

Chyby a manažment softvérových projektov

TOMÁŠ VANDERKA

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
tomas.vanderka@ynet.sk*

Abstrakt. Všetci dobre vedia že ľudia robia chyby, robia ich neustále, ale nik nevie dopredu povedať s určitosťou kedy a aké. Už niekoľko desaťročí ľudia vyvíjajú techniky na zredukovanie chýb a zvýšenie efektivity vývoja softvéru. Napriek tomu softvérové projekty stále zlyhávajú, aj napriek pokrokom v technológii. Tento článok sa snaží v krátkosti popísať a vysvetliť niekoľko dôvodov prečo sa tak deje, a načrtnúť čo môžeme robiť aby sme tieto problémy zvládli.

Úvod

Softvérové inžinierstvo je relatívne nová oblasť, aspoň v porovnaní s inými oblasťami. Pri vývoji softvéru ide zväčša o tímové úsilie vytvoriť niečo podľa špecifických potrieb zákazníka. Na celý tím sú vytvárané tlaky, či už zvnútra alebo zvonku spoločnosti pre ktorú pracujú, ale aj úplne nezávislé vplyvy ako rodinné problémy, a množstvo ďalších. Všetky tieto tlaky prispievajú k vzniku chýb v riadení, rozhodovaní, programovaní atď. Je nemožné brať do úvahy všetky tieto vplyvy pri manažovaní projektu, keď ich ani nedokážete všetky vymenovať.

Napriek tomu všetkému, existuje veľké množstvo prístupov k manažovaniu takýchto projektov, ktoré znižujú riziko problémov. Žiadny z nich ale výskytu chýb nemôže zabrániť úplne. Pretože nech sa použije ktorýkoľvek z nich, vždy existuje šanca že niekto spraví chybný krok. Jediné čo môžete spraviť je snažiť sa znížiť jeho dopady, poučiť sa z neho a pokračovať ďalej.

Preto sa nebudem venovať konkrétnym metódam manažovania projektov, ale skôr bežným chybám, ktoré môžeme spraviť nezávisle od použitej techniky. Budem sa snažiť popísať dôvody ich vzniku, ich dopady a poskytnúť rady ako sa ich vyvarovať.

Najbežnejšie chyby

Nasledujúci zoznam zhrňa najbežnejšie chyby, ktorých sa môžete dopustiť pri manažovaní softvérového projektu. Ich poradie nijak neurčuje ich závažnosť.

Chyby a manažment softvérových projektov, január 2006, s. 77-81.

- Vytýčenie nerealistických cieľov
- Zlé odhady potrebných zdrojov
- Negatívna motivácia v prípade tiesne
- Nedostatočné pochopenie požiadaviek
- Slabé sledovanie pokroku práce
- Slabá komunikácia
- Použitie neoverenej technológie
- Podľahnutie komerčným tlakom
- Nedostatočné testovanie

Ku každej spomenutej chybe popíšem v krátkosti jej následky a možné riešenie.

Prehnané ciele projektu

V rámci komunikácie obchodníkov so zákazníkom sa môže ľahko stať že sa ponúkaný systém zvrhne na kolos, ktorý bude riešiť všetky používateľove najtajnejšie sny. To neskôr ovplyvní aj jeho pohľad na riešenie, ktoré v skutočnosti potrebuje. Projektový manažér by si mal dať pozor aby projekt ostal v hraniciach možností. Ak sa rozhodne implementovať projekt, na ktorý nemá dostatok prostriedkov tak ho nedokončí ani keby na to mal nekonečno času.

Zlé odhady prostriedkov a času

Odhady sú samé o sebe zložitá vec. Nemôžem povedať čo je dobrý a čo zlý odhad, pretože až na konci sa ukáže či bol dobrý alebo nie. Odhady môže robiť iba skúsený manažér, ktorý už má za sebou viacero podobných projektov. Nikdy by sme nemali strieľať odhady len tak z hlavy, mali by byť vždy podložené nejakými reálnymi skúsenosťami.

Najdôležitejšie ale je tieto odhady neskracovať podľa želania zákazníka, alebo iných nadriadených. Takto sa nedá nič získať, projekt bude trvať vždy rovnako dlho. Dokonca ak je odhad príliš mimo môže projekt trvať ešte dlhšie, pretože neskôr nebude možnosť dynamicky zmeniť plán a pribrať nových ľudí. O tom hovorí aj staré známe pravidlo od Brooksa [1], ktoré hovorí že to spomalí projekt ešte viac.

Ďalej je vždy lepšie použiť pesimistické odhady ako tie optimistické, pretože v prípade optimistických sa môžeme dostať iba do negatívneho sklzu.

Negatívna motivácia

Niektorí manažéri majú pod tlakom tendenciu prenášať tento tlak na ostatných členov tímu. Väčšina ľudí ale na tlak reaguje negatívne, hlavne ak ide o kreatívnu prácu akou tvorba softvéru určite je. Takýto tlak iba zvýši nervozitu, napätie a zníži vynaliezavosť, čiže dosiahne sa pravý opak požadovaného výsledku. Nesymetricky pozitívna

motivácia má kladný efekt [7], a úspešné smerovanie projektu treba dať pocítiť aj členom tímu, napríklad malými oslavami k ukončeniu určitého míľnika.

