

TRADIČNE ALEBO AGILNE? PREČO SA MA ZAUJÍMAŤ AJ ZÁKAZNÍK?

*Tak ako vývoj dieťaťa sleduje mama aj otec, tak musí
vývoj softvéru sledovať vývojár aj zákazník.*

Dávid Bado

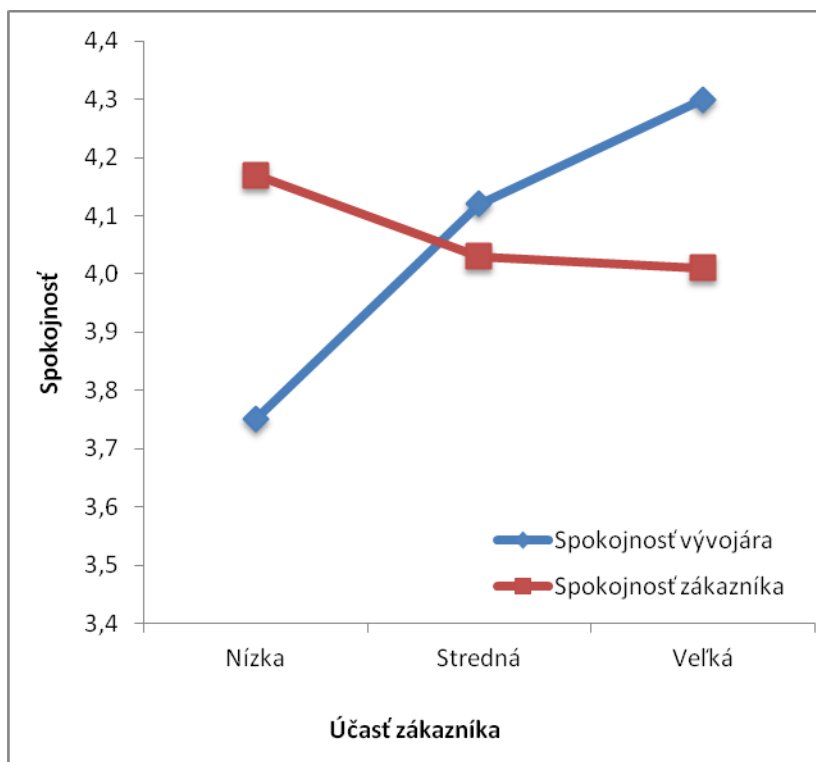
Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
bado.david@gmail.com

Abstrakt. *V súčasnej dobe veľké percento IT projektov zlyháva. Preto sa veľa ľudí snaží skúmať a odhaľovať príčiny týchto zlyhaní. Výskum ukazuje, že jeden z hlavných dôvodov zlyhania softvérových projektov je nedostatočná účasť zákazníka na projekte. Esej hovorí o tom, prečo je dôležité, aby zákazník softvérový projekt monitoroval. Poukazuje na to, že tradičný vývoj softvéru tento trend nepodporuje a dokonca kladie zákazníkovi pri monitorovaní prekážky. Naopak agilné metódy zákazníka podporujú v tom, aby projekt monitoroval a aby sa na ňom podieľal. Esej apeluje na zákazníka, aby si uvedomil, že výberom metódy vývoja a svojou účasťou môže zmenšiť riziko zlyhania projektu.*

Kľúčové slová: *monitorovanie, agilný vývoj, zákazník, riziká, úspech projektu*

Úvod

V roku 2010 bolo podľa prieskumu úspešných iba 34 % IT projektov. Ako jeden z najdôležitejších faktorov, ktoré úspech IT projektu ovplyvňujú, bola označená účasť zákazníka [6]. Tieto fakty jasne naznačujú, že účasť zákazníka priamo súvisí so spokojnosťou vývojárov aj zákazníka a v konečnom dôsledku teda aj s úspechom softvérového projektu (viď Obr.1). No vyplýva z toho aj to, že zákazníci sa podieľajú na vývoji softvéru nedostatočne, čo spôsobuje tak nízku úspešnosť IT projektov.



Obr. 1 Závislosť spokojnosti od účasti zákazníka [6].

Cieľom každého softvérového projektu je vytvoriť čo najkvalitnejší produkt. Produkt, ktorý bude spĺňať požiadavky definované zákazníkom, bude dokončený v stanovenom termíne a za vopred dohodnuté náklady. To samozrejme nie je vždy možné. Vždy sa vyskytnú prekážky a problémy. Preto by mala byť snaha takýmto rizikám predchádzať monitorovaním projektu.

