VODOPÁD ALEBO AGILNÉ METÓDY – KAM ZA KVALITOU?

Malé zamyslenie sa nad kvalitou nielen v softvérových projektoch.

František Nagy

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
frantisek.nagy.ml@gmail.com

Abstrakt. *Kvalita je pri vývoji akéhokoľvek produktu nesmierne dôležitá. To platí aj pri vývoji softvéru. V tejto eseji sa pozrieme na dva rôzne prístupy k procesu vývoja, vodopádový model a agilné metódy a porovnáme ako zabezpečujú kvalitu. Vyhodnotíme najmä pri ktorom z nich má ako vývojár tak aj používateľ nad kvalitou výsledného produktu väčšiu kontrolu.*

Kľúčové slová: *kvalita softvéru, kontrola kvality, vodopádový model, agilné metódy*

Úvod

V dnešnej dobe nás pri výbere produktu zaujíma viacero kritérií. Pri kúpe auta to môže byť napríklad jeho cena, výkon, spotreba atď. Pri drvivej väčšine výrobkov nás zaujíma aj ich kvalita. Nechceme predsa aby bolo naše nové auto každý mesiac v oprave, alebo aby sa nám nový dom zrútil na hlavu. Pri softvéri, ktorý si vyberáme postupujeme rovnako. Málokedy sa stane, že nemáme na výber a musíme si vybrať len jeden softvér. Rovnakú funkcionalitu nám väčšinou ponúka viacero riešení a my si teda môžeme vybrať to najkvalitnejšie. Ešte väčšiu úroveň kvality očakávame, ak si zákazník nechá vyvinúť softvér na mieru podľa svojich požiadaviek. Keďže nás ale zaujíma vývoj softvéru, budeme sa navyše zaoberať kvalitou z pohľadu vývojára.

Čo ale máme rozumieť pod pojmom kvalita? Podľa normy ISO 9000 je kvalita stupeň splnenia požiadaviek súborom inherentných znakov. Podľa tejto definície by bol softvér kvalitný, ak spĺňa požiadavky zákazníka. To zodpovedá funkcionálnym požiadavkám na softvér. Podľa [3] však pojem kvalita zahŕňa aj ďalšie faktory, okrem iného včasnosť, teda splnenie časového plánu a efektívnosť nákladov, tj. schopnosť dokončiť hotový produkt v rámci nákladov, ktoré sú naňho vyhradené. Podľa môjho názoru sú z pohľadu zákazníka tieto faktory minimálne rovnako dôležité, ako splnenie požiadaviek. Napríklad ak dve firmy vyvinú rovnaký systém, ale jednej z nich to trvá dvakrát tak dlho, alebo si za vývoj vypýta dvojnásobnú sumu, tak hoci sú oba systémy identické a spĺňajú požiadavky, zákazník by si určite vybral lacnejší produkt, prípadne ten s kratším časom vývoja. Pre zákazníka je tento produkt jednoducho kvalitnejší.

Pre vývojára sú navyše dôležité aj ďalšie vlastnosti softvéru, napríklad znovupoužiteľnosť, prenosnosť, udržiavateľnosť, pochopiteľnosť zdrojového kódu a iné [3]. Toto sú nefunkcionálne požiadavky na softvér. Keďže vývojár softvér vytvára, má nad jeho kvalitou spravidla oveľa väčšiu kontrolu ako zákazník. Určite si rozmyslí, či bude mať projekt so zrozumiteľným zdrojovým kódom, rozdelený do logických častí, ktoré je jednoduché upraviť a prípadne znovu-použiť, ako keby mal mať projekt nezrozumiteľný, že mu po pár dňoch nerozumie ani on sám a namiesto aby ho vedel upraviť ho radšej vytvorí celý od začiatku.

Aby bol softvér kvalitný, musí teda spĺňať ako funkcionálne, tak aj nefunkcionálne požiadavky. Zrejme každý vývojár sa snaží dodať kvalitný softvér, inak by na trhu veľmi dlho neprežil. Predtým než však začne softvér vyvíjať, musí si vybrať model vývoja, ktorý použije. Ovplyvní však tento výber kvalitu výsledného produktu? Porovnajme si dva rôzne prístupy k vývoju softvéru a aký je ich prístup ku kvalite. Budeme porovnávať vodopádový model a agilné metódy vývoja.

