

MANIFEST PÁROVÉHO PROGRAMOVANIA V TÍME

Párové programovanie nie je len to, že by jeden programoval a druhý sa pozeral

Róbert Kocian

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
xkocianr[zavináč]stuba[.]sk

Abstrakt. *Stále viac a viac rastú nároky na rýchlejší vývoj a kvalitu softvéru s čoho vyplýva že, pre samotných programátorov je to čím ďalej tým náročnejšie. Vyvíjať kvalitnú a komplexnú aplikáciu s neustále rastúcimi nárokmi na jej vývoj je v tejto stále rýchlejšej dobe extrémne ťažká úloha, a preto si táto situácia vyžaduje použitie extrémnych metód vývoja, ktoré tento problém vo veľkej miere eliminujú a je tak možné vyvíjať kvalitný softvér za čo najkratší čas. Vhodnosť použitia Párového programovania je závislá ako od veľkosti projektu, tak od veľkosti tímu. S toho vyplýva otázka. Je Párové programovanie vhodnou metódou pre menšie tímy, ktoré pracujú na menších projektoch? Ak áno, dá sa uskutočniť párové programovanie v menších tímoch tak, aby sa dosiahlo požadovaných kritérií kladených na projekt?*

Kľúčové slová: *extrémne programovanie, párové programovanie, agilné metódy*

Úvod

V dnešnej dobe sú kladené obrovské nároky na vývoj softvéru a jeho kvalitu. Kvalitný vývoj softvéru by sa neobišiel bez vhodnej metodiky. Výber vhodnej metódy vývoja softvéru je kľúčovým krokom, už len preto, lebo zabezpečuje splnenie požadovaného výsledku za stanovený čas. Pri výbere vhodnej metódy sú veľmi dôležité aspekty, ako veľkosť projektu a veľkosť tímu. Ak je metodika zvolená správne je prínosom nielen pre vyvíjaný projekt, ale aj pre samotných členov tímu, ktorým ušetrí veľa času a námahy.

Metód je veľa, ale pokiaľ sa jedná o menšie tímy, kvalitu a rýchly čas vývoja, je agilný prístup vhodným kandidátom, pretože je natívne zameraný na kvalitu, čo už naznačuje aj samotný agilný manifest. Jednou z agilných metód je práve párové programovanie. Párové programovanie je zaujímavá metóda pre niektorých nepoužiteľná ale pre iných je to najvhodnejšia cesta vývoja softvéru. Otázkou preto zostáva, či je vhodné použiť párové programovanie v malých tímoch a projektoch a či dokáže v týchto podmienkach zabezpečiť potrebnú kvalitu vývoja softvéru.

Manažment kvality

Manažment kvality je veľmi dôležitá oblasť vývoja softvéru, keďže bez zabezpečenia kvality by bol vyvíjaný softvér plný chýb. Taktiež jeho samotný vývoj by bol neefektívny, pretože u softvéru platí, že ak je jeho proces výroby kvalitný, tak je kvalitný aj samotný softvérový produkt [2].

Oblasť manažmentu kvality zasahuje do celej štruktúry organizácie. Proces kvality zahŕňa v sebe rôzne normy, štandardy a koncepcie riešenia softvérového projektu. Je dynamický a neustále sa mení tak, aby sa dosiahol čo najkvalitnejší výsledok vynaloženej práce, ktorý spĺňa záujmy organizácie. Samotná firma tento proces riadenia kvality vykonáva pomocou rôznych typov kontrol .

V Európe sa pre riadenie a zaistenie kvality používa medzinárodná sústava noriem ISO 9000. Samozrejme, že zavedenie tejto normy má pre organizáciu svoje výhody aj nevýhody. Jednou z výhod, ktorú by som chcel spomenúť je nesporne to, že normy môžu slúžiť, ako dobrý základ pre dobre fungujúcu výrobu, zlepšujú renomé a hodnotu firmy. Prečo ju teda nepoužívajú všetky firmy? Odpoveď je jednoduchá. Je to príliš nákladné a časovo náročné, a to sa samozrejme prejaví aj na cene softvérového produktu [2].

V každom softvérovom projekte existujú štyri základne premenné, ktoré je možné ovplyvňovať. Sú to cena, čas, rozsah a kvalita. Tieto parametre sú vzájomne prepojené.

Ako vplývajú tieto parametre na softvérový projekt? Ak napríklad znížime náklady, bude to mať za následok pokles kvality a zmenšenie rozsahu projektu, alebo ak budeme chcieť skrátiť čas dodania projektu, povedie to ku zníženiu kvality, rozsahu a pravdepodobne vzrastie aj cena projektu [3].

