

MÔŽE BYŤ 2 > 1+1?

Viac hláv vie viac.

Tomáš Zboja

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

Abstrakt. *Párové programovanie, známe z extrémneho programovania, bolo skúmané vo viacerých oblastiach. Táto esej má za cieľ pozrieť sa na párové programovanie z pohľadu vplyvu na kvalitu vyvíjaného produktu. Identifikuje oblasti kvality softvéru a na tieto sa neskôr zameriava. Ide predovšetkým o štandardy pri tvorbe softvéru, aj z nich vyplývajúca bezpečnosť, by mala byť neodmysliteľnou súčasťou požiadaviek. Veľká časť sa venuje chybovosti produktu – chybám vytvoreným počas fázy implementácie. Niektoré zdroje výskumu uvádzajú, že párové programovanie má negatívny vplyv na kvalitu softvéru. Väčšina je však opačného názoru. Esej obsahuje zamyslenie sa, či je správne meranie uvádzajúce negatívny vplyv. Aj keď všeobecne prijateľný výsledok je, že tento typ programovania je vhodný z pohľadu kvality, zamýšľa sa nad typmi softvéru, ktoré sú vhodné pre tvorbu párovým programovaním. Taktiež uvádza podmienky, ktoré by mali pomôcť k dosiahnutiu nameraných výsledkov – zlepšenej kvalite.*

Kľúčové slová: *programovanie v páre, kvalita softvéru, agilné programovanie, extrémne programovanie, párové programovanie.*

Úvod

Párové programovanie je spôsob programovania typický pre extrémne programovanie. Prebieha vo dvojici za jedným počítačom. Jeden z dvojice programuje (vodič, angl. „driver“), druhý medzi tým kontroluje vytváraný kód (navigátor, angl. „navigator“). Tieto roly si účastníci pravidelne menia.

Existuje viacero názorov na tento spôsob programovania. Taktiež existuje viacero uhlov pohľadu – napr. čas, kvalita, komunikácia v tíme, náklady na „plytvanie ľuďmi“.

2 Tomáš Zboja

Veľa ľudí vie z vlastnej skúsenosti, či už školy, vlastných projektov alebo zamestnania, že tvoriť v tíme je ľahšie. Vzniká síce réžia s dohadovaním sa – jednotlivец by sa sám rozhodol, alebo s nevyužívaním času na samotnú prácu – nechcená komunikácia, zábava. Aj napriek tomu si myslím, že na kvalitu produktu má takýto spôsob pozitívny vplyv. Spoločné zdieľanie vedomostí a nápadov posúva kód, a teda aj celý produkt do iných úrovní.

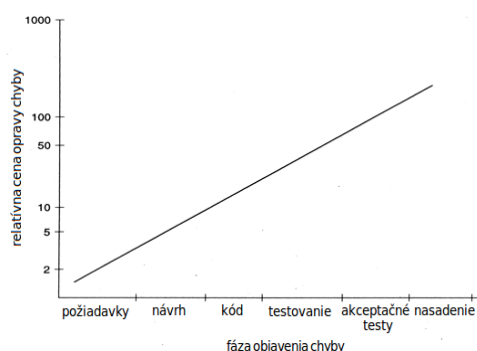
Párové programovanie má vplyv aj na kvalitu softvéru. Dotýka sa jej vo viacerých oblastiach. Medzi najdôležitejšie patria:

- bezpečnosť,
- chybovosť,
- štandardy zdrojového kódu,
- čitateľnosť – udržiavateľnosť.

Štandardy

V tíme, ktorý vyvíja softvér, by mali platiť bežne uznávané alebo interné štandardy pre tvorbu kódu. Tieto majú pozitívny vplyv na kvalitu kódu – stáva sa čitateľnejším (v porovnaní s osobnými štandardmi každého člena) – ako v prípade rozšírenia, tak v prípade hľadania a opravy chýb. Predovšetkým noví členovia tímu alebo členovia novovytvoreného tímu by s týmito mohli mať problém. Párové programovanie napomáha k osvojeniu si týchto štandardov, a teda zvýšeniu kvality kódu. Keď nový člen tímu píše kód, skúsenejší člen ho napraviť. Keď skúsenejší píše kód, novší vidí ako dodržiavať tieto štandardy.

Vzhľadom na kontrolu kódu počas jeho písania vzniká menej chýb. Čo programátor prehľadne, si veľmi pravdepodobne všimne navigátor. Menej chýb znamená, že je menej času potrebného na ich opravu v neskorších fázach – ďalšej implementácii, testovaní a hlavne údržbe. To isté platí o nákladoch na neskoršie opravy chýb – ako vidíme na obrázku (obr. 1).



