STRET SQA S AGILNÝM PRÍSTUPOM VÝVOJA. PRÍNOS, ALEBO POHROMA ?

Človek získava silu tým, že spoznáva svoje slabosti.

Michal Dulačka

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

Abstrakt. *Zložitosť súčasného softvéru rastie obrovskou rýchlosťou. Stáva sa čoraz komplexnejším, dômyselnejším a prepracovanejším. Jeho vývoj a udržiavanie už nie sú triviálnymi záležitosťami, preto je potrebné klásť veľký dôraz už na samotný proces jeho vývoja. Kvalita softvéru a kvalita vývoja softvéru už nie sú zriedkavé pojmy. Vývoj sa aj v tejto oblasti pohol dopredu a v rámci firiem začali vznikať skupiny špecialistov, ktorí sa zaoberajú problematikou SQA (Software Quality Assurance). Úlohou SQA je zaručiť , že všetky štandardy a procesy použité pri vývoji projektu budú vhodné a korektne implementované. Zasahuje teda do samotného procesu vývoja a dochádza teda ku konfrontácii s prístupmi, ktoré sú taktiež súčasťou vývoja, ale ich úloha je odlišná. Zaujímavý je v tomto prípade stret SQA s agilnými metódami. V tejto eseji sa zamýšľam nad tým, čo je výsledkom tejto interakcie a nad úplne novým pohľadom na dosahovanie kvality softvéru.*

Kľúčové slová: *agilné metódy, konvenčné metódy, software quality assurance, kvalita*

Úvod

Dnešný svetový trh a v skutočnosti aj takmer celý svet je zaplavený množstvom ponúkaných služieb a produktov. Vyrábajú sa ich doslova milióny. V obchodoch nachádzame stovky druhov rôznych zariadení, elektroniky, ale aj potravín. Pri ich nakupovaní sme ovplyvnení ich dizajnom, pekným balením a v neposlednom rade aj rôznymi psychologicko-reklamnými ťahmi. Veľmi dôležitým a rozhodujúcim kritériom však v mnohých prípadoch býva kvalita. Jej cieľom je spokojnosť zákazníka. Od kvality výrobkov často závisí prežitie spoločnosti, ktorá daný produkt ponúka. Pojem „kvalita" je v súčasnosti už taký rozšírený a žiadaný, že jej dosahovanie a udržanie sa stalo takmer samostatným vedným odborom. Už nie je ničím výnimočným, že mnohé spoločnosti zakladajú špecializované oddelenia, ktoré sa touto problematikou zaoberajú. Každé odvetvie sa však môže na kvalitu pozerať z vlastného uhla pohľadu. Je totiž veľký rozdiel, ak sa snažíme vyrobiť kvalitný rýľ, alebo mikroprocesor. Ešte špeciálnejším prípadom je dosahovanie kvality pri vývoji softvéru.

Softvér súčasnosti už nie je taký jednoduchý ako v minulosti. Stáva sa čoraz inteligentnejší, komplexnejší a prepracovanejší. Navyše je potrebné zabezpečiť, aby pracoval dostatočne efektívne, bol ľahko udržiavateľný a rozšíriteľný. Metódy dosahovania kvality pri vývoji softvéru taktiež podliehajú vývinu a vzájomnému ovplyvňovaniu s metódami, ktoré majú vo vývojovom procese svoje špecifické a dôležité postavenie. Dokonalým príkladom je dosahovanie kvality v spojení s agilnými alebo konvenčnými metódami vývoja. Tieto vzájomné konfrontácie sú často zdrojom zaujímavých úkazov, ktoré stoja za hlbšiu úvahu.

Ako dosiahnuť kvalitu ?