Vodopádový model je jedným z najstarších a najznámejších modeloch vývoja. Je to sekvenčný model, teda jednotlivé procesy po sebe nasledujú postupne a ďalší začne až keď sa predchádzajúci ukončí, resp. výstup jedného procesu je vstupom pre druhý. Väčšinou sú procesy rozdelené na: definíciu požiadaviek, návrh, implementáciu, verifikáciu a údržbu. Niektoré zdroje môžu uvádzať mierne odlišné fázy, ale princíp je vždy rovnaký.

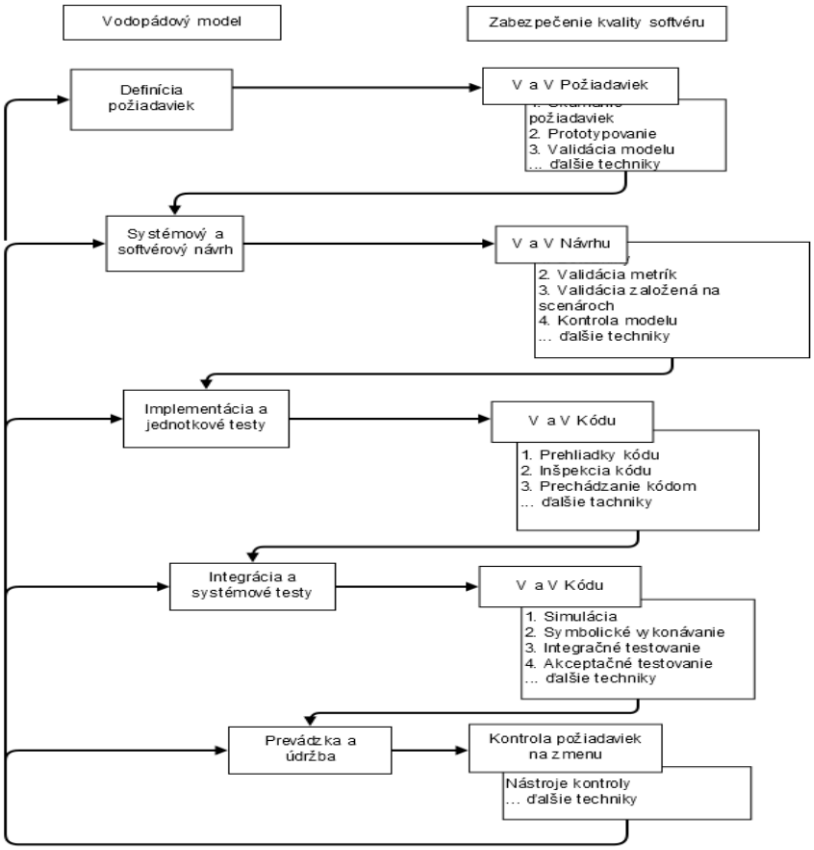
Ako protiklad oproti vodopádovému modelu si vyberieme agilné metódy. Agilné metódy vývoja sú založené na iteratívnom a inkrementálnom vývoji. Vývoj teda postupuje v cykloch, keď vždy na konci cyklu je čiastočný produkt prezentovaný zákazníkovi, ktorý má možnosť následne upraviť svoje požiadavky do ďalšieho cyklu, aby výsledný softvér bol pre neho ešte vhodnejší.

Vodopádový model a kvalita

Ako zabezpečuje kvalitu vodopádový model? Ako sme už spomínali, vodopádový model je rozdelený na niekoľko častí, pričom výstup jednej je vstupom pre nasledujúcu. Celá jedna fáza vývoja je venovaná verifikácií a validácií implementovaného softvéru, no čo ak bol celý softvér zle navrhnutý? Kontrola kvality preto musí byť prítomná aj v ostatných fázach projektu. Najväčšia časť zabezpečenia kvality projektu, ktorý je vyvíjaný týmto

spôsobom spočíva najmä v úseku medzi dvoma časťami (Obrázok 1). Teda výstup je preverený, či je dostatočne kvalitný a ak je posúdené, že spĺňa požiadavky, je posunutý do ďalšieho procesu. Ak by však takouto kontrolou kvality neprešiel, bude vrátený späť do predchádzajúceho procesu na prepracovanie.

