

A čo s tou kvalitou?

LUKÁŠ SIM

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
lukas.sim@gmail.com*

Abstrakt. Spolu so vzrastajúcou dôležitosťou a komplexnosťou softvérových produktov sa stále viac skloňuje slovné spojenie „kvalita softvéru“. Aj napriek tomu, že je tento pojem bežne používaný, len málo ľudí si dokáže predstaviť, čo sa za týmto abstraktným pomenovaním skrýva. Úlohou tejto eseje je predstaviť rôzne spôsoby, akými je kvalita softvéru v súčasnosti vnímaná. Taktiež budú popísané výhody a nevýhody jednotlivých definícií tohto pojmu z hľadiska tvorcu softvéru ako aj jeho používateľa. V dokumente bude takisto venovaný priestor pre popísanie možností pre meranie kvality softvéru, a popísané silné a slabé stránky jednotlivých prístupov.

Prečo sa tým vôbec zaoberať?

Bolo to len pár desiatok rokov dozadu, čo takmer nikomu nenapadlo uvažovať nad tým, že by ľudia pri svojej každodennej práci mohli požívať počítače. No našťastie, alebo bohužiaľ, časy a aj myslenie ľudí sa menia, a v dnešnej dobe si už nikto, až na pár výnimiek nevie predstaviť, že by ľudia počítače pri práci nepoužívali. S postupným a stále bežnejším používaním počítačov však vznikla aj potreba vzniku softvéru pre tieto počítače. Spočiatku sa nejednalo z dnešného pohľadu o žiadne zázraky. Spočítať pár čísel, napísať nejaký text alebo nebudaj niečo vypísať na obrazovku boli jedny z prvých úloh ktoré bolo potrebné vyriešiť.

V tých dobách ešte nikomu nenapadlo zaoberať sa otázkou kvality programov, alebo ak chceme kvalitou softvéru. Program bol dobrý a kvalitný ak sa dokázal spustiť, ukončiť, nezamrznuť a vykonať presne to čo sa od neho očakávalo. No ako sa roky nabaľovali jeden na druhý, programov bolo treba stále viac a viac, bolo treba aby boli komplexnejšie, riešili rozličné úlohy a dokonca boli používateľné aj ľuďmi, ktorých profesným oborom bolo niečo iné ako informatika.

A v niekedy v tomto období si ľudia začali uvedomovať, že s tou kvalitou programov to nie je také čierno-biele ako sa kedysi zdalo, a že povedať ktorý program je dobrý, nie je až také jednoduché, a že aj ten najlepšie vyzerajúci program poskytujúci najväčšie množstvo najlepších funkcií nemusí byť úplne kóšer, pokiaľ dvakrát za deň zamrzne, spadne a zoberie so sebou do prepادلiska medzidiskového

priestoru poldňovú neuloženú prácu.

Nie je kvalita ako kvalita.

Určiť ktorý softvér je do akej miery kvalitný, môže byť pomerne náročná otázka. Konieckoncov už samotný pojem „Kvalita softvéru“ je dostatočne abstraktný na to aby dovolil interpretáciu mnohými spôsobmi. Čo je softvér je pomerne zrejmé, no čo je kvalita je už otázka náročnejšia. Odvolávajúc sa na múdre knihy napísané múdrymi ľuďmi uvedieme nasledujúce definície.

1. Miera v ktorej, množina základných charakteristík spĺňa požiadavky [1]
2. Zhoda s požiadavkami [2]
3. Stupeň výnimočnosti niečoho, meraný v porovnaní s inými podobnými vecami. [3]

Prvé dve z týchto definícií sa v podstate zhodujú. Tretia definícia naproti tomu hovorí o tom, že pokiaľ má byť niečo, produkt, služba alebo softvér, kvalitný musí byť výnimočný a v niečom lepší ako ostatné dostupné veci. Je len logické, že takáto definícia nemohla padnúť na úrodnú pôdu v komunite programátorov, ktorý sa snažia všetko robiť podľa možnosti čo najrýchlejšie a najjednoduchšie. A navyše, keby že sa celý svet pridrža tejto definície, ľudia by zrazu s prekvapením zistili, že väčšina toho čo používajú je jednoducho niečo nekvalitné. No a to by bola tvrdá rana pravdepodobne pre každého žijúceho v dnešnej spoločnosti. Aj keď si to väčšina ľudí nechce priznať.