Kvalita softvéru sa dá vnímať z rôznych aspektov. Niekto ju vníma ako nízky počet chýb vo výslednej aplikácii a úspešný prechod záverečný testovaním, niekto ako stabilnosť aplikácie a korektnosť jej výstupov. Keď teda chceme skúmať kvalitu aplikácie a jej vývoja, je potrebné najskôr definovať, čo tento pojem predstavuje. Kvalita softvéru podľa [3] znamená: *„Kvalita softvéru je plánovaná a systematická množina aktivít, ktoré zaručia, že kvalita je súčasťou softvéru. Pozostáva z metodológie zaručovania kvality, kontroly kvality softvéru a inžinierstva kvality softvéru. Ako atribút, kvalita softvéru predstavuje:*

- *stupeň, ktorý softvér, proces, alebo komponent dosahuje pri spĺňaní požiadaviek*
- *stupeň, ktorý softvér, proces, alebo komponent dosahuje pri napĺňaní používateľových potrieb a očakávaní."*

Je teda logické, že kvalitu nezabezpečí napríklad len samotné testovanie. Nemôžeme ju do produktu „vtestovať". Je potrebné nielen zabezpečiť a využiť metódy, ktoré zabezpečia kvalitu softvéru, ale aj metódy, ktoré skvalitnia samotný proces vývoja. Práve touto stránkou veci sa zaoberá SQA (Software Quality Assurance), teda už spomínaná metodológia zaručovania kvality. Jej úlohou je totiž zabezpečiť výber vhodných procedúr, procesov a štandardov, ktoré sú v vývoji použité a ich korektnú implementáciu. V súčasnosti je postavenie SQA skutočne veľmi významné a je objektom mnohých štúdií. Je v nej obsiahnutých množstvo modelov a rôznych prístupov (CMM, CMMI a pod.).

Proces vývoja ovplyvňuje cielenými aktivitami a tieto zahŕňajú napríklad kontrolu procesu, dokumentáciu, audity, verifikácie a validácie. Zaujímavý je však fakt, že SQA nejakým spôsobom ovplyvňuje, alebo sa stáva súčasťou už jestvujúceho procesu vývoja softvéru. Ten zahŕňa rôzne postupy, praktiky a techniky, ktoré v jeho vnútri plnia špecifické úlohy a ako celok formujú jeho charakter a vlastnosti. Dochádza tu teda ku konfrontácii a vzájomnej interakcii SQA s inými súčasťami samotného procesu. Ak spojíme dokopy SQA s konvenčnými metódami vývoja softvéru, výsledok bude iný, ako kombinácia SQA a agilných vývojových procesov. Obe tieto interakcie sú zdrojom rozdielnych výsledkov s rôznym výsledným efektom.

