

PÁROVÉ PROGRAMOVANIE – IDEÁLNE IBA NA PAPIERI

*Nie je nič horšie, ako spolupráca s človekom, ktorý
počúva iba sám seba.*

Michal Fornádel'

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

Abstrakt. *Párové programovanie ako súčasť extrémneho programovania je založené na aktívnej spolupráci dvoch programátorov pracujúcich na rovnakej úlohe bok po boku. Z pohľadu zvýšenia efektivity a kvality je nesmierne dôležité identifikovať spoločne vykonávané aktivity a neprogramovať v páre po celú dobu. Odlišné charakterové vlastnosti, schopnosti dvojice a takisto aj typ projektu výrazne vplyvajú na to, či je párové programovanie vhodnou technikou pre použitie. Výhody párového programovania sú z teoretického hľadiska veľmi prínosné pre členov vývojového tímu ako aj pre manažérov, ktorí šetria finančnými prostriedkami a kvalitatívne zvyšujú úroveň projektu. Prax však ukázala, že množstvo programátorov má s touto technikou viac negatívne ako pozitívne skúsenosti spojené najmä s ťažkosťami ohľadom časovej synchronizácie dvojice ľudí, vzájomným neporozumením, vnútorným nepokojom, pocitom neustálej kontroly a stratou koncentrácie. Párové programovanie je vhodné najmä v rámci aktivít, ktoré vyžadujú diskusiu a rôzne pohľady na ich prípadne riešenie. Ostatné aktivity môžu z pohľadu vzájomného pôsobenia dvojice programátorov so sebou prinášať viacero nevýhod.*

Kľúčové slová: *extrémne programovanie, párové programovanie, miera spolupráce, charakteristiky ľudí, roly programátorov*

Úvod

Jednou z najviac obľúbených techník extrémneho programovania je párové programovanie. Počas posledných rokov sa táto technika javila ako veľmi perspektívna najmä pre vytváranie vysoko kvalitného softvéru v časovo prijateľných podmienkach a nie je preto prekvapením, že je používaná pri vývoji softvéru po celom svete.

Párové programovanie je spôsob, kde na rovnakej úlohe spolupracujú dvaja vývojári súčasne. Jedná sa predovšetkým o analýzu a návrh riešenia, implementáciu a testovanie [7]. Kooperácia nad takýmito činnosťami prináša výrazne zlepšenie kvality prevádzaných riešení a aj keď by sa na prvý pohľad mohlo zdať, že zdvojnásobenie počtu hodín korešpondujúcich ku danej úlohe je mrhaním času aj peňazí, opak je pravdou. Napriek tomu je ale nutné si uvedomiť, že efektívnosť párového programovania je podmienená chápaním, kedy a za akých okolností je vhodné túto techniku uviesť do praxe [4]. V prípade, že manažéri dôsledne nezvážia na základe zloženia vývojového tímu a charakteru projektu relevantnosť použitia, môže nastať zbytočné predraženie a narušenie celkového chodu projektu.

Roly dvojice participujúcich programátorov

Pri párovom programovaní vystupujú obaja programátori ako jeden fungujúci celok, ktorý je zodpovedný za každý aspekt artefaktu, na ktorom pracujú. Existujú pritom viaceré pohľady na rozdelenie rolí.

Publikácie [1,7] uvažujú prístup, v ktorom prvý z dvojice programátorov píše programový kód, zatiaľ čo druhý premýšľa o súvislostiach z hľadiska optimálnosti a vhodnosti zvoleného riešenia. Takéto ponímanie vyžaduje hlbšie odborné znalosti, pretože systém je posudzovaný skôr z abstraktnejšej úrovne pohľadu.

Druhý, z môjho pohľadu reálnejší pohľad, je upustenie od koncepcie dvoch odlišných rolí. Štúdie [2,3] potvrdzujú, že v praxi pracuje dvojica skôr na jednej úrovni abstrakcie, pričom konkrétna úroveň závisí od druhu vykonávanej úlohy. Takýto prístup prispieva k tomu, že dvojica rieši problém na rovnakej úrovni detailov, čo prispieva k urýchleniu dosiahnutia požadovaného výsledku.