O zabránenie zlyhaniu projektu by sa mali rovnako snažiť zákazník aj vývojár a podľa toho by sa mala odvíjať aj ich spolupráca na projekte [8]. Zákazník by sa mal z vlastnej vôle a vo vlastnom záujme snažiť priebežne monitorovať projekt a vývojári by mu to mali umožniť a podporovať ho v tom. Predsa zlyhanie projektu, najmä veľkých rozmerov, spôsobí značné straty nie len vývojárom, ale hlavne samotnému zákazníkovi.

Riziká nezájmu zákazníka

Vyššie bolo spomenuté, že pre úspech projektu je vhodné, aby sa zákazník podieľal na projekte. Neskôr budem hovoriť, prečo je to dôležité. Čo sa však stane, keď zákazník len predostrie svoje požiadavky a nijak nespupracuje s vývojármi. Na začiatku projektu vysvetlí, čo od softvéru požaduje. Dohodne náklady a termín dodania projektu a ďalej sa o projekt nezaujíma. V takomto prípade sa výrazne zvyšujú riziká.

Základným rizikom je riziko nesprávneho pochopenia požiadaviek zákazníka zo strany vývojárov, čo môže mať za následok zväčšovanie ďalších súvisiacich rizík. Napríklad riziko vytvárania nesprávneho produktu, riziko nesprávneho rozplánovania

projektu, riziko nezabezpečenia dostatočných zdrojov a s tým súvisiace riziko nesprávneho odhadu rozpočtu. Vymenované riziká smerujú k situácii, ktorej sa ľudia snažia vyhnúť vo všetkých projektoch. Je to situácia, keď dôjde k nedodržaniu termínu alebo prekročeniu ceny, alebo vytvoreniu nesprávneho produktu alebo ku kombinácii týchto zlyhaní.

A práve monitorovanie projektu zákazníkom a vývojárom pomáha vyššie spomenuté riziká výrazne znižovať [8].

Čo monitorovať a prečo?

Pri vývoji softvéru je dôležité vždy na začiatku zvoliť prístup, ktorým sa bude vývoj riadiť. Od tejto voľby sa bude samozrejme odvíjať aj monitorovanie projektu. Budú od nej záležať metriky, ktoré sa pri monitorovaní budú sledovať, intervaly, v akých sa metriky budú vyhodnocovať a samozrejme aj miera zainteresovania zákazníka do procesu monitorovania.

Vhodne zvolené metriky významne ovplyvňujú schopnosť vývojárov, ale aj zákazníkov, dostatočne rýchlo a vhodne reagovať na vzniknuté prekážky v procese vývoja. Umožňujú včasné odhalenie rizík a predvídanie možných zlyhaní pri práci na projekte.

Preto je dôležité, aby tieto metriky boli zvolené s vedomím, že ich budú sledovať vývojári aj zákazník. To znamená, že budú zrozumiteľné pre obe skupiny [8]. Zákazník musí byť schopný výsledky monitorovania vyhodnotiť, aby v prípade potreby mohol podľa nich rozhodovať a konať [8]. Rovnako je dôležité, aby metriky odrážali skutočnosť a nevytvárali falošný obraz o projekte. Aby nenastala situácia, keď podľa oficiálneho monitorovania pôjde všetko podľa plánu a v skutočnosti budú práce na projekte pozadu, čo sa môže odhaliť až pri hraničnom termíne. Takéto prekvapenie v lepšom prípade projekt predĺži a predraží, v horšom prípade môže spôsobiť jeho zlyhanie.

Intervaly, v akých sa budú metriky vyhodnocovať, sú ďalším dôležitým aspektom monitorovania. Ovplyvňujú, či sa problémy a prekážky odhalia včas, aby bolo na ne možné reagovať rýchlo a efektívne. Veľmi krátke intervaly a prílišná participácia zákazníka môžu spôsobiť zbytočné komplikácie a spomalenie prác na projekte [6]. Príliš dlhé intervaly môžu spôsobiť, že zákazník stratí prehľad o projekte. To môže mať pre projekt fatálne následky. Intervaly je potrebné nastaviť dohodou zákazníka a vývojára tak, aby monitorovanie pomáhalo projektu a nie mu vytváralo prekážky.