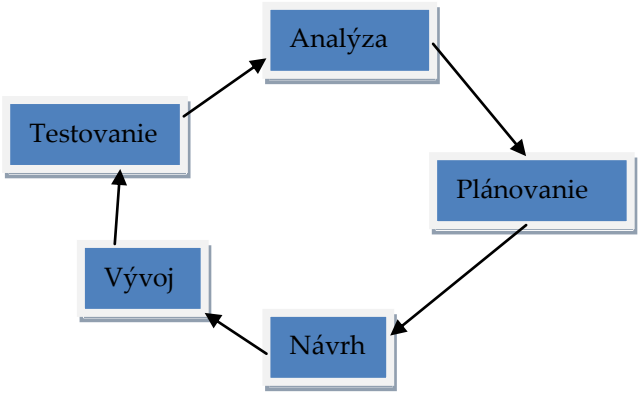
SQA a konvenčné metódy vývoja

Konvenčné metódy vývoja sú vo svojej podstate orientované smerom k procesom. Proces vývoja je zložený z vopred preddefinovaných aktivít, ako zber požiadaviek, analýza, návrh architektúry, testovanie a implementácia atď., ktoré majú svoje vopred definované poradie. Konvenčné metódy vývoja bývajú často kombinované aj s modelmi, pričom najpoužívanejšie sú vodopádový a špirálovitý. Charakteristickou črtou je zber požiadaviek a ich analýza v úvodnej etape vývoja. Rozhodnutia a štandardy sú určované manažmentom organizácie. Základná charakteristika konvenčných metód je teda už aspoň čiastočne objasnená. S ňou súvisí aj spôsob, akým SQA v tomto prípade skvalitní proces vývoja, ale aj jeho výstup. Aktivitami, ktoré sú pre dosahovanie kvality v tomto prípade ťažiskové sú evaluácie procesu vývoja, audity a stretnutia, ktoré monitorujú, či proces spĺňa a sleduje definované štandardy. Bežné sú rôzne revízie a inšpekcie softvéru, ale aj dôsledná dokumentácia. Pre dosahovanie a udržanie kvality vývoja je kľúčová aj validácia a verifikácia. Mechanizmov pre skvalitňovanie je tu teda naozaj mnoho. Všetky tie testy, kontroly a neustály monitoring musia predsa zabezpečiť maximálnu kvalitu. Navonok sa javí všetko dokonale zabezpečené a ošetrené. No na druhej strane tu zabezpečovanie kvality pôsobí trochu stiesnene. Nad procesom vývoja je akoby vrstva, ktorá vykonáva ustavičnú kontrolu a odhaľuje chyby, ktoré už vznikli. Iná možnosť však v tomto prípade zrejme ani nejestvuje. Konvenčné metódy vývoja už majú svoju danú formu, ktorú SQA nedokáže nijako ovplyvniť. Vynára sa tu otázka, či neustále pridávanie nových a nových kontrolných mechanizmov vždy smeruje k zvyšovaniu kvality, alebo skôr k ťažkopádnosti a zbytočnej robustnosti. Ideálnejšie by určite bolo, keby kvalita bola určitým spôsobom už súčasťou samotného procesu. Vďaka tomu by nad procesom nemuselo prebiehať toľko kontroly, vďaka čomu by bol odľahčený o náklady spojené s kontrolnou vrstvou.

SQA a agilné metódy vývoja

Aby sme dokázali dostatočne pochopiť postavenie SQA v agilných metódach je vývoja, je potrebné najskôr pochopiť aspoň ich samotný základ. Slovo agilnosť v názve značí živosť, čulosť a obratnosť. Agilné metódy sa predovšetkým snažia zbaviť určitej „ťažkopádnosti“, ktorá býva spájaná hlavne s konvenčnými metódami vývoja. Agilné metódy sú: iteratívne, inkrementálne a v centre pozornosti sa nachádza človek, nie proces. Preferujú viac kratších iterácií v priebehu vývoja, pričom každá iterácie zahŕňa niekoľko fáz vývoja (Obr. 1). Po ukončení každej iterácie, je produkt odoslaný používateľovi na revíziu. Nová iterácia

potom dopĺňa do produktu ďalšiu funkcionalitu, pričom zohľadňuje používateľovu spätnú väzbu. Dôraz je teda kladený na lepšiu spoluprácu medzi ľuďmi a lepšiu a rýchlejšiu reakciu na zmenu. Naopak vytváranie a striktnosť dokumentácie už nemá takú veľkú váhu. Ako sami vidíme, agilné metódy v porovnaní s konvenčnými, na proces vývoja softvéru a kvalitu pozerajú úplne odlišným spôsobom. Kde konvenčné metódy kladú dôraz na samotný proces, dokumentáciu, dodržiavanie zmluvy a dodržiavanie plánu, tam sa agilné metódy zameriavajú ľudí, fungujúci softvér, spoluprácu so zákazníkom a schopnosť reagovať na zmeny. Z toho vyplýva, že aj aktivity SQA budú značne odlišné. Pojem SQA v agilných metódach je však priveľmi všeobecný. Aby bolo možné vytvoriť si dokonalejšiu predstavu a lepšie pochopiť, akým štýlom tu býva dosahovaná kvalita, je potrebný bližší pohľad a konkretizácia. Existuje totiž niekoľko konkrétnych príkladov agilných metód vývoja a každý tento konkrétny prípad má svoje špecifiká a využíva rôzne aktivity SQA.