Základným kameňom úrazu je z môjho pohľadu tendencia pravidelného vymieňania si rolí počas práce. V spomínaných publikáciách sa autori príliš nezaoberajú dôsledkami výmeny programátorov na kvalitu a spôsob vypracovania požadovanej úlohy, avšak osobne si myslím, že vymieňanie si rolí je podmienené dostatočnou odbornou aj osobnostnou kvalifikáciou oboch programátorov. Za tohto predpokladu už následný charakter rolí vyplynie z konkrétnych znalostí a skúseností participujúcich ľudí, pričom posudzované sú tri základné varianty rozloženia schopností dvojice.

Prvým variantom je prípad, kedy sú obaja programátori na porovnateľnej vysokej úrovni. Pochopenie základných aspektov správnej architektúry a fungovania vybratého systému, ako aj písanie programového kódu pre nich nepredstavuje problém a eventuálna výmena rolí je vítaná. Čo som v práci však postrehol je fakt, že príliš vysoká kvalifikácia znižuje mieru kolaborácie a tým negatívne pôsobí na atmosféru v tíme.

Druhým prípadom je dvojica skúseného a menej skúseného programátora. Na základe vlastných skúseností si myslím, že táto kombinácia predstavuje najlepšie riešenie,

pretože každý z dvojice programátorov prispieva ku výsledku odlišným spôsobom, čoho dôsledkom je zvýšená miera spolupráce, distribúcie poznatkov a tímového ducha. Skúsenejší člen dohliada na kvalitu a spôsob vykonávaného riešenia a usmerňuje programátora tvoriaceho programový kód. Ten, ak by aj v najhoršom prípade nemal svojou odbornosťou okrem zápisu kódu ako prispieť ku dosiahnutiu riešenia, môže logickým uvažovaním nad vykonávanými činnosťami v teoretickej rovine priviesť skúsenejšieho člena k lepšiemu riešeniu. Výmena rolí však v tomto prípade nie je vhodnou voľbou. Z dlhodobého hľadiska napriek tomu dochádza ku vzájomnému obohacovaniu sa programátorov a aj ku odbornému rastu menej skúsených členov tímu.

Uvažovať nad párovým programovaním a rolami, ak by ani jeden z členov nemal dostatočnú odbornosť, je prakticky nemožné, pretože by to neprospievalo ani kvalite softvérového produktu, ani programátorom samotným.

Miera kooperácie dvojice programátorov

Pôvodná myšlienka párového programovania prezentovaná v publikácii [7] bola založená na nepretržitej spolupráci dvoch programátorov bez ohľadu na to, či sa jedná o analýzu, návrh, implementáciu alebo testovanie. Teoreticky by takáto symbióza mohla prinášať svoje ovocie, obávam sa však, že realita leží niekde úplne inde.

Predstava úplnej synchronizácie časového harmonogramu dvoch ľudí je možná azda iba z veľmi krátkodobého pohľadu. Dvaja ľudia prichádzajú do práce v rovnakom čase, obedujú v rovnakom čase, odchádzajú v rovnakom čase. Možno to niekde funguje, ale moje skúsenosti ma postupom času len utvrdili v tom, že základom úspechu je najšť a identifikovať činnosti, ktoré je nutné vykonávať v páre práve kvôli efektívnosti a kvalite.

Zaujímavý pohľad prinášajú autori publikácie [5]. Tí tvrdia, že kritickými činnosťami pre potenciálny úspech projektu sú analýza a návrh riešenia. Tieto aktivity totižto vyžadujú zhodnotenia viacerých alternatív a nemalú kreativitu a znalosti. Práve preto uvažovanie dvoch ľudí výrazne eliminuje riziko zlej koncepcie navrhovaného riešenia. Ostatné činnosti sú potom na zvážení, či je potreba ich vykonávania v páre alebo samostatne.