Vývojár by mal svoje postupy prác na projekte komunikovať so zákazníkom relatívne často. Keďže zákazník nie je technicky znalý a vývojár nevidí do domény, v ktorej softvér vyvíja, môžu vzniknúť nedorozumenia [6]. A čím skôr zákazník bude vidieť, ako sa jeho požiadavky plnia, tým skôr si môže vytvoriť lepšiu predstavu. To mu pomôže posúdiť, či to, čo požadoval, aj skutočne od softvéru dostane alebo je potrebné požiadavky korigovať. Rovnako, čím skôr sa vývojár dozvie o potrebných zmenách, tým menej bude musieť zasahovať do už vytvorenej práce. Zmena bude jednoduchšia, rýchlejšia a lacnejšia.

Toto všetko v konečnom dôsledku ovplyvní, či bude projekt úspešný alebo či zlyhá.

Kde tradičné metódy zlyhávajú

Jednou z hlavných hrozieb pri tradičnom vývoji softvéru je spomínaný nezáujem zákazníka o monitorovanie projektu, čo môže viesť k dodaniu softvéru, ktorý nebude spĺňať požiadavky zákazníka alebo dokonca zlyhaniu projektu. No to nie je jediný aspekt, ktorý môže spôsobiť problémy. To, že zákazník projekt monitoruje ešte neznamená, že sú riziká minimalizované. Nevhodne zvolené metriky monitorovania projektu sú rovnako nebezpečné ako nezáujem zákazníka [5].

Tradičné metódy monitorovania projektu sa sústreďia na sledovanie a meranie aktuálneho progresu na projekte a jeho porovnávanie s plánom [7]. Sledujú napr. koľko úsilia bolo vynaloženého na danú úlohu. Takáto metrika sa však ukazuje ako nevhodná. Hlavne pre zákazníka, ktorý chce na základe tejto metriky zistiť, ako pokročili práce na projekte. Údaj o tom, že koľko času bolo stráveného na plnení danej úlohy mu vôbec neposkytne predstavu o tom, do akej miery bola úloha splnená alebo koľko času je potrebného na jej dokončenie. Aj keď z plánu vie aký je optimálny čas dokončenia úlohy, stále mu tento údaj nedá skutočný pohľad na progres prác. To môže spôsobiť podcenenie rizík alebo si ich neuvedomenie.

Ďalším aspektom, ktorý sa pri tradičných prístupoch sleduje, je miera otestovania kódu. Údaj, ktorý sa udáva v percentách, by mal predstavovať určitú hodnotu kvality. Martin Fowler o miere otestovania povedal: „Z času na čas počujem pýtať sa ľudí aká miera otestovania by mala byť ich cieľom, alebo ich počujem chváliť sa svojou mierou otestovania.“[4] Touto poznámkou sa podľa mňa snažil naznačiť, že táto metrika stále predstavuje len koľko, kódu bolo otestovaného. Nič viac a nič menej. Veľmi presne nám vie povedať koľko kódu je ešte potrebné otestovať, ale nič nehovorí o kvalite. Softvér môže byť pokojne otestovaný na 90 %, no stále to neznamená, že spĺňa podmienky na 90 %. Prečo? Lebo každý robí testy inak. Je jednoduché napísať testy na 100 % kódu. Náročnejšie je napísať testy tak, aby nám odhalili, či naša funkcionálna skutočne spĺňa to, čo má. A aj sebe lepšie testy nám nikdy neodhalia všetky možné scenáre a riziká. Preto sa na mieru otestovania nemôže spoľahnúť ani zákazník. Ten síce dostane číselný údaj, ktorému rozumie, ale je preňho zavádzajúci a mätúci.

Následne môže vzniknúť situácia, že zákazník dostane údaj, že softvér je na 95 % otestovaný. To v ňom vzbudí dojem, že projekt je už takmer hotový, a nevyvíja žiadnu snahu, aby zrýchlil vývoj. Je spokojný s prácou vývojárov. Keď sa však v hraničnom termíne zistí, že v skutočnosti je dokončený projekt len na 70 %, nastanú problémy pre zákazníka aj pre vývojára. A to v dôsledku nesprávne nastavenej metriky.