Samozrejme že problémy treba riešiť, ale konštruktívne. Je možné skúsiť napríklad brainstorming a odhaliť nejaké nové nápady. Alebo si na chvíľu dať s problémom pokoj, a neskôr sa riešenie objaví veľmi rýchlo, keď sa na problém pozrieme z inej strany. V prípade negatívnej motivácie nastáva aj útlm nových nápadov, ktoré by mohli priniesť riešenie, pretože v strese sa každý správa negativisticky a skôr hľadá prekážky a nedokáže vidieť pozitíva.

Nepochopenie požiadaviek

Celý projekt je v podstate implementácia požiadaviek. Ak tieto nie sú jasné hneď na začiatku, môže sa ľahko stať že sa projekt začne uberať zlým smerom, a tento smer už neskôr nebude možné jednoducho zmeniť. Preto treba dbať na to aby boli požiadavky poriadne vyjasnené so zadávateľom čo najskôr na začiatku projektu. Niektorí sa bránia zmenám požiadaviek v priebehu projektu pomocou kontraktov so zákazníkom, to ale niekedy môže mať za následok že klient súhlasí s niečím, čo sa mu na začiatku možno páči, ale neskôr všetci prídu na to že chcel niečo iné. Ak aj firma má kontrakt s funkčnými požiadavkami, klient nemôže namietat', ale nakoniec nebude spokojný.

Nikdy sa netreba spoliehať na to že klient vie čo od systému očakáva. Z doterajších procesov môže mať zafixované určité rutiny, ktoré sa mu zdajú byť jediným správnym spôsobom ako dosiahnuť cieľ. Takéto predsudky automaticky blokujú akékoľvek iné možné riešenia.

Preto sa treba na začiatku čo najviac zamerať na pochopenie čo budúci používateľ naozaj potrebuje, a nie na to čo si sám pýta [7]. Na to aby vznikla dohoda je ale nutné s ním všetko konzultovať až sa dospeje k obojstranne jasnému výsledku.

Niekedy je vhodné keď sa požiadavky menia aj počas realizácie projektu, v úzkej spolupráci s klientom pri každej fáze projektu. Samozrejme v obmedzenom rozsahu, inak každá zmena príliš spomaľuje vývoj. Na druhej strane sa ale viac približuje k požadovanému výsledku.

Nedostatočné sledovanie postupu prác

Niektorí ľudia sa v prípade problémov zvyknú tváriť že všetko je v poriadku, ale skutočnosti nevedia ako ďalej. Takáto situácia môže trvať dlhší čas, až nakoniec vyjde najavo, že v skutočnosti projekt stojí. Vtedy je to ako by ste šliapli na mínu. Z ničoho nič nastane zmätok, reorganizácia, zmena plánov, časový sklz.

Práve dostatočné reportovanie a kontrola reálneho stavu projektu môže zabrániť podobným situáciám. Niekedy nie je postačujúce uspokojiť sa s tým čo nám iní ľudia povedia, treba veci overiť dôkladne. To je efektívne možné iba keď sú procesy v rámci projektu dostatočne transparentné. Napríklad projekt je vhodnejšie rozčleniť na malé časti, ktoré je jednoduchšie kontrolovať. Takisto sa môže využiť sledovanie verzí, automatické testovanie a reportovanie.

Dôležité je, že to netreba preháňať. Každá práca vývojára pomimo jeho hlavnej úlohy v projekte ho môže značne spomaľiť. Aj pár minútové prerušenie počas produktívnej fázy ho dokáže vyvieť zo sústredenia a spomaľiť aj o niekoľko hodín.

Preto sa treba zamerať najmä na automatizované zaznamenávanie a sledovanie pokroku v projekte, aby sa tým čo najmenej obmedzovala kreatívna práca vývojárov.

Slabá komunikácia v tíme aj mimo neho

S rozsahom projektu sa zväčšuje okruh ľudí, ktorí musia navzájom komunikovať pri riešení úloh. Nie je dokonca zriedkavé, že na projekte pracuje viacero oddelených tímov z viacerých spoločností.

Ak je táto komunikácia nedostatočná, ľudia spolu nehovoria, niektoré problémy sa nedostanú do uší ľuďom, ktorí by ich mohli vyriešiť. Takto sa postupne nazbiera masa skrytých prekážok, valiacich sa spolu s projektom. A až nakoniec niektorá vypláva na povrch, jej odstránenie vyžaduje veľké úsilie. Tiež môže nastať situácia, keď viac ľudí pracuje na tom istom iba preto, lebo o tom nik iný nevie. Takto sa zbytočne plytvá ich časom.

Tímový manažér by mal priebežne sledovať čo sa v tíme deje, usporadúvať stretnutia, na ktorých by sa riešili všetky problémy. Raz za čas je dobré usporiadať stretnutie kde každý člen z tímu spomenie niečo na čo narazil ale nepočul o tom hovoriť nikoho iného. Niektoré veci sa totiž zdajú byť zrejme a jasné určitým ľuďom a vôbec ich nenapadne žeby o tom ostatní nevedeli. Takto je možné pomerne skoro odhaliť nové skryté problémy, alebo riešenia na tie už známe.