Obr. 1. Iteratívny a inkrementálny vývoj.

Všetci za jedného, jeden za všetkých

Prvým ukážkovým reprezentantom agilných metód vývoja je takzvané Extrémne programovanie. Jednou z techník, ktorá sa tu používa na dosahovanie kvality je refaktoring. Ide o upravenie vnútornej štruktúry kódu bez toho, aby došlo k zmene funkcionality. Táto technika sa využíva hlavne na odstránenie duplicít, upravenie komunikácie a reštrukturalizáciu. Veľké pozitívum je, že táto technika sa snaží predchádzať vzniku komplikácií a zvyšovanou nákladov v budúcnosti. Je jasné, že refaktoring zdokonaľuje štruktúru kódu a teda zvyšuje kvalitu. Určite by však bolo lepšie vyhnúť sa potrebe refaktoringu tak, že už novovytvorený kód bude dostatočne kvalitný. Napriek tomu, že je vyvíjané veľké množstvo rôzneho softvéru, samotné zdrojové kódy v týchto produktoch majú veľa spoločného, pretože sa snažia dodržiavať rovnaké princípy. Na základe štúdií by teda mohli byť vytvorené určité šablóny a postupy. Na základe nich a pomocou získaných skúseností, by tak programátor mohol písať čistejší kód, čo by znížilo potrebu refaktoringu.

K zvyšovaniu kvality prispieva aj takzvané „kolektívne vlastníctvo“. Znamená to, že nikto nemá zodpovednosť za konkrétnu časť kódu alebo systému. Každý vývojár môže kedykoľvek ľubovoľnú časť kódu pozmeniť. Ako veľmi výhodná sa táto technika javí napríklad, ak ide o opravovanie chýb. Každý programátor môže opraviť chybu bezprostredne po tom, ako ju nájde. Problém však môže nastať, keď omylom pri opravovaní jednej chyby nechtiac vytvorí novú. Kód, ktorý upravuje môže mať rôzne závislosti, ktoré nemusia byť hneď očividné. Netreba zabúdať ani na fakt, že ak viacero vývojárov pracuje s jedným kódom, môže to produkovať nedorozumenia. Nebezpečenstvo nastáva hlavne v prípade, ak nie je zabezpečený komunikačný spôsob, ktorý informuje o tom, aké zmeny boli v zdrojovom kóde vykonané a čo ešte treba upraviť.

Najpreslávenejšou a najzaujímavejšou SQA technikou v Extrémnom programovaní je Programovanie v pároch (angl. Pair programing). Ide o techniku, kde vývojári pracujú vo dvojiciach a pracujú obidvaja na jednom a tom istom kóde, na jednom počítači. Zdieľajú vzájomne svoje myšlienky a nápady, učia sa od seba, no predovšetkým sa istým spôsobom vzájomne kontrolujú. Táto kontrola je však úplne prirodzenou súčasťou vývoja a nijako nezaťažuje proces ďalšími veľkými nákladmi a spotrebou času. Riziko výskytu závažných chýb v kóde, prípadne návrhu je teda výrazne znížené, čo tiež zvyšuje kvalitu produktu. Takáto úzka spolupráca navyše môže byť predpokladom k rýchlejšiemu profesionálnemu rastu a získavaniu skúseností, čo môže mať tiež v budúcnosti dopad na kvalitu procesu vývoja.

