

Čo zaručuje úspech softvérového projektu

RASTISLAV BERTUŠEK

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava*

Abstrakt. Skoro každý softvérový projekt je niečím jedinečný. Poznatky získané z jeho vývoja sa ťažko aplikujú na iné softvérové projekty. Napriek tomu sú určité faktory, ktoré pri tvorbe softvérového projektu zohrávajú kladnú alebo naopak zápornú úlohu. Dnešnú dobu charakterizuje stále sa zrýchľujúci proces vývoja softvérových projektov. Preto je priam nevyhnutné, aby sme v tomto procese využívali poznatky z analýzy príčin úspechu a neúspechu softvérových projektov vytvorených v minulosti. Manažment projektu by mal ovládať tieto poznatky a intuitívne sa podľa nich riadiť. Aké sú príčiny, ktoré vedú k úspechu alebo neúspechu softvérového projektu? Následkom akých rozhodnutí manažmentu projektu sú tieto príčiny? Čím lepšie budeme schopní odpovedať na tieto a podobné otázky, tým úspešnejšie budeme vedieť vytvárať softvérové projekty. V eseji sa snažím zachytiť, ale najmä analyzovať tieto príčiny. Vychádzam pritom z prieskumu, ktorý sa zaoberal úspešnosťou softvérových projektov.

Úvod

Úspech softvérového projektu (ďalej iba projektu) je subjektívny pojem. Napríklad pre niekoho je neúspešný projekt taký, ktorý nebol dodaný vo vopred stanovenom čase alebo za daných podmienok, v stanovenej kvalite, v rámci stanoveného rozpočtu, zadaných požiadaviek. Čo pre niekoho znamená úspech alebo aspoň čiastočné naplnenie cieľov projektu, pre iného môže znamenať neúspech. Preto je na každom z nás, kde si určíme pomyselnú hranicu úspechu a neúspechu.

Takisto softvérový projekt je veľmi široký pojem. Za softvérový projekt je možné považovať skript vytvorený študentom, ktorý sa jednorazovo použije, a na druhú stranu projekt s celosvetovým dosahom (napr. operačné systémy, databázové systémy a iné generické projekty) alebo s niekoľko miliardovým rozpočtom. Preto sa články, zaoberajúce sa manažmentom softvérových projektov, zameriavajú najmä na komerčné projekty s vopred stanovenými a merateľnými podmienkami. Prieskum, ktorý sa zaoberá príčinami úspechu resp. neúspechu projektu, bol publikovaný v článku [5] (ďalej uvádzam iba ako prieskum). Zaoberá sa 122

komerčnými projektmi vytváranými v rámci finančných, bankových a úverových inštitúcií. Tiež nechali ohodnotenie úspešnosti projektu na jednotlivých respondentov prieskumu. V tomto článku sa snažím rozobrať niektoré zaujímavé výsledky prieskumu. V eseji popisujem úspešnosť softvérových projektov vo všeobecnosti. Následne rozoberám niektoré problémové oblasti:

- model vývoja,
- projektový manažment,
- plánovanie,
- požiadavky na projekt,
- centrálna správa,
- manažment rizík a analýza po ukončení projektu.

Úspešnosť softvérových projektov

Pri nekomerčných projektoch je miera úspechu prakticky nestanoviteľná, vzhľadom na osobitné prostredie pri ich tvorbe. Sú to najmä nešpecifikované časové termíny, ciele, požiadavky, testovanie a pod. Preto je vo veľkom množstve takýchto projektov prinajmenšom ťažké určiť mieru ich úspechu/neúspechu a nemá zmysel hľadať jeho príčinu.

Podľa prieskumu iba 55 % vývojových projektov bolo úspešných. Po započítaní projektov určených na údržbu a zlepšovanie už vyvinutých projektov percentuálne číslo stúplo iba na hodnotu 62 %. Z toho vyplýva alarmujúca informácia, že ani silné finančné zázemie a potenciál, ktorý majú inštitúcie zúčastnené v prieskume, nezaručuje úspešné ukončenie projektu. Teda kľúčom k úspechu alebo naopak príčinou neúspechu nie je v prvom rade finančné zabezpečenie projektu.

Model vývoja

Jednou z hlavných príčin úspechu softvérového projektu je jeho model vývoja. Viac ako polovica projektov bola vytváraná vodopádovým modelom (57 %), čo svedčí o konzervatívnosti politiky manažérov a manažmentu inštitúcie. Naopak, modely prototypovania (5 %), inkrementálny (2 %) a špirálový (2 %) mali spolu iba 9 %. Je zjavné, že novšie vývojové modely ako prototypovanie a inkrementálny, ktoré ešte nie sú dostatočne podrobne zadefinované a podporované vývojovými nástrojmi, nemajú u manažérov zelenú. Ich používanie sa presadzuje napríklad v agilnom programovaní alebo v extrémnom programovaní (ang. extreme programming). Pravdou však je, že vodopádový model má veľmi dobrú čitateľnosť, kontrolu kvality, časových ohraničení. Tieto požiadavky na projekt sú pre finančné inštitúcie veľmi dôležité. Myslím si, že ak by sa prieskum konal v iných organizáciách, výsledok percentuálneho rozloženia

modelov by mohol byť odlišný. Napriek tomu, že modelom prototypovania a JAD modelom (joint application design) sa riadilo iba 13 projektov, 9 projektov bolo úspešných, čo je výrazne viac ako priemerný počet úspešných projektov. Zaujímavosťou bolo, že iba 3 projekty z 8, ktoré používali UML pri dokumentovaní požiadaviek, boli úspešné. Príčinou toho podľa môjho názoru bolo nedostatočné zvládnutie novej terminológie. Pri využívaní UML a obzvlášť pri využívaní RUP procesu vývoja, je nevyhnutné si jasne stanoviť jednotlivé kroky vývoja projektu. Jeden z mnohých príkladov podrobného rozpracovania procesu vývoja je v dokumente [4].