Použitie novej a neoverenej technológie

Vždy keď sa objaví nejaká nová aspoň trochu úspešná technológia, začne sa okolo nej tvoriť veľký záujem. To že o niečom všetci naokolo hovoria ešte neznamená že je to použiteľné a vhodné pre konkrétny projekt. Ak chcete takúto novinku použiť mali by ste si dobre rozmyslieť jej pozitíva, negatíva a aj prípadné riziká, ktoré môže priniesť ak sa neosvedčí.

Záujem o nové technológie sa väčšinou riadi podľa známej „hype“ krivky. Po období veľkého záujmu nastáva útlm, a až neskôr sa situácia stabilizuje. Bezpečný spôsob je najskôr nechať iných overiť jej zrelosť. Určite sa nájdú takí čo naskočia na kolotoč a nie všetkým sa to vyplatí. Treba si brať skúsenosti z iných projektov, ktoré sa ju snažili implementovať, či už sa im to podarilo alebo nie. Takto môžete získať prehľad o prípadných problémoch, na ktoré môžete s touto technológiou naraziť.

Podľahnutie komerčným tlakom

Pri každom komerčnom projekte je prirodzený tlak na minimalizáciu nákladov a času potrebného na ukončenie projektu. Netreba sa nechať dotlačiť k stene a prijať podhodnotené odhady a rozpočet. Ak si necháte vnútiť niečo čo nie je možné splniť, raz musíte naraziť na hranicu možností, a potom sa situácia skomplikuje, čo nakoniec povedie k celkovému predraženiu a zníženiu kvalitu softvéru. Čo je samozrejme horšie ako hneď na začiatku povedať *nie*, aj keď sa to viacerým ľuďom nemusí páčiť.

Najlepšie je úplne sa odpútať od finančných implikácií projektu, jedine tak môžete mať nenarušený pohľad na skutočné nároky. Potom je možné znížiť požiadavky v prípade že klient si nemôže dovoliť kompletný systém, treba pritom ale

pamätať na to, že je veľká pravdepodobnosť dopracovania odstránených častí v budúcnosti.

Nedostatočné testovanie

Toto úzko súvisí so sledovaním postupu projektu. Zo zväčšovaním implementovaných častí projektu sa zväčšuje aj pravdepodobnosť zanesenia rôznych chýb. Čím neskôr sa na chybu príde tým viac úsilia si vyžaduje jej odstránenie. Je to tým že čím ďalej tým viac častí projektu je touto chybou ovplyvnených. Neskôr je pri odstraňovaní chyby nutné upraviť všetky takéto časti.

Rovnako ako včasné reportovanie zaostávajúcich činností je dobré priebežne sledovať a testovať stav plnenia funkčných požiadaviek na výsledok projektu. Čím skôr sa odhalia problémy a budú sa riešiť, tým lepšie. Dobrá protipožiarna ochrana je vždy lepšia ako schopní požiarnici.

Netreba sa báť použitia automatických systémov testovanie, ktoré už momentálne sú na dostatočne vysokej úrovni a ponúkajú veľmi dobrú podporu projektu [4]. Hlavne s využitím nástrojov na sledovanie verzií sú nevyhnutnou súčasťou každého väčšieho projektu, pretože zvyšujú transparentnosť a pomáhajú rýchlo odhaliť chyby a presnú zmenu, ktorá k nim viedla. Manuálne testovanie má stále svoju úlohu, ale je vhodné skôr na testovanie užívateľského rozhrania. Čo sa týka manuálnych testov tak by ich určite nemali robiť tímoví vývojári, ktorí sú ovplyvnení svojím pohľadom, ale špeciálna skupina, ktorá je oboznámená s požiadavkami používateľa.

Záver

V tomto článku som zhrnul a popísal niekoľko chýb, ktorých sa môže dopustiť projektový manažér, a ktoré môžu skomplikovať, prípadne úplne zastaviť softvérový projekt. Dúfam že tento krátky prehľad niekomu pomôže sa zamyslieť, možno aj predísť chybám a úspešne zvládnuť projekt.

Použitá literatúra

1. Brooks, F.: The mythical man-month: Essays on Software Engineering, 20th Anniversary Edition. Addison Wesley, 1995. (ISBN: 0-20-183595-9)
2. Charette, R.N: Why software fails, In: IEEE Spectrum, Sep, 2005
3. Gaitros, D.A.: Common Errors in Large Software Development Projects, In: *Crosstalk, The journal of defense software engineering*, Mar, 2004.
4. Hurley, F.: Seven Tips for Keeping Software Development Projects Healthy, In: *IT Professional*, Jul-Aug, 2002, 60-63.
5. Marasco, J.: Software Development Edge, The: Essays on Managing Successful Projects, Addison Wesley, April, 2005, (ISBN: 0-32-132131-6)
6. May, L.J.: Major Causes of Software Project Failures, In: *Crosstalk, The journal of defense software engineering*, Jul, 1997.