Možno práve preto sa kvalita v súčasnosti meria podľa prvých dvoch definícií. Okrem ich nespornej pravdivosti ktorú všetci akosi podvedome cítia, dávajú možnosť vzniku omnoho väčšiemu množstvu kvalitných vecí ako oxfordská definícia. V tom tkvie ich sila, a to je ich veľkou prednosťou a výhodou. Všetkým predsa chce robiť kvalitné veci kvalitne, bez toho aby musel preukazovať zbytočnú výnimočnosť. Bez ohľadu na to či sa jedná o výrobu hrncov, tlač novín, alebo tvorbu informačného systému pre uchovávanie informácií o krvných skupinách dobrovoľných darcov krvi.

Ďalšou výhodou alebo nevýhodou je fakt, že poskytujú neuveriteľne široký priestor pre manipuláciu s pojmi kvalita. Ide o jednoduchý princíp spočívajúci v tom, že pokiaľ sú splnené zadané požiadavky, produkt je kvalitný. Za touto jednoduchou implikáciou sa toho skrýva viac ako sa zdá. Inými slovami aj úplne zle fungujúci program ktorý nedokáže správne spočítať dve čísla sa môže s čistým svedomím označiť za kvalitný pokiaľ zadávateľ omylom, schválne alebo z nepozornosti súhlasil práve s touto jeho vlastnosťou. Konieckoncov nevádi, že to nerobí to čo má, hlavne že je to podľa požiadaviek a je to kvalitné.

Hlavný problém ktorý z týchto definícií vyplýva je teda nasledovný: nie je kvalita ako kvalita. Pokiaľ niekto očakáva od filmu vzájomné kynoženie monštruóznych robotov a obyvateľov našej planéty, bude zjavne považovať za kvalitné

úplne iné filmy ako niekto s citlivou dušou pachtiaci po romantike.

No stačilo by hádam tohto trošku obšírnejšie úvodu, ktorý sa miestami vzdiali slovíčku „softvéru“ ale snažil sa čo najlepšie objasniť slovíčko kvalita. Je čas položiť si ďalšiu podstatnú otázku.

A čo to má spoločné so softvérom?

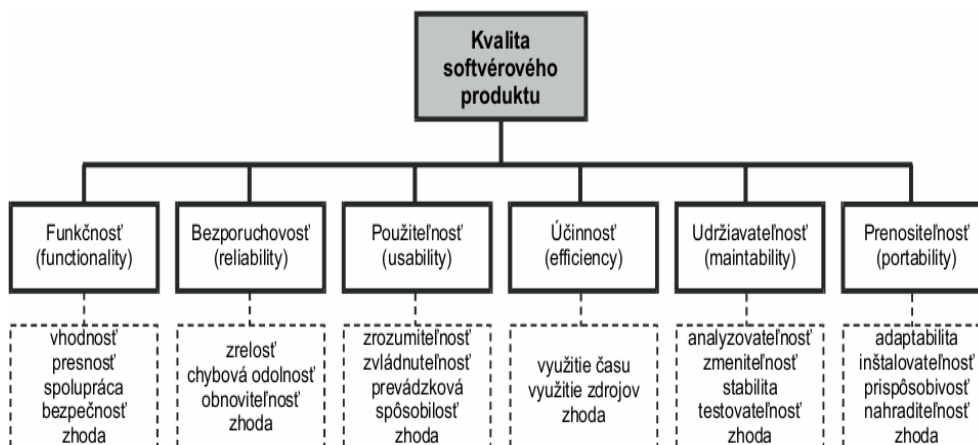
No v podstate všetko aj nič. Je jedno či vyrábate spomínané hrnce alebo programujete informačné systémy. Nakoniec aj tak najmete človeka, ktorému zaplatíte nehorázne peniaze za to, že vám vo firme rozvešia rôzne plagátky, rozdá červené kruhové samolepky a spraví prednášku o tom, aká je kvalita dôležitá, a čo by ste mali robiť pre to aby ste pracovali kvalitne. Výsledkom toho je, že získate certifikát, kvality ktorý si zavesíte do vstupnej haly firmy a na spodnú stranu hrncov alebo prednú stranu krabice s vaším programom pridáte žiarivú striebornú nálepku hlásajúcu, že tento produkt spĺňa požiadavky kvality podľa nory ISO-XXXX.