Nízka kvalita softvéru síce šetrí počiatočné náklady, ale o to viacej bude dodatočných nákladov v budúcnosti. Pre dosiahnutie vyššej kvality sú síce nutné vyššie počiatočné náklady, ale vyššia kvalita potom výrazne šetrí náklady na prevádzku a údržbu. Čo sa týka rozsahu projektu, menšie projekty je možné vyvíjať rýchlejšie, lacnejšie a s lepšou kvalitou než projekty rozsiahlejšie.

Agilný manažment kvality

Potreba rýchlejšieho plánovania a vývoja v rýchlo sa meniacom obchodnom prostredí vyústila do vytvorenia agilných metodík.

Je táto metóda lepšia než ostatné? Agilné metodiky viacej podporujú komunikáciu, majú väčšie zapojenie zákazníkov tak, aby rozdiel medzi očakávaniami zákazníkov

a skutočným softvérom bol menší. Umožňujú, že väčšina softvérových problémov môže byť vysledovaná až ku zlyhaniu komunikácie.

V manifeste, ktorý vytvorili tvorcovia agilných prístupov je presne definované, že najvyššou prioritou sú potreby zákazníka a ich uspokojenie.

Agilný prístup zahŕňa spokojnosť zákazníkov, ale aj kvalitný produkt. Pre zaistenie kvality tradičného projektového manažmentu by projektoví manažéri produkovali kvalitný plán projektu. Agilní projektoví manažéri toto však nerobia, pretože agilný proces sám o sebe poskytuje zabezpečenie kvality a kontroly. Agilita dáva výrobku kvalitu prostredníctvom kombinácie postupov agilného projektového monitorovania a kontroly agilnej realizácie projektu [4].

Prečo práve extrémne metódy vývoja?

Jednou z najznámejších, ale aj veľmi účinnou a pružnou agilnou metódou je práve extrémne programovanie. Pružná metóda je to hlavne preto, že v extrémnom programovaní môžeme dynamicky určovať termíny implementácie funkcií a reagovať na meniace sa potreby alebo zadanie. Neustála spätná väzba vyplývajúca z krátkych cyklov poskytuje extrémnemu programovaniu veľmi dobrú účinnosť.

Vývoj v extrémnom programovaní sa líši od klasického programovania hlavne zautomatizovaným testovaním, párovým programovaním a ďalšími.

Základnou jednotkou extrémneho programovania je práve párové programovanie. Tento výraz pojednáva o tom, že kód programu tvoria dvojice programátorov za jedným počítačom, pričom sa dvojice obmieňajú, čo umožňuje rýchle šírenie znalostí v tíme.

Vývoj v extrémnom programovaní je riadený testami. Čo to presne znamená? Celý proces vytvárania kódu je rozdelený do malých cyklov a po každej iterácii sa hneď implementuje kód. Predtým než dvojice programátorov začnú programovať, tak vytvárajú testy, a ak nie sú všetky testy vytvorené nemôže sa začať s vytváraním zdrojového textu. Programovať sa začína len vtedy, ak sa už na žiadne testy nemôže prísť. Dvojice členov tímu nezaisťujú len to, aby testy fungovali ale rozvíjajú tiež návrh systému a implementáciu.

Stratégia vývoja v extrémnom programovaní začína pomocou iteračného plánovania. Nepretržitá integrácia znižuje vývojové kolízie a vytvára prirodzené zakončenie jednej vývojovej časti. Spoločné vlastníctvo povzbudzuje tím, aby sa zdokonaľoval a párové programovanie dáva celý proces do hromady [1].

Párové programovanie vs. menšie tímy

Jedným zo základných aspektov extrémneho programovania je práca v pároch alebo takzvané párové programovanie. Tento spôsob vývoja softvéru má veľa priaznivcov ale na druhej strane u mnohých programátorov býva zavrhaný. Prečo je tomu tak? Párové programovanie nie je pre každého, pretože nie každý sa dokáže stotožniť s tým, že sa naňho bude niekto vkuse pozerať a opravovať ho, či horšie, ak si niekto nesadne zo svojím kolegom ale aj na tieto problémy existuje v extrémnom programovaní riešenie.

Párové programovanie dáva veľký priestor pre zlepšenie schopností programátora a umožňuje rýchlejší vývoj aplikácie, pretože čas, ktorý programátor zaberie hľadaním

4 *Róbert Kocian*

riešenia a následným učením sa, je oveľa väčší ako čas, ktorý zaberie komunikáciou so skúsenejším kolegom.