Skrumáž

Medzi agilné metódy vývoja patrí aj SCRUM. Ide tu o metódu vývoja, ktorá sa zameriava skôr na vrstvu organizácie a manažmentu projektu v rámci tímu. Hlavnými charakteristikami SCRUMu sú komunikácia a odozva. Na organizáciu a sprehľadnenie procesu vývoje je tu využívaný Product Backlog. Ide o zoznam, ktorý je udržiavaný počas celého procesu vývoja. Zaujímavé je, že môže do neho zasahovať každý, od zákazníka až po manažéra. V tomto zozname sú obsiahnuté všetky funkcionality a požiadavky, ktoré by mali byť implementované a ich popis. U každej úlohy a funkcionality býva taktiež uvedená priorita a približný odhad, koľko úsilia a času jej splnenie zaberie. Vďaka tomu má aj samotný zákazník aspoň základný prehľad o tom, koľko úsilia a času vývoj zaberie a koľko ho to bude približne stáť. Táto technika zvyšuje kvalitu procesu predovšetkým v oblasti komunikácie všetkých zainteresovaných strán. Jej inovatívnosť spočíva hlavne v otvorenom prístupe, ktorý určite prispieva ku spokojnosti zákazníka. Vďaka nej sa totiž aspoň čiastočne znižuje riziko prílišného predraženia projektu, keďže zákazník má prehľad o tom, čo sa bude diať.

Ďalšou podstatnou technikou SQA je Daily SCRUM. Ide o každodenné stretnutie celého vývojového tímu, ktoré trvá približne pätnásť minút. Vedie ho SCRUM manažér a podstatou tohto stretnutia je zisťovanie, či členovia vývojového tímu nenarazili na nejaké problémy, ktoré treba riešiť. Navyše v rámci stretnutia každý člen hovorí o tom, ako pokračuje v práci, koľko toho stihol, koľko nestihol a hodnotí svoj progres v rámci plnenia svojej úlohy. Vďaka tomu je možné takmer ihneď identifikovať problémy a navyše to núti členov tímu k zodpovednejšiemu prístupu k práci. Ide o veľmi dobrý spôsob monitoringu a odhaľovania problémov priamo v ich zdroji, teda v ľudskej nevedomosti, lenivosti

a náklonnosti robiť chyby. Aj napriek tomu táto forma nie je zárukou úspechu. Nič totiž nebráni individuálnym členom tímu, aby zamlčali stagnáciu, alebo prípadné problémy, na ktoré narazili. Môže to byť spôsobené napríklad nesprávnym prístupom manažéra k zamestnancom a riešeniu problémov.

Najskôr testy, potom práca, potom zábava

Akýkoľvek vývoj, alebo výrobu produktu si bez testovania už ani nevieme predstaviť. Poradie je jasné a logické. Keď je produkt hotový, otestujeme ho. Skutočnosť že by to mohlo byť aj opačne, sa zdá trošku pritiahnutá za vlasy. Napriek tomu je realitou a navyše veľmi účinnou.

Jednou z techník dosahovania kvality, ktorá sa využíva u viacerých agilných metód vývoja je Vývoj riadený testom. U agilných metód vývoja však testovanie produktu viac nie je v kompetencii testera, ale vývojára samotného. Zaujímavé však je, že testy bývajú vytvorené skôr, ako samotná časť systému, ktorá bude testovaná. Keď sú testy vytvorené, vývojár implementuje príslušnú časť kódu tak, aby prešla testom. Výhoda spočíva v tom, že je nútený vopred sa lepšie zamyslieť nad návrhom implementovanej časti. Navyše, keďže je systém vyvíjaný inkrementálne a iteratívne, chyby sú odhalené skôr a jednoduchšie. Výsledkom Vývoja riadeného testom je zníženie nákladov na opravovanie chýb. Systém je testovaný prakticky pravidelne od úvodnej inkrementácie, kedy ešte nenadobúda také obrovské rozmery a komplexnosť.

Prínos a nános agilných metód

Prínos agilných metód v oblasti kvality vývoja a produktov je zrejmý. Zdá sa, že nový pohľad na problematiku SQA konvenčné metódy skôr či neskôr vytlačí z hry. Veľkým tromfom agilných metód je inkrementálny a iteratívny vývoj. Budovanie systému po častiach spojené s pravidelnou komunikáciou so zákazníkom často ušetrí množstvo nákladov a vedie k jeho väčšej spokojnosti. Je to spôsobené tým, že zákazník sa podieľa na procese vývoja, počas celého jeho trvania, nie len v úvodných fázach. Vďaka tomu sú odhalené nedorozumenia v chápaní požiadaviek vo fázach, keď systém nie je až taký robustný a jeho úprava nie je taká nákladná. Naopak, jestvujú aj prípady, kedy nie je možné byť s klientom v pravidelnom kontakte, vtedy môže byť potreba pravidelnej komunikácie nevýhodou.