Ďalším možným kritériom, na ktoré môžeme pri kontrole kvality prihliadať je jej forma. Postupy kontroly môžeme rozdeliť na statické a dynamické [1]. Statické postupy sa odlišujú od dynamických tým, že nezahŕňajú vykonávanie kódu. Takéto postupy môžu byť napríklad revízia spolupracovníkov (peer review), analýza zdrojového kódu, napríklad aj s pomocou podporného softvéru a pod. Naopak dynamické postupy sú najmä testovanie a simulácia. Ako môžeme vidieť už z členenia vodopádového modelu, tieto postupy prichádzajú na rad až v neskorších častiach vývoja.



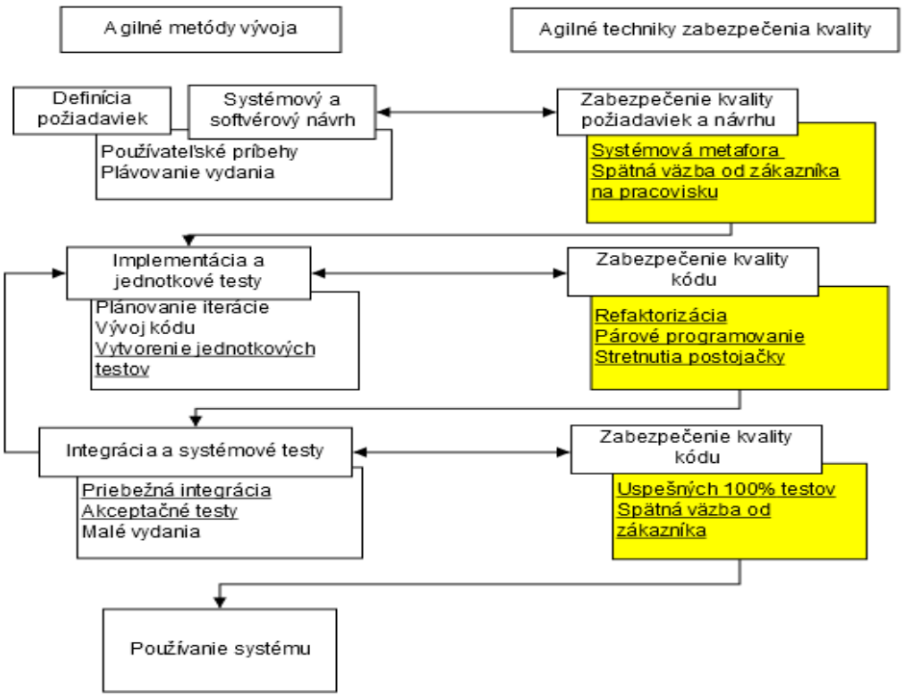
Obr. 1.

Vidíme teda, že kontrola kvality je vyhradená na určité úseky. Podľa môjho názoru je najväčším negatívom takéhoto postupu, že hoci je kontrola kvality prítomná počas celého projektu, sú kontroly príliš rozfázované a pri dlhých projektoch môžu byť od seba jednotlivé kontroly kvality príliš vzdialené. Ak sa potom pri kontrole kvality výstupu

danej fázy zistí, že je nedostatočná, môže byť náročné napraviť takýto problém, ktorý mohol vzniknúť ešte na začiatku fázy a mohol tak ovplyvniť jej veľkú časť.