Samozrejme tu sa to celé komplikuje. Postup pri tvorbe softvéru a hrncov je, no povedzme, že pomerne odlišný. Hrnce sa väčšinou podarí spraviť podľa návrhu na prvýkrát, a ak aj nie, tak pokiaľ sú pokazené netreba ich chodiť za zákazníkom opravovať a ani mu posielat' kúsky kovu potiahnuté teflonom ako záplaty na diery ktoré vznikli používaním.

Práve preto je kvalita softvéru tak dôležitá. Jedno nesprávne rozhodnutie, jedna chyba a o nekvalitný softvér sa treba starať do konca života. Alebo minimálne pokiaľ je to napísané v zmluve.

Ako teda merať?

Pre potreby merania kvality softvéru vznikli určité metriky a charakteristiky pre kvalitu softvérového produktu. Jednou zo základných a najjednoduchšie pochopiteľných je hodnotenie kvality na základe charakteristík kvality. Jedná sa o takzvané statické posudzovanie kvality softvéru.



Obr.1. Charakteristiky kvality podľa ISO/IEC 9126

Kvalita celého produktu je daná tým do akej miery spĺňa jednotlivé charakteristiky. Je pomerne náročné určiť, ktoré z nich sú pre určenie celkovej kvality najdôležitejšie. No je pomerne jednoduché určiť, ktoré z týchto vlastností sú prioritné pre jednotlivé skupiny ľudí pracujúcich so softvérom.

Priority používateľa sú aby program robil to čo má, nepadal a hlavne aby sa s ním dobre pracovalo. Teda pre bežného používateľa nie je dôležité, či dokáže bez problémov doprogramovať do svojho textového editora modul pre spoluprácu s expertným systémom, ale či bude vedieť pohodlne a ľahko pridať do svojho dokumentu obrázky, bez toho aby mu program zamrzol a spadol. A chybová hláška ktorá sa po spadnutí ukáže možno pomôže programátorovi lokalizovať a odstrániť chybu, ale normálne používateľa len ešte viac vytočí do biela. Používajúc jazyk normy ISO/IEC 9126: použiteľnosť, funkčnosť a bezporuchovosť.

Na druhej strane, vývojár bude pri programovaní dbať na to a aby bol program čo najprehľadnejšie napísaný aby sa s jeho jednotlivými časťami dalo pekne jednoducho manipulovať, meniť ich a prispôbovať tomu čo práve potrebuje do programu pridať, odstrániť alebo zmeniť. Inak povedané udržateľnosť a prenositeľnosť.

Jediná položka ktorá zostala nepovšimnutá je účinnosť. Aj napriek tomu, že efektivita je veľmi dôležitá, podľa môjho názoru je až niekde na chvoste rebríčka dôležitosti týchto charakteristík. Programátor totiž ako tvor lenivý, samozrejme až na svetlé výnimky, radšej ožerie nejakú tu inštrukciu alebo príkaz naviac, ako by mal zmeniť časť, alebo nedajbože celý program. Samozrejme nehovoríme tu o bohapustom plytvaní strojového času, ale ruku na srdce, väčšina ľudí predsa len žije a aj programuje cestou menšieho odporu.

