokumentácie, tak ako ich uvádzajú autori v [3]:

1. Záleží na faktoch: Na prvý pohľad samozrejماً vec je však z pohľadu tvorby dokumentácie absolútne kľúčová. Dokumentácia musí byť v prvom rade absolútne vecne správna.
2. Záleží na detailoch: Aj drobný preklep môže spôsobiť dezorientáciu čitateľa.
3. Informácia musí byť úplná: Táto chyba sa vyskytuje veľmi často v prípadoch, keď autor nepozná dostatočne tému, o ktorej píše.
4. Informácia musí byť jasná: Zbytočne sa nesnažme informácie „zaobaľovať“. Jasný a zrozumiteľný štýl technického písania je najlepšій.
5. Informácia musí byť stručná: Nesnažme sa zbytočne vysvetľovať vetami to, čo ide vysvetliť pár slovami.

4 Peter Jurčík

6. Kontext dokumentácie: Dokumentácia by mala byť napísaná štýlom, aby ju bolo možné pochopiť aj bez kontextu na predchádzajúce časti. Čitateľ totiž nemusí mať záujem čítať dokumentáciu ako celok, ale môže sa zaujímať len o jej jednotlivé časti.

Pochopenie a správne praktizovanie týchto pravidiel by nás malo doviesť k cieľu, ktorým je kvalitná technická dokumentácia vytváraného diela. „Rovnako tak ako totiž dokáže dobrá dokumentácia pomôcť a poradiť, môže zle vypracovaná dokumentácia pomýliť a v konečnom dôsledku aj ublížiť.“[2]

Záver

To že písať dokumentáciu je potrebné vedieť alebo teda aspoň počuli už asi všetci. Argumenty jasne hovoria o potrebe dokumentácie rovnako tak pre riadiacich ako aj pre výkonných členov tímu. A že tvorba dokumentácia zdržuje od vývojárskej práce, ktorá je na projekte prvoradá? Opak je pravdou. Manažérom pomáha sa rýchlejšie zorientovať v procese vývoja. Programátorom zase pomáha sa rýchlejšie orientovať vo vytvorenom kóde. „Získavanie aspoň základu dokumentácie pomocou automatického generovania obsahu z komentárov v zdrojovom kóde programu je činnosť, ktorá predstavuje iba pár klikov myšou a o ostatné sa už postará zvolený nástroj.“[1] Organizácia vďaka tomu bude disponovať základom technickej dokumentácie produktu, ktorý bol vytvorený s minimom vynaloženého úsilia.

Použitá literatúra

1. Albing, Bill.: Combining Human-Authored and Machine-Generated Software Product Documentation. *Professional Communication Conference 2003*, 6 pp.
2. Drechsler, Rolf; Muranko, Beate.: Technical Documentation of Software and Hardware in Embedded Systems. *Very Large Scale Integration 2006*, 261-266.
3. Rivera, T., A. Tate, and S. Will. Deadly Sins of Technical Documentation. *IEEE SoutheastCon, 2004. Proceedings. 2004*, 297-301.

Annotation

Document?! I've something else to do

Technical documentation is a substantial component of the overall product. Even though in comparison to software development it is regarded as of little importance. There are more reasons why this happens. Quality documentation is much more than just a user manual. It can provide valuable information for programmers and also for managers of development team. Writing of clear comments in source code is absolute minimum how can programmers contribute to the process of creating documentation. From these comments can be extracted the draft of the documentation using the tools for automatic source code documentation. These very comments are also helpful for programmers to better orientate in the created source code. Even the fact it is very difficult to

measure the quality of the technical documentation objectively, we name the basic rules for producing high-quality documentation.