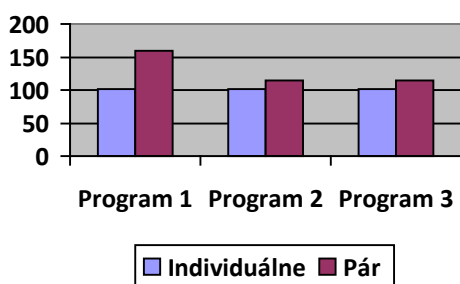
Obr. 1. Náklady na opravu chyby vo fázach vývoja¹.

¹Obrázok upravený

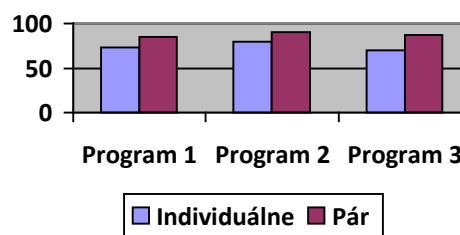
z <http://www.cs.colorado.edu/~kena/classes/5828/s07/lectures/04/EarlyDefectRemoval.png>.

Merania párového programovania

Intuitívny predpoklad je, že párové programovanie spotrebuje dvakrát viac času ako programovanie individuálne, a preto nie je vhodné na tvorbu softvéru. Podľa štúdie [3] to nie je pravda. Fakt je, že párové programovanie je časovo náročnejšie, ale iba o 15% (viď obr. 2). Z pohľadu kvality [3] je však výhodné, ako vidieť na obrázku 3. Počet úspešne vykonaných testov stúpol približne o 15%. Pri pohľade na graf však možno povedať, že počet neúspešne vykonaných testov klesol o približne 50%.



Obr. 2. Čas na vytvorenie produktu [3].



Obr. 3. Úspešné testy [3].

Tabuľka z inej štúdie [4] tiež poukazuje na vyššiu kvalitu (väčší počet úspešných testov) v prípade párového programovania.

Tab. 1. Merania úspešne absolvovaných testov v štúdii [4].

	Jednotlivci	Dvojice
Program 1	73,40%	86,40%
Program 2	78,10%	88,60%
Program 3	70,40%	87,10%
Program 4	78,10%	94,40%

Podľa zdroju [1] párové programovanie nie je také výhodné z pohľadu chybovosti. Obrázok 4 ukazuje odchýlky v kóde od párom definovaných štandardov pre tvorbu kódu. Ide hlavne o:

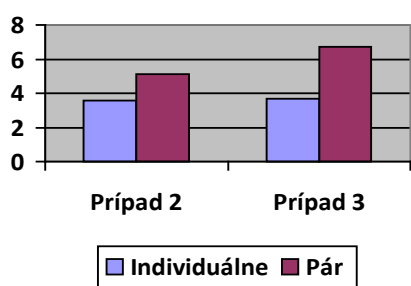
- nedodržanie štandardov,
- pomenúvanie premenných,
- komentáre metód.

Nie som si úplne istý skutočnou výpovednou hodnotou tohto merania. V štúdii sa píše, že sa dohodli štandardy, ktoré sa majú používať. Otázna je podľa mňa miera, do akej boli tieto štandardy odlišné od bežne používaných pre daných ľudí, a zároveň aj dĺžka trvania týchto pokusov. Ak sa každý programátor mal učiť nové štandardy, na ktoré nie je zvyknutý a používať ich nedostatočne dlhý čas, myslím si, že je samozrejmé prečo dvojica

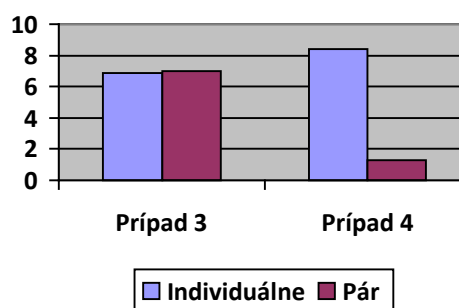
urobila viac chýb. Konflikt dvoch rôznych spôsobov programovania sa ťažšie ustriečne ako jeden samotný.

V párovom programovaní často pracujú spolu ľudia, kde jeden je skúsenejší, a tak odhalenie uvedených problémov pre neho nie je náročné. Zároveň on sám sa dopúšťa menšieho počtu daných chýb.

Rovnaký zdroj zároveň udáva, že pri párovom programovaní vznikne podľa inej metriky menej chýb ako pri klasickom spôsobe (obr. 5) [1]. Meraný bol počet chýb na štýl programovania a počet vytvorených riadkov daným štýlom. V prípade č. 3² bol počet všetkých chýb trikrát väčší ako v prípade č. 4, a preto je mu v štúdiu pripisovaný väčší význam.



Obr. 4. Odchýlky od definovaných štandardov na 100 riadkov kódu (zdroj [1]).



Obr. 5. Počet chýb na 1000 logických riadkov kódu (zdroj [1]).