Nie som celkom stotožnený s tým, že vynechanie kooperácie pri implementácii je prospešné. Písanie programového kódu je pomerne náchylné na chyby. Ak si odmyslím, že syntaktické chyby zachytáva kompilátor, mnohé zlyhania sú spôsobované volaním nesprávnych metód, poprípade nekorektným spôsobom používania už implementovanej funkcionality. Z toho dôvodu sa človek, ktorého pozornosť nie je sústredená na písanie na klávesnici môže plne koncentrovať na dodatočnú kontrolu. Z dlhodobého hľadiska sa dá výrazne predísť zbytočným zlyhaniam spôsobených nepozornosťou ľudského faktoru neodhliadajúc od faktu, že kód, ktorý prešiel kontrolou dvoch programátorov je braný ako kvalitnejší. Napriek tomu existujú implementačné úlohy, ako napríklad návrh grafického používateľského rozhrania alebo zmeny názvoslovia, ktoré si skutočne nevyžadujú pozornosť dvoch ľudí. Z tohto pohľadu je to skôr výnimka, ktorá potvrdzuje pravidlo.

Je tu ešte jedna vec, ktorá podľa môjho uváženia výrazne vplýva na efektívnosť párového programovania, a to je charakter vykonávanej úlohy a prakticky aj celého projektu. Posledné dva roky pracujem na projekte, ktorý sa skladá z viacerých druhov aktivít, ako napríklad dorábanie existujúcej funkcionality, opravovanie chýb

reportovaných zákazníkom a testovanie prevedených riešení. Pomer kreatívnych činností, ktoré si neodmysliteľne vyžadujú párové zhodnotenie a činností, ktoré by sa eventuálne dali vykonávať aj samostatne, je výrazne naklonený ku druhej variante. Manažment firmy od začiatku preferoval koncepciu práce v pároch. Ako sa ukázalo, efektivita programovania vo dvojici bola dosť kontraproduktívna v situáciách, ktoré spočívali v prehľadávaní existujúceho kódu a úprave zopár riadkov za účelom odstránenia kozmetickkej chyby alebo testovaniu funkcionality podľa fixne zadefinovaných krokov.

Postupom času som dospel k presvedčeniu, že dlhodobé projekty strácajú na množstve kreatívnych úloh a pomerne jednotvárne činnosti vykonávané ľuďmi pracujúcimi na projekte od začiatku prinášajú väčšiu efektivitu pri práci osamote, ktorá je doplnená párovým programovaním iba v prípade nutnosti.

Problémy charakterov ľudí

Párové programovanie je veľmi jednoduchý koncept – dvaja programátori spolu bok po boku pracujú na jednej úlohe a prispievajú vzájomnou komunikáciou ku dosiahnutiu výsledku. Na tomto tvrdení by nebolo nič zvláštne, keby sa nejednalo o ľudí.

Napriek tomu, že je párové programovanie kvôli efektívnej výmene znalostí a zvýšenej miere kolaborácie vo všeobecnosti označované ako priaznivé a pomerne bezproblémové [6,8], množstvo programátorov vrátane mňa zaznamenalo aj negatívne aspekty tejto techniky. Charakterové vlastnosti a návyky ľudí vedia predstavovať veľké prekážky, ktoré do značnej miery zabraňujú fungovaniu vzájomnej spolupráce, a to hlavne z dlhodobého hľadiska. Autori spomínaných publikácií vyjadrujú názor, že ochota poznať a riešiť konflikty sú základnou súčasťou väčšiny vývojových tímov preferujúcich koncepciu párového programovania. Mój názor leží niekde úplne inde.