Spomenuté metriky ukazujú, že navrhnuť efektívne metriky na monitorovanie projektu nie je jednoduché. A tak ako sa vyvíjajú metódy vývoja softvéru, tak sa vyvíjajú aj metriky monitorovania. Pôvodné metódy vývoja, tzv. tradičné sa inšpirovali pri určovaní metrik inými oblasťami, ktoré vytvárali fyzické produkty. Neskôr sa ukázalo, že softvér má niektoré vlastnosti, ktoré fyzické produkty nemajú a ktoré je potrebné pri vývoji zohľadniť. Jednou z problémových vlastností je neviditeľnosť softvéru. Ukázalo sa, že spôsobuje dovedy neexistujúce problémy pri monitorovaní. Preto niektorí ľudia začali nad vývojom a monitorovaním softvéru uvažovať inak. Pochopili, že keď chcú projekt úspešne monitorovať musia určiť metriky, ktoré budú s neviditeľnosťou softvéru počítať. Nasledujúca kapitola ukazuje, že sa takéto metriky aspoň do istej miery podarilo vytvoriť.

Prečo radšej agilne?

Pri vývoji softvéru použitím agilného vývoja vznikajú dve výhody oproti tradičnému vývoju. Lepšia viditeľnosť softvéru a väčšia kontrola nad vývojom. Pri agilnom vývoji sa pravidelne vytvára fungujúci softvér (prototyp) a prezentuje sa zákazníkovi. To umožňuje zákazníkovi monitorovať, čo sa v skutočnosti na projekte robí a takmer okamžite vidieť, či vývojári pochopili jeho požiadavky a či dostáva to čo chce. Takáto informácia má preňho oveľa väčšiu hodnotu, ako akákoľvek dokumentácia. Navyše mu poskytuje väčšiu kontrolu nad projektom, pretože okamžite vidí, či sa plní časový plán, či je rozpočet dobre nastavený, atď. Na základe tohto pravidelného monitorovania môže rýchlo zareagovať na problémy, zdržania a nasmerovať projekt správnym smerom a predísť tak stratám. Či už finančným, časovým alebo aj kvalitatívnym [7].

Netradičné agilne metódy majú samozrejme upravené metriky, ktoré sa pri vývoji softvéru sledujú. Vzhľadom na to, že jedným z princípov agilného vývoja je „Ludia z biznisu a vývoja musia spolupracovať denne po celú dobu projektu.“ [2], sú metriky vytárané tak, aby boli univerzálne. Musia byť použiteľné rovnako pre zákazníka aj vývojára. Obom musia poskytnúť reálny pohľad na prácu na projekte. Hlavne je potrebné, aby ich chápal aj zákazník a aby ich vedel sám vyhodnotiť.

Pri agilnom vývoji sa používa na ukážku pokroku prác na projekte metrika rýchlosť (angl. velocity) [5]. Metrika hovorí o tom, koľko používateľských scenárov sa urobilo na projekte za určitý čas (napr. mesiac). Zákazník podľa nej získa konkrétnu predstavu koľko práce sa urobilo, resp. aká časť softvéru bola dokončená. Presne vidí, ktoré funkcionality boli vyvinuté. To mu pomôže spraviť si oveľa lepšiu predstavu o plnení plánu, ako pri tradičných metrikách. Keď nastane situácia, že rýchlosť je oveľa nižšia, ako sa predpokladalo, zákazník aj vývojári relatívne rýchlo zistia, že nastal problém, ktorý treba riešiť. Rýchlosť je metrika jednoduchá, no napriek tomu zrozumiteľná pre vývojára aj zákazníka. Preto si myslím, že je pre monitorovanie veľmi vhodná.

Príkladom metriky agilného vývoja, ktorý odzrkadľuje kvalitu softvéru je metrika RTF (angl. running tested features). Metrika, ktorá udáva počet fungujúcich otestovaných funkcionality. RTF je podobná tradičnej miere otestovania softvéru, ale má jednu výhodu. Hovorí o funkcionality, ktoré sú plne funkčné, otestované a pripravené na nasadenie. Opäť metrika, ktorá zákazníkovi zrozumiteľne ukazuje kvalitu softvéru. Nie je zavádzajúca, práve naopak. Keď je nejaká funkcionality označená za fungujúcu a otestovanú, zákazník môže považovať prácu na nej za ukončenú. Metrika RTF je ďalším príkladom metriky vhodnej na monitorovanie projektu, ktorá zákazníkovi včas ukáže, či práce na projekte prebiehajú tak ako majú. Ukazuje dôležitosť monitorovania vývoja softvéru zo strany zákazníka, pre ktorého by malo byť sledovanie množstva dokončených funkcionality kľúčové a rovnako dôležité, ako pre vývojára. Od neho záležia všetky aspekty, ktoré by mali byť pri vývoji softvéru sledované. Termín dokončenia, cena a kvalita softvéru.