Podľa môjho názoru by sa všetky uvedené merania mali vyskúšať na väčších vzorkách, aby sa dali vytvoriť štatisticky prijateľnejšie výsledky. Meranie 2-3 projektov považujem za nedostatočné na vytvorenie záveru o pozitívnom alebo negatívnom prínose párového programovania, aj keď podľa uvádzaných výsledkov predovšetkým toho negatívneho.

Ako poukazujú zdroje, párové programovanie je vhodné predovšetkým pre programátorov začiatočníkov („juniorov“) [1,2]. Čím je programátor skúsenejší, tým menší vplyv na kvalitu má párové programovanie. Pri programátoroch junioroch stúpila správnosť kódu až o 73%, pri pokročilých o 4% a pri skúsených („senioroch“) klesla o 8%. Tieto informácie pochádzajú z výskumu [2], ktorého sa zúčastnilo 99 individuálnych programátorov a 98 párov. Uvedené štatistiky môžu byť ovplyvnené tým, že programátori začiatočníci nie sú natoľko zvyknutí na klasický spôsob programovania. A taktiež sa domnievam, že si skôr pripustia nápad svojho spolupracovníka, nakoľko ich skúsenosti nie sú na úrovni profesionálov a môžu predpokladať, že spolupracovník vie o danom probléme viac.

Zdroje [1,3] naznačujú že párové programovanie je vhodné na učenie. Ako dôvody udáva [3] vzájomnú komunikáciu, zdieľanie vedomostí a schopnosť riešiť problémy, ktoré by programátori samostatne nevyriešili. V [1] sa aj priamo píše, že programátori považovali párové programovanie za najvhodnejšie na riešenie zložitých úloh, učenie sa na začiatku (ako programátori začiatočníci) a hľadanie chýb v jednoduchých programoch.

² Slovom prípad sa v tomto texte označuje konkrétny projekt z merania z uvedeného zdroja [1] rovnako ako v zdrojovom dokumente.

Zaujímavé výsledky by mohol priniesť výskum, ktorý by porovnával kvalitu skúsených programátorov zvyknutých na klasický spôsob tvorby softvéru a skúsených programátorov, ktorí od začiatku svojej programátorskej kariéry tvorili produkty spôsobom párového programovania.

Použiť párové programovanie považujem za najlepšie v projektoch, ktoré nevyžadujú finančné zdroje zákazníka a prísne časové termíny. Tento záver vychádza z faktu, že párové programovanie zvyšuje kvalitu, ale v konečnom výsledku môže trvať dlhšie ako bežné programovanie. Ako vhodné typy projektov sa teda javia projekty s otvoreným zdrojovým kódom a voľnočasové projekty, pretože väčšinou spĺňajú hore uvedené požiadavky o čase a financiách. Zároveň požadujú alebo očakávajú vysokú kvalitu, ktorá je podľa prieskumov párovým programovaním dosiahnutá.

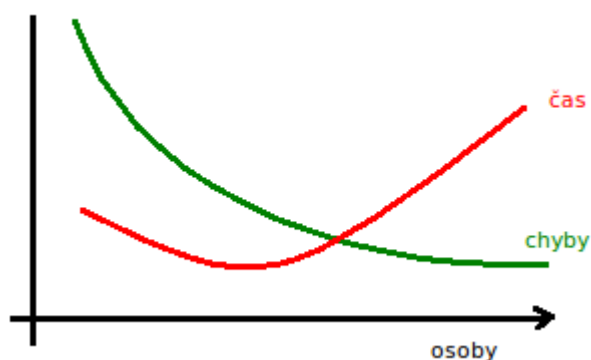
Iné pozitíva párového programovania

Každý programátor disponuje svojimi programátorskými zlozvykmi:

- zabúdanie komentovania dôležitých častí kódu,
- využívanie menej efektívnych konštrukcií či príkazov,
- zabúdanie na ošetrovanie výnimiek alebo
- nejasné pomenovanie premenných.

Tieto zlozvyky sú do značnej miery odstraňované druhým z dvojice, ktorý ich nenadobudol ako programátor začiatočník.

V prípade správnej komunikácie a porozumenia v tíme by so zvyšujúcim sa počtom ľudí mala narastať kvalita kódu (znižovať sa počet chýb v kóde), zároveň však od istého okamihu zvyšovať sa čas potrebný na ukončenie projektu – kvôli réžii s komunikáciou (obr. 6). Vidíme tu, že pre dvojicu klesne počet chýb ale aj čas potrebný na vyriešenie problému. Keby sme k dvojici pridávali ďalších ľudí, počet chýb bude klesať vďaka skúsenostiam, avšak čas potrebný na ukončenie projektu začne postupne stúpať, práve kvôli komunikácii. Párové programovanie je teda niekde v intervale najvýhodnejších spôsobov. Zvýšená kvalita v prijateľnom čase.