Kvalita v agilných metódach

Na druhej strane pri agilných metódach vývoja je kontrola kvality oveľa častejšia. Keď si prirovnáme jednu iteráciu cyklu agilného vývoja k vodopádovému modelu, nájdeme takmer na rovnakých miestach aj kontrolu kvality. Takýchto iterácií je však počas celého vývoja pomerne veľké množstvo. Kvalita je tak kontrolovaná priebežne, čo zabráni vzniku rozsiahlych chýb, ktoré môže byť potom náročné odstrániť. Takisto nesmieme zabudnúť na jeden z hlavných znakov agilných metód, ktorým je interakcia s používateľom. Po ukončení každého cyklu vývoja je používateľovi predstavený funkčný prototyp a ten tak má jasnejšiu predstavu o vývoji a môže prípadne upraviť svoje požiadavky. Vďaka kratším cyklom a častejšej kontrole kvality je oveľa jednoduchšie zmeniť produkt, aby čo najviac vyhovoval zákazníkovým požiadavkám a bol teda čo najkvalitnejší.



Obr. 2.

Takisto, ako pri vodopádovom modeli, aj pri agilných metódach vývoja sa využívajú statické aj dynamické postupy kontroly kvality. Pri agilnom vývoji sa však kladie oveľa väčší dôraz na dynamickú kontrolu, keďže už od skorých fáz projektu je prítomný spustiteľný prototyp. Agilné metódy navyše využívajú niektoré menej štandardné

postupy kontroly kvality, ktoré sú však účinné minimálne rovnako, ak nie viac. Medzi tieto postupy patrí napríklad párové programovanie, refaktorizácia, neustála integrácia, písanie testov pred písaním kódu a pod. Niektoré z týchto postupov navyše neslúžia výhradne len na kontrolu kvality, ale aj na vývoj.

Z pohľadu zákazníka je jasne vidieť, že pri agilných metódach má prehľad o kvalite výsledného produktu na základe prototypov a zároveň ju aj kontroluje, keďže môže podávať vývojárom spätnú väzbu. Podľa môjho názoru sú pri agilných metódach vyhliadky na kvalitný softvér oveľa väčšie ako pri vodopádovom modeli, pretože sa ľahko môže stať, že zákazník nevie na začiatku projektu dostatočne definovať svoje požiadavky a až po dodaní výsledného softvéru zistí, že by niečo potreboval inak.

V štúdii [2] prišli k záveru, že chýba dostatok dôkazov pre určenie nadradenosti medzi vodopádovým modelom a agilnými metódami z pohľadu vývojára. Ja však vidím pre vývojára prínos agilných metód v častejších iteráciách kontroly kvality, čo by malo zabrániť vzniku veľkých chýb, ktoré by bolo následne náročné odstraňovať.

Záver

Po prehliadke oboch metodológií môžeme skonštatovať, že obe obsahujú aj kontrolu kvality. Podľa môjho názoru je však kontrola kvality pri agilnom vývoji vďaka kratším cyklom precíznejšia a zabezpečuje v konečnom dôsledku kvalitnejší produkt ako pre vývojára, ale najmä pre koncového zákazníka. Ten je pri agilných metódach oveľa viac zapojený do vývoja a preto má o vývoji jednak lepší prehľad a väčšiu kontrolu. Hoci z pohľadu vývojára nemusí byť až tak veľký rozdiel medzi zabezpečením kvality vo vodopádovom modeli a v agilných metódach, ako zákazník by som si určite vybral radšej agilné metódy.

Použitá literatúra

1. Ming Huo; Verner, J.; Liming Zhu; Babar, M.A.; , "Software quality and agile methods," Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International , vol., no., pp. 520- 525 vol.1, 28-30 Sept. 2004
2. Mitchell, S.M.; Seaman, C.B.; , "A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review," Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on , vol., no., pp.511-515, 15-16 Oct. 2009
3. Mnkandla, E.; Dwolatzky, B.; , "Defining Agile Software Quality Assurance," Software Engineering Advances, International Conference on , vol., no., pp.36, Oct. 2006

Annotation

Waterfall or agile methods – where to quality?

Quality is immensely important when developing any product. That holds even for development of software. In this essay, we will take a look on two different approaches to process of development, waterfall model and agile methods and compare how they assure quality. We will especially evaluate, which offers more controll over quality of final product to both developer and customer.