Svetlou stránkou agilných metód je aj spôsob testovania. Po každej iterácii je totiž systém testovaný. Kód, ktorý takto vzniká je teda kvalitnejší, pretože chyby sa v ranných fázach vývoja odstraňujú jednoduchšie. Naopak, u konvenčných metód toto testovanie prebieha až v posledných fázach, kedy je už systém komplexný a jeho úprava je náročnejšia. Keďže sa u agilných metód o testovanie nestará tester, ale vývojár dochádza k šetreniu času v oblasti komunikácie (spisovanie záznamov o chybách, ich naštudovanie a pod.). Naopak trochu nepríjemný je fakt, že kým sa vývojový pracovník venuje testovaniu, nemôže sa venovať vývoju, čím dochádza k menšiemu spomaleniu. Techniky ako Refaktoring, Programovanie vo dvojiciach a Vývoj riadený testom, taktiež veľmi prispievajú ku skvalitneniu kódu. Pomáhajú predchádzať vzniku chýb, odstraňujú z neho

už jestvujúce chyby, duplicity a zjednodušujú ho. To má za následok výrazné zníženie nákladov na údržbu a jednoduchšie rozširovanie systému.

Naopak slabinou agilných metód programovania je dokumentácia a dodržiavanie štandardov, ktoré môžu niektorí zákazníci vyžadovať. Nedostatkom je aj slabá kvalita na organizačnej úrovni. To býva prekážkou najmä pri ich nasadzovaní na väčších komplexnejších projektoch, kde pracuje niekoľko nezávislých agilných tímov. V agilných metódach vývoja totiž vôbec nie je definovaný spôsob komunikácie medzi viacerými nezávislými tímami. V tomto momente je dôležité zamyslieť sa nad otázkou, či nejestvuje spôsob, ako prepojiť určité vlastnosti konvenčných metód s agilnými. Konvenčné metódy vývoja predsa vynikajú v oblasti organizácie a dodržiavania štandardov.

V niektorých agilných tímoch prevláda často tzv. „sebaorganizácia". Prostredie je teda voľnejšie, menej stresové, lepšie sa v ňom pracuje, čo tiež zvyšuje kvalitu. Dôležité však je, aby boli zamestnanci správne motivovaní a nezneužívali danú voľnosť. Vtedy je možné dosiahnuť maximálnu efektivitu. V opačnom prípade môže byť rovnosť a voľnosť časovanou bombou. U agilných metód teda treba dávať pozor na to, aby bola dodržiavaná pracovná disciplína a morálka. Musí taktiež byť stanovená určitá hierarchia, ktorá bude určite prospešnou hlavne, keď je potrebné urobiť závažnejšie rozhodnutie.

Kvalita si tu často vyžaduje kvalitný personál. Vývojár predsa komunikuje so zákazníkom, vytvára testy, plánuje a samozrejme vyvíja. Pri tom často nad ním nie je stanovený žiadny komplexnejší kontrolný mechanizmus. Preto tento zamestnanec musí byť dostatočne skúsený.

Človek verzus proces

Ako už bolo spomenuté, máme tu dva spôsoby vývoja softvéru, ktoré sa snažia o dosiahnutie kvality. Každý z týchto prístupov má pritom na tento problém svoj vlastný pohľad. Agilné metódy vyzdvihujú stavajú do centra pozornosti človeka a konvenčné metódy proces. Stojí teda za to, zamyslieť sa nad tým, čo je v tomto prípade rozumnejšie. Na prvý pohľad sa to zdá jasné. Veď produkt je výstupom procesu. Ak teda máme dostatočne monitorovaný a kontrolovaný proces, všetko musí byť v poriadku. Zdá sa však, že nový pohľad agilných metód túto teóriu vyvracia. Stojí teda za to, zamyslieť sa nad tým, v čom spočíva podstata úspechu agilných metód.