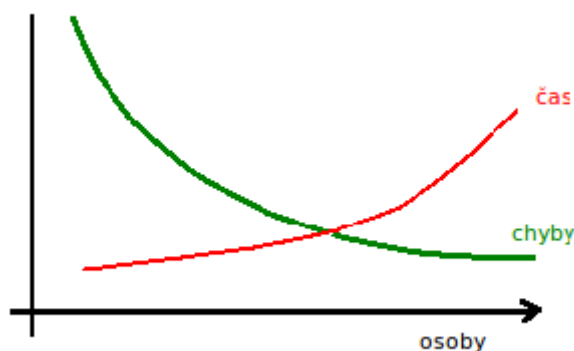


Obr. 6. Odhad vývoja času a kvality s nárastom počtu osôb v tíme.

Tento odhad je založený na vlastnej skúsenosti s riešením projektu samostatne, vo dvojici v rámci niektorých predmetov na univerzite a taktiež vo väčšom tíme na univerzite, kde samotná komunikácia a vyjasňovanie si rôznych problémov zabralo viac času, ako by bolo žiaduce.

Ide však len o odhad, pretože neboli k nemu zbierané exaktné údaje či vedené štatistiky. Preto by bol potrebný výskum, ktorý by preukázal alebo vyvrátil tento predpoklad.

Zohľadnením údajov zo zdrojov [1,3] možno upraviť graf, ako vidieť na obrázku 7.



Obr. 7. Odhad vývoja času a kvality s nárastom počtu osôb v tíme so zohľadnením uvedených meraní.

Záver

Myslím, že bez ohľadu na akýkoľvek výskum nie je možné povedať: „Párové programovanie je lepšie.“ alebo „Párové programovanie je horšie.“ Keďže produkt je výsledkom práce ľudí, väčšia časť záleží na nich. Som presvedčený o tom, že existujú páry, ktoré vyprodukovávajú kvalitnejší produkt spolu, a jedinci ktorí tvoria kvalitnejšie samostatne.

Predpokladám, že pre zvýšenie kvality vďaka párovému programovaniu je potrebných niekoľko podmienok.

Prvou z nich je vzájomné rešpektovanie sa. To je dôležité preto, aby si ani jeden z dvojice nemyslel, že je lepší, a aby si ani jeden z dvojice nemal pocit, že je menejcenný. Taktiež by takýto postup pravdepodobne viedol k prílišnému napĺňaniu ideí dominantného programátora, čím by samozrejme trpela aj kvalita.

Ďalším dôležitým faktorom je čas potrebný na prechod z klasického spôsobu programovania na párové. Považujem za prakticky nemožné, aby hneď po zmene spôsobu programovania toto fungovalo správne. Ako však výskumy ukázali, v konečnom dôsledku je párové programovanie výhodné. Ako programátori samotní, tak i zákazník by si mal uvedomovať, že v budúcnosti táto zmena bude viesť k zvýšeniu kvality produktu.

Posledným predpokladom úspešného párového programovania je, že sa členovia dvojice budú správať profesionálne. Tým myslím, že sa budú venovať programovaniu a nie zábave, čo sa často stáva.

Použitá literatura

1. Abrahamsson, P., Hulkko, H.: A Multiple Case Study on the Impact of Pair Programming on Product Quality, *ICSE '05 Proceedings of the 27th international conference on Software engineering*, New York (2005), 495-504.
2. Arisholm, E., Gallis, H., Dybå, T., Sjøberg, D.I.K.: Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise, *IEEE Transactions on software engineering*, Vol. 33, No. 2 (2007), 65-86.
3. Cockburn, A., Williams, L.: The Costs and Benefits of Pair Programming, *Extreme programming examined*, Addison-Wesley Longman Publishing Co., Inc. Boston (2001), 223-243.
4. Williams, L., Kessler, R. R., Cunningham, W., Jeffries R.: Strengthening the Case for Pair-Programming, *IEEE Software*, Vol. 17, North Carolina (2000), 19-25.

Annotation

Can be 2 > 1+1?

Pair programming, known from extreme programming, was researched in various fields. This essay aims to look at pair programming how it relates to the quality of developed product. It identifies fields of quality of software, and later focuses on them. These are primarily standards during software development, the consequent quality from them, which should be part of requirements. A large part is dedicated to product errors – errors created during the implementation phase. Some sources of investigation indicate that pair programming has a negative impact on software's quality. The majority thinks the opposite, though. The essay contains a reflection, whether the measurement that indicates the negative impact is correct. Although, generally, the acceptable result is, that this kind of programming is appropriate in terms of quality, there's a reflection on which kinds of software are the most suitable to be created with pair programming. It also states the conditions that would help to achieve the measured result - improved quality.