Ja som párové programovanie využil viacej krát a vždy sa mi nadmieru vyplatilo ho použiť. Z mojich skúseností viem, že pri programovaní aplikácie v jazyku, v ktorom sú moje znalosti slabšie, nie je jednoduché nájsť riešenie, ktoré splňuje konvencie kvality daného jazyka. Veľa krát sa mi stalo, že som nemohol nájsť správne riešenie aj niekoľko hodín, a keď som ho našiel objavil sa iný problém a po ňom veľa ďalších, čo by samozrejme vyriešila bežná komunikácia, ale ak je problémov viacej a objavujú sa stále takmer po každom ďalšom vyriešení jedného problému, je podľa môjho názoru použitie párového programovania najlepším riešením pri malých projektoch a tímoch s veľkosťou od 2 do 10 členov. Táto metodika je veľmi vhodná pre malé programátorské tímy, ktoré vyvíjajú softvér s dynamickým zadaním. Vhodnosť pre malé tímy je určená aj tým, že neustála komunikácia, plánovanie stretnutí, či vkladanie kódu do systému po malých častiach a kontrolovanie týchto častí sa vykonáva efektívnejšie a prehľadnejšie v menších tímoch ako vo väčších. Pri veľmi malých tímoch do šiestich členov je výhodné programovať tak, aby boli všetky tri dvojice v jednej miestnosti, tým sa urýchlí komunikácia a zvýši sa kvalita už tým, že budú robiť lepšie rozhodnutia založené na viacerých skúsenostiach.

Samotná kvalita vyvíjaného kódu je tak zabezpečená už komunikáciou medzi dvomi ľuďmi, ktorý sa snažia analyzovať, navrhovať, testovať a spoločne pochopiť, ako programovať lepšie. Keď jeden programátor píše, druhý môže premýšľať o stratégii. Kam vedie ich smer vývoja? Ak je v páre jeden programátor, ktorý je o veľa skúsenejší, ako ten druhý bude sa neskúsený programátor určitú dobu učiť, pozeráť a menej písať. Časom však zachytí banálne chyby toho skúsenejšieho a bude ho opravovať.

Behom pár týždňov začne neskúsenejší programátor preberať vzory od skúsenejšieho a zistí odchýlky od týchto vzorov. Väčšinou po pár mesiacoch už nie je rozdiel medzi programátormi tak zistiteľný ako na začiatku.

Prečo vlastne párové programovanie funguje? Vďaka podpore komunikácie. Ak sa niekto z tímu naučí novú dôležitú informáciu tak sa rýchlo šíri do celého tímu pretože sa páry vymieňajú.

Ďalším výhodným prvkom, ktorý zabezpečuje kvalitu párového programovania je obmedzenie zlozvykov členov tímu. V strese sa ľudia vracajú k svojim zlozvykom. Vynechávajú písanie textu, odkladajú refaktorizáciu, vyhýbajú sa integrácii. Ak sa však kolega v tíme pozerá je pravdepodobnejšie, že ho opraví alebo navrhne lepší, rýchlejší algoritmus, čím sa zvýši kvalita a zároveň sa tak predchádza chybám pri nepozornosti [1].

Záver

V súčasnosti je kladený veľký dôraz na kvalitu a dokončenie projektu v krátkom čase. Je zrejmé, že kolektívne sa problémy zvládajú lepšie. Preto by bolo veľkým plytvaním prostriedkov nevyužiť výhody, ktoré poskytuje párové programovanie. Je veľmi efektívne hlavne pre menšie tímy, pričom zachováva vysokú kvalitu. Vzhľadom na efektivitu, kvalitu a množstvo iných aspektov vývoja softvéru je vhodnejšie vytvoriť dvojice z členov tímu, než každému prideliť jednu úlohu, aby na nej pracoval sám.

Použitá literatúra

1. Beck, K.: *Extreme Programming Explained*, Vol. 1, Addison-Wesley, 2001. ISBN: 0201616416.
2. Bieliková, M.: *Softérové inžinierstvo: Princípy a manažment*. FEI STU, Bratislava. 2000. Diel II.
3. KŘENA, B., KOČÍ, R.: *Úvod do softwarového inžinierství*. Fakulta informačných technológií VUT Brno, 2006.
4. Thomas S.: *Agile Quality Management*. Dostupné na internete: <http://itsadeliverything.com/agile-quality-management> [cit. 2011-november]

Annotation

Manifesto of pair programming in team

Requirements for faster software development and quality are getting higher and higher, so for programmers it is more and more challenging. Development of a high-quality and comprehensive application with continuously increasing requirements is extremely difficult task in this accelerating time so this situation requires the use of extreme development methods making this problem largely eliminated so it is possible to develop high-quality software in the shortest time. Appropriate use of pair programming depends on the size of the project as well as the size of the team. This leads to question. Is Pair programming a suitable method for small teams working on smaller projects? If so, is it possible to do pair programming in smaller teams in order to achieve the required criteria imposed on the project?