Proces vývoja je vždy do veľkej miery ovplyvnený ľuďmi, ktorí sa na ňom podieľajú, alebo ho ovplyvňujú. Nie sú to teda len vývojári, ale napríklad aj zákazník. Ak všetkým týmto zainteresovaným stranám vytvoríme kvalitné podmienky pre prácu a spoluprácu, máme predpoklady pre zvýšenie kvality samotného procesu vývoja. Kvalita je teda akoby vstavaná už do základu procesu. Ak je kvalita už jeho súčasťou, nie je potrebná robustná kontrola a monitoring. Ak k tomuto všetkému navyše pridáme vývoj po častiach a kvalitný spôsob testovania, výsledkom je proces, ktorý má kvalitu zakomponovanú už vo svojom základe. Dochádza tu teda k dokonalejšiemu splynutie SQA s procesom samotným, pričom proces sa čiastočne kvalite prispôsobuje. V tejto stránke konvenčné metódy vývoja značne zaostávajú.

Záver

Pojem „kvalita“ je vo vývoji softvéru už takmer kľúčový. Spokojnosť zákazníka je základom nie len pre prežitie produktu, ale aj samotnej spoločnosti. Jestvujú dva hlavné prúdy vývoja softvéru. Konvenčný spôsob, ktorý sa zameriava na proces a agilný, ktorý kladie do centra pozornosti človeka. Práve druhý spôsob sa javí ako výhodnejší a praktickejší. Oba tieto štýly sa pozerajú na kvalitu z odlišných uhlov pohľadu, a obidva majú svoje výhody aj nevýhody. V tejto práci som sa bližšie zameral na jednotlivé techniky dosahovania kvality. Vyzdvihol som ich výhody a kritizoval ich nedostatky. Práve pri odhalení nedostatkov v kvalite agilných metód sa vynorila zaujímavá otázka. Nebolo by možné prevziať niektoré výhodné vlastnosti konvenčných metód vývoja a integrovať ich s agilnými ? Práve hľadaniu odpovede na túto otázku by mohli byť venované ďalšie práce v tejto oblasti.

Použitá literatúra

1. Feldman S.: *Quality Assurance: Much more than testing*, ACM Queue, Feb. 2005
2. Glazer, H., Dalton, J., Anderson, D., Konrad, M., Shrum, S.: *CMMI or Agile: Why Not Embrace Both !*, Carnegie Mellon University, Pittsburgh, 2008
3. http://www.hq.nasa.gov/office/codeq/software/umbrella_defs.htm
4. Imran, U., M., Waqae, A., Z.: *Quality Assurance Activities in Agile*, Blekinge Institute of Technology, Sweden, September 2009, Master Thesis
5. Mnkandla, E.: Defining Agile Software Quality Assurance. In: *Proc. of the ICSEA*, 2006, 72-78

Annotation

Combination of SQA and Agile development methods. Contribution or disaster ?

The complexity of current software is growing rapidly. Software becomes more complex, sophisticated and refined. Its development and maintenance are no longer trivial actions, therefore it is necessary to put a strong emphasis on the process of development. Quality software and quality software development are no longer rare terms. Developments in this area moved forward and many companies formed a group of specialists who deal with SQA (Software Quality Assurance). The role of SQA is to ensure that all standards and processes used in developing the project are appropriate and correctly implemented. Thus SQA interferes in the process of development and therefore there is a confrontation with the approaches that are also part of the development. Some clashes are very interesting, especially between SQA and agile methods. This essay to reflect on what is beneficial and what is the essence of an entirely new perspective on achieving quality.