Používateľ tiež veľmi rád odpustí svojmu obľúbenému programu to, že štartuje o nejakú tú sekundu dlhšie, že ten ktorý výpočet nebeží tak rýchlo ako u iného programu, podstatné je, že tento program robí presne to čo robiť má, a že s ním je aj

napriek jeho muchám zžitý a vie s ním pracovať bez väčšej námahy.

No napriek tomu sa práve účinnosť stala povestnou čerešničkou na torte, ktorá oddeľuje dobrých od najlepších a najlepších od excelentných. Paradoxne charakteristika, nad ktorou väčšina ľudí privrie oko a mávne rukou je tým, podľa čoho si každý trochu rozumný používateľ nakoniec vyberie medzi dvoma softvérovými produktami ktoré poskytujú rovnakú funkčnosť, stabilitu a používateľskú prívetivosť. Vývojári budú zase na druhej strane celý pyšný na to, aký optimálny program sa im podarilo napísať a v dokumentácii k projektu nezabudnú priložiť sériu grafov na ktorých je jasne vidno že pri spracovaní údajov na bežnej vstupnej množine práve ten ich program pracuje presne o 0,7223 sekundy rýchlejšie, ako ostatné programy, ktoré navyiac zožerú o celé dva megabajty pamäte viac.

Okrem vyššie popísaného spôsobu sa dajú na meranie kvality softvéru použiť aj metriky s pomerne výstižným názvom.

Metriky kvality softvérového produktu?

Metriky pre hodnotenie kvality softvérového produktu tvoria spolu s metrikami pre hodnotenie procesov a metrikami pre hodnotenie projektu[4] komplexný pohľad na problematiku hodnotenia produktu. Ďalej bude venovaná z pochopiteľných dôvodov pozornosť predovšetkým metrikám pre hodnotenie kvality produktu. Pre získanie obrazu postačí keď povieme, že metriky hodnotenia procesov hodnotia veci ako

- Čas vývoja softvéru
- Čas potrebný na zistenie chyby
- Čas potrebný na odstránenie chyby

Metriky hodnotiace projekt sa zaoberajú hodnotením nasledujúcich veličín

- Čas
- Cena
- Ľudské zdroje
- Produktivita

No ale späť k metrikám hodnotiacim kvality. V podstate sa jedná o prístup, ktorý hodnotí kvalitu softvéru na základe nasledovných veličín[5].

- Priemerný čas do zlyhania
- Hustota chýb
- Zákaznícke problémy
- Zákaznícka spokojnosť

Priemerný čas do zlyhania predstavuje dobu, ktorú je priemerne systém schopný neustále prevádzky bez toho aby sa vyskytla chyba, čiže čas medzi dvoma chybami systému. Táto veličina je pomerne dôležitá hlavne pri systémoch, ktoré sú zodpovedné

za prevádzku vlakových systémov, zbraňových systémov, nemocníc ale aj výrobných liniek. V podstate sa všade tam kde je nutné nepretržité riadenie nejakej činnosti počítačom, alebo za pomoci počítača. Napríklad pravdepodobnosť výskytu chyby v systémoch riadiacich letovú prevádzku by mala byť menšia ako 0,0000000001 na hodinu.

Hustota chýb naopak predstavuje veličinu, ktorá je veľmi často využívaná v bežnej obchodnej sfére. Ide o počet výskytov chýb v programe, vzhľadom na počet riadkov kódu. Problém však nastáva už pri určení toho čo je jeden riadok kódu. Môže ísť o vykonávanie riadkov kódu, alebo vykonávanie riadkov kódu aj s definíciami premenných, prípadne komentáre alebo sa počítajú riadky ukončené logickými oddeľovačmi. Určiť, ktorá z týchto definícií riadka kódu je správna je ťažké a tak sa pri výbere toho, ktorú použiť zvyčajne vyberie tá, ktorá poskytuje v závere čo najmenšiu chybu.