Metriky monitorovania agilného vývoja spomenuté vyššie sú dôkazom toho, že existujú metriky, ktoré počítajú pri monitorovaní s neviditeľnosťou softvéru a zároveň sú natoľko jednoduché a zrozumiteľné, že ich dokážu používať okrem vývojárov aj zákazníci. Preto by bolo zo strany zákazníkov nerozumné nevyužiť tieto metriky a monitorovanie

ako také, aby znížili riziko zlyhania projektu. Zákazníci by sami seba dostali do konkurenčnej nevýhody.

Záver

Vývoj softvéru je jeden veľký a zložitý proces skladajúci sa z veľa malých podprocesov, ktoré je potrebné koordinovať. Preto nie je jednoduché dotiahnuť softvérový projekt do úspešného konca. Štatistiky ukazujú, že úspešnosť softvérových projektov nie je na takej úrovni, ako by si mnohí želali. Prečo? Výskum odhalil, že jednou z hlavných príčin zlyhania softvérových projektov je nedostatočná účasť zákazníka na projekte.

Zákazník by mal vo vlastnom záujme brať do úvahy všetky možné príčiny zlyhania projektu. Preto by sa mal vyvarovať svojej nedostatočnej účasti na projekte, keďže je to jedna z hlavných príčin zlyhania projektov. Touto príčinou by sa mal zaoberať hlavne preto, lebo sa dá úplne odstrániť, čo sa o všetkých potenciálnych príčinách zlyhania projektov povedať nedá. Ale aj preto, lebo sú k dispozícii prostriedky, ktoré na tento účel slúžia. Takéto prostriedky poskytujú agilné metódy vývoja, ktoré zákazníka priamo začleňujú do vývoja softvéru, podporujú ho v monitorovaní projektu a uľahčujú mu ho používaním vhodných metrík.

Samozrejme v prvom rade musí mať zákazník vôľu sa na projekte podieľať. Musí si uvedomiť, že zvýšenie pravdepodobnosti úspešnosti projektu môže do veľkej miery ovplyvniť podieľaním sa na vývoji tak, že bude projekt monitorovať. Monitorovanie projektu síce stojí zákazníka nejaké prostriedky, no v značnej miere pomáha minimalizovať riziko zlyhania projektu. Preto by si zákazník mal vyberať takých vývojárov, ktorí vyvíjajú agilne a jeho ochotu monitorovať podporia. Tak má vyvíjaný softvérový produkt najväčšiu šancu na úspech.

Použitá literatúra

1. Ambler, S.: *Portfolio Management and Agile Software Development: Monitoring and Governance*. 2009.
Online: <http://www.cutter.com/research/2009/edge090310.html>
2. Beck, K.; et al.: *Manifesto for Agile Software Development*, 2001.
3. Cheng, T.-H., Jansen, S., & Remmers, M.: *Controlling and monitoring agile software development in three dutch product software companies*. *2009 ICSE Workshop on Software Development Governance*, IEEE Computer Society, Washington (2009), 29-35.
4. Fowler, M.: *Test Coverage*. 2012.
Online: <http://martinfowler.com/bliki/TestCoverage.html>
5. Ktata, O.: *Designing and Implementing a Measurement Program for Scrum Teams : What do agile developers really need and want ?*, ACM, New York (2010), 101-107.
6. Subramanyam, R., Weisstein, F. L., & Krishnan, M. S.: *User participation in software development projects*. *Communications of the ACM*, vol. 53, no. 3(2010), 137-141.
7. Thomas, S.: *Agile Project Monitoring and Control*. 2008.
Online: <http://itsadeliverything.com/agile-project-monitoring-and-control>

8. Tsunoda, M., Matsumura, T., & Matsumoto, K.-ichi.: Modeling Software Project Monitoring with Stakeholders. *2010 IEEE/ACIS 9th International Conference on Computer and Information Science*, 2010, 723-728.

Annotation

Traditional or agile? Why should customer care?

Nowadays a large percentage of IT projects fail. Because of that, many people are trying to find and examine causes of these fails. Research shows that one of the main causes of fail of software project is insufficient participation of customer on project. Essay talks about why it is important to monitor software project by customer. It points out that traditional software development does not support this trend and even creates obstacles for customer during monitoring. On the other side agile methods encourage customer monitor project and participate on it. Essay appeals to customer to realize, that he can minimize risk of project failing by choosing development method and by his participation in project.