Prvým problémom je ľudské ego, ktoré je často buď príliš veľké, alebo na druhú stranu príliš malé. Počas posledných dvoch rokov som zaznamenal skúsenosti s oboma variantmi, pričom asi pravdepodobnejším je natrafenie na tzv. programátorov hrdinov, ktorí chcú všetko spraviť sami a sú presvedčení, že oni to dokážu najlepšie. Privysoké ego je doménou skúsenejších programátorov, ktorí majú za sebou už niekoľko úspechov a za záhadných okolností dospeli k presvedčeniu, že majú počúvať iba samých seba a v minimálnej miere rešpektovať názory ostatných. Práca s takýmto človekom je dosť problematická či už z pozície za počítačom alebo aj mimo neho. Programátor hrdina skrátka nespolupracuje a tým porušuje základnú myšlienku párového programovania. Riešením je buď vytvorenie dvojice navzájom sa rešpektujúcich rovnako geniálnych programátorov hrdinov alebo odsúdenie problematických typov na prácu osamote.

Malé ego je doménou menej skúsenejších a začínajúcich programátorov, ktorí sa mnohokrát strádajú aktívnej spolupráce z obavy, že by ich návrhy mohli byť okamžite zamietnuté a vyšli by na posmech. K tejto chybnej domnienke prichádzajú v začiatkoch svojho pôsobenia najmä pri spolupráci s programátormi hrdinami, ktorí ich nevhodnými poznámkami a náznakmi ignorácie odstavujú na vedľajšiu koľaj. Riešením takejto situácie je okamžité oddelenie dvojice a vytvorenie charakterovo príbuznejšieho variantu.

Napriek tomu aj spolupráca dvoch kooperujúcich ľudí je pomerne náročná, a to hlavne z dlhodobého hľadiska. Mnohokrát som si postupom času pri práci za klávesnicou a kolegom za chrbtom pripadal oveľa nervóznejší a menej produktívny. Možno je to

záležitosť povahy človeka, ale ku koncu dňa ma to už unavovalo byť neustále bombardovaný múdrosťami a pripomienkami o chybách, ktoré som aj všimol, ale nestihol som ich dostatočne rýchlo opraviť, čo mi bolo hneď vykázané. Netvrdím, že toto pokladám za negatívum párového programovania, skôr chybu osobného prístupu. Môj názor je však taký, že človek potrebuje na produktívnu prácu veľké sústredenie a práca pod dohľadom nemusí každému vyhovovať. Riešením konfliktov podobného tímu je časté preskupovanie dvojíc programátorov za účelom eliminácie takpovediac ponorkovej nemoci.

S tvrdením, že predísť vonkajším aj vnútorným konfliktom sa dá jednoducho dôslednou analýzou zloženia tímu ešte pred jeho začiatkom [2] súhlasím čiastočne, pretože analýza ľudskej povahovej charakteristiky je možná iba po uplynutí dostatočne dlhkej doby, počas ktorej manažér identifikuje vlastnosti svojich pracovníkov. Ak by teda vývojový tím nového projektu tvorili existujúci dlhodobí zamestnanci, je v kompetencii vedúceho koncepcie poskladať tím vhodný pre párové programovanie, ak je to zámerom spoločnosti. Základnými podmienkami by mala byť rovnováha medzi skúsenými a menej skúsenými programátormi a voľba priemerne dominantných, nekonfliktných a chápaných ľudí. Ak by však súčasťou tímu mali byť čerstvo prijatí ľudia, je azda nemožné iba na základe prijímacieho pohovoru správne identifikovať ich skutočné povahové črty.

Záver

Zvládnutie párového programovania a použitie tejto techniky v rámci projektu sa na prvý pohľad môže zdať pomerne jednoduché, nie je tomu však tak. Je potrebné jednoznačne zadefinovať kedy a za akých okolností.

Začínajúce projekty majú výhodu, že za predpokladu vytvárania tímu z existujúcich zamestnancov je možné výber členov podmieniť charakterovými vlastnosťami potrebnými pre prácu vo dvojici. Dvojice vytvára vedúci projektu za účelom dosiahnutia rovnováhy medzi skúsenými a menej skúsenými programátormi. Zloženie dvojíc nie je z dlhodobého hľadiska kritické, pretože sa menia na pravidelnej báze práve za účelom odstránenia konfliktov medzi jednotlivými členmi.