Kým predchádzajúce veličiny pracovali s reálnymi chybami programu, ktoré sú zavinené nesprávnou implementáciou alebo návrhom, veličina Zákaznícke problémy berie do úvahy aj chyby, za ktoré nie je zodpovedný len programátor. Ide o problémy spojené s použiteľnosťou programu, nejasnou dokumentáciou alebo chybnými informáciami. Hlavným problémom pri získavaní informácií potrebných pre určenie tejto veličiny je duplicita údajov získaných od zákazníkov. Táto veličina je vyjadrená v jednotke PUM (Problem per User Month). Popisuje celkový počet problémov, ktoré nastali v období jedného mesiaca používania softvéru zákazníkom.

Zákaznícka spokojnosť. Predstavuje veličinu, v ktorej sa získava priama spätná väzba od zákazníka. Naproti Zákazníckym problémom predstavuje veličinu, ktorá poskytuje pohľad na softvérový produkt očami zákazníka. Jej meranie vo väčšine prípadov prebieha formou dotazníkov, v ktorých je produkt hodnotený zákazníkmi vzhľadom na ich očakávania a skúsenosti s produktom. Pri hodnotení sa využíva stupnica o piatich stupňoch.

- Veľmi spokojný
- Spokojný
- Neutrálny
- Nespokojný
- Veľmi nespokojný

Konečná hodnota veličiny môže predstavovať :

- Percento veľmi spokojných zákazníkov
- Percento spokojných zákazníkov (veľmi spokojných a spokojných)
- Percento nespokojných zákazníkov (veľmi nespokojných a nespokojných)
- Percento neuspokojených zákazníkov (neutrálny, nespokojný a veľmi nespokojný)

V praxi sa nakoniec zvyčajne použije tá variácia, ktorá čo najviac redukuje počet

nespokojných zákazníkov a vytvára priaznivý obraz o produkte[5].

Aký je teda záver?

Hlavným rozdielom pri použití charakteristík softvéru a metrík pre kvalitu softvéru je pravdepodobne v tom, že kým charakteristika softvéru hovorí: "Toto je dobrý softvér". Metrika hovorí: "Tento softvér je na toľkoto percent dobrý, alebo zlý". Charakteristika predkladá nejakú víziu ideálneho, kvalitného programu tak ako by ho napísal sám Pánboh, ktorý sa chcel zavďačiť používateľovi. Naproti tomu pri použití metriky dostaneme reálny odhad nakoľko sa náš program približuje predstave zákazníka, ktorý práve v procese tvorby programu zohráva úlohu malého božstva, o ktorom všetci vedia, že tu je, že niečo chce, ale nikto nevie poriadne čo.

Inými slovami by sa dalo povedať, že program treba navrhovať tak aby spĺňal charakteristiky kvalitného programu, ale taktiež aby mal čo najlepšie výsledky pri použití metrík pre kvalitu softvérového produktu.

Použitá Literatúra

1. ISO-9000:2000, Standards for quality management systems
2. Crosby, P.: Quality is Free, 1979
3. Oxford dictionary, http://www.askoxford.com/concise_oed/quality?view=uk, 23,20,2007
4. Akingbehin, K.: Taguchi Larger-the-Best Software Quality Metrics
5. Kan, S.: Software Quality Metrics Overview, URL: <http://www.informit.com/articles/article.aspx?p=30306>, stiahnuté 25,10,2007

Annotation

And what about quality?

According to increasing complexity a importance of software products, the phrase "Software quality" is becoming more important. However we are using this phrase more and more often, there are not many people which can fully understand meaning of this two abstract words. The Goal of this essay is to introduce various ways in which is quality of software perceived in present days. There will be also described pros and cons of this ways according to user's and programmer's view. The will be part describing the measurment and matric used to measure quality of software.