Pri spoločnom vykonávaní úloh je vhodné pre oboch programátorov pracovať na jednej úrovni abstrakcie. Takýto prístup viac združuje spoluprácu a prináša množstvo výhod v podobe zvýšenia miery spolupráce a distribúcie poznatkov v porovnaní s tým, ako keď človek za klávesnicou pracuje v konkrétnej technickej rovine, pričom druhý člen poníma úlohu abstraktnejšie a konceptuálnejšie.

Miera času stráveného spoločne dvojicou programátorov za jedným počítačom závisí od druhu vykonávanej činnosti. Najvhodnejšie činnosti sú analýza a návrh riešenia, ktoré si vyžadujú kreatívne myslenie a zváženie viacerých alternatív riešenia. Implementácia až na pár výnimiek tiež spadá do kategórie vhodnej pre pozornosť oboch členov, pretože duplicitnou kontrolou písaného kódu sa dá jednoducho predísť zbytočným chybám, ktoré by časom aj tak vyšli na povrch. Činnosti ako testovanie alebo oprava reportovaných kozmetických chýb môžu byť vykonávané členmi samostatne. To sa týka aj pravidelne opakujúcich sa činností v dlhodobých projektoch, kde sa miera kooperácie postupne vytráca.

Párové programovanie je vhodné pre projekty, v rámci ktorých bude implementovaná nová funkcionálna náchylná na úroveň a prevedenie riešenia. Pre

udržiavanie a správu zabehnutých a prakticky funkčných systémov nie je nutnosťou použitie dvoch programátorov, pretože by to znamenalo skôr plytvanie finančných aj ľudských zdrojov.

Použitá literatúra

1. Beck, K.: *Extreme programming explained: embrace change*, Addison-Wesley Longman Publishing Co., 2000.
2. Bryant, S., Romero, P., du Boulay, B.: *Pair Programming and the Mysterious Role of the Navigator*, International Journal of Human-Computer Studies (in Press), Vol. 66, 2008.
3. Chong, J., Hurlbutt, T.: *The Social Dynamics of Pair Programming*, Software Engineering ICSE 29th International Conference, 2007.
4. Chong, J., Plummer, R., Klemmer, S. R., Eris, O., Tove, G.: *Pair Programming: When and Why it Works*, 17th Workshop of the Psychology of Programming Interest Group, Sussex University, 2005.
5. Flor, N.V., Hutchins, E.L.: *Analyzing Distributed Cognition in Software Teams: A Case Study of Team Programming during Perfective Software Maintenance*, Proc. Empirical Studies of Programmers: Fourth Workshop, Ablex Publishing, New York, 1991.
6. Nosek, J.T., *The Case for Collaborative Programming*, Communications of the ACM, Vol. 41, 1998
7. Williams, L., Kessler, R.: *Pair programming illuminated*, Addison-Wesley Longman Publishing Co., 2002.
8. Williams, L., Kessler, R.: *Strengthening the Case for Pair-Programming*, IEEE Software, Vol. 17, 2000.

Annotation

Pair programming – ideal only theoretically

Pair programming as a part of extreme programming is based on active cooperation of two programmers working on the same task side by side. Regarding the increase of efficiency and the quality, it is absolutely significant to identify mutual activities and refuse to program in pair all the time. Different character qualities, abilities of programmers and the type of project are relevant for considering the fact if pair programming is suitable technique for usage. From the theoretical point of view, advantages of pair programming are very beneficial for both members of development team and managers who reduce financial resources and qualitatively increase the level of project. Unfortunately, the practical experiences have shown that many programmers are a bit skeptical because of difficulty according to the time synchronization of pair, mutual misunderstanding, personal discomfort, feeling of endless control and loss of concentration. Pair programming is suitable for activities demanding further investigation and different conceptual views. Other activities from the programmer's point of view could bring some disadvantages.