Projektový manažment

Model vývoja v organizácii sa stanovuje najmä projektovým manažmentom. Ďalším výsledkom prieskumu bolo zistenie, že zmena projektového manažmentu výrazne negatívne ovplyvnila úspech projektu. Preto je veľmi podstatné, aby sa kládol veľký dôraz na výber správneho projektového manažmentu ešte pri jeho formovaní. Pritom by sa mal projektový manažér vyberať najmä vzhľadom na jeho schopnosti riadenia. Jeho znalosti v odbore sú v tomto prípade minoritnou záležitosťou. Je dôležité mať skôr všeobecný náhľad než vynikajúce znalosti v jednej špecifickej oblasti. Manažér musí mať schopnosť komunikácie a musí vedieť vytvoriť priateľskú atmosféru v tíme. Tieto vlastnosti manažéra posúvajú projekt bližšie k úspechu.

Keď sa projekt vytvára podľa dobre stanoveného plánu a jasnou víziou cieľov, každý člen tímu si ľahšie a presnejšie naplánuje svoj individuálny plán splnenia požiadaviek na neho kladených. Projekt musí mať preto čo najpresnejšie a najpodrobnejšie stanovený harmonogram jednotlivých krokov. Z tohto pohľadu je tragické, že väčšina prvotných časových a rozpočtových špecifikácií projektu vyjednáva a schvaľuje vrcholový manažment. Nielenže projektovým manažérom nie je na začiatku projektu umožnené dohodnúť sa o časových rámcoch, obsahu, cieľoch a finančných otázkach projektu, ale čo je horšie, je im aj v priebehu riešenia softvérového projektu bránené tieto otázky otvárať a modifikovať. Napokon sa projektu prideli projektový manažment, ale ten už má nadiktované rámcové mantinely. Nanešťastie je zrejmé, že sa časové a rozpočtové mantinely musia až príliš často meniť v priebehu projektu. Vrcholový manažment by mal čo najskôr ustanoviť manažment projektu a následne mu prenechať hlavnú úlohu pri prvotnej komunikácii so zákazníkom a pri stanovovaní prvotnej špecifikácie projektu.

Samozrejme aj priebežná komunikácia o zásadných otázkach, ako je dátum odovzdania projektu do testovania, do prevádzky alebo rozpočet projektu, by sa mala vykonávať čo najpriamejšie medzi manažmentom projektu a zákazníkom bez sprostredkovateľa ako napr. vrcholový manažment. Vrcholový manažment by do tohto procesu mal zasahovať v čo najmenej nevyhnutnej miere. S tým úzko a neoddeliteľné súvisí zhromažďovanie a spracovanie požiadaviek na projekt. Ak manažment projektu nemá už od začiatku projektu priamy kontakt so zákazníkom, nie je schopný vytvoriť prvotný katalóg požiadaviek a potom nemá dostatočné podklady pre stanovenie

rozpočtu a časového harmonogramu. Z toho logicky vyplýva, že je nevyhnutné mať pri plánovacích rozhodnutiach relevantné informácie a požiadavky zákazníka.

Pracovníci v tíme nebudú môcť plniť dobre stanovený časový harmonogram, ak pracujú v stresovom prostredí. Nevhodné vedenie a nedostatočná komunikácia zo strany manažmentu projektu má za následok postupný nezáujem o projekt alebo naopak nadmernú psychickú záťaž pracovníka. Pre dosiahnutie dobrých výkonov by mali pracovníci pociťovať príjemný prístup zo strany manažmentu. Musia byť priebežne povzbudzovaní či už jednoduchou pochvalou alebo finančnou odmenou. Ak pracovníci pociťujú priebežnú oporu, sú pozitívne naladení, dokážu lepšie komunikovať s okolím, pracovať tvorivejšie a tým pádom aj efektívnejšie a rýchlejšie.

Plánovanie

Projektový manažment by tiež mal byť jednou zo základných zložiek projektového plánovania. Jednou z hlavných príčin zlého plánovania je optimistické stanovenie nákladov a časového harmonogramu. Prieskum [5] potvrdil nutnosť mať jasnú víziu o budúcom smerovaní projektu ako najlepší predpoklad úspechu zo strany finančného manažmentu. Tento problém je už dlhšiu dobu známy, ale ťažko riešiteľný. Nie je totiž jednoduché vyjednávať, ak nemáme dostatočnú špecifikáciu projektu, a preto sa snažíme započítať riziká a stanoviť náklady a časový plán s rezervou. Zákazník sa zasa naopak snaží minimalizovať časový harmonogram a cenu, za ktorú sa daný projekt má realizovať.

Vo viac ako dvoch tretinách projektov vrcholový manažment, zákazník alebo používateľ stanovovali náklady a časový harmonogram projektu, čo samozrejme spôsobilo, že z toho viac ako dve tretiny boli zle stanovené náklady a harmonogram. Dalším prekvapujúcim zistením prieskumu bolo tvrdenie, že nedostatočne stanovené požiadavky na začiatku projektu nespôsobili vo výraznej miere neúspech projektu. Podľa môjho názoru, ale táto zdanlivá nezávislosť bola spôsobená iba dodatočnou komunikáciou a došpecifikovaním, resp. korigovaním prvotnej špecifikácie projektu manažmentom projektu. Teda v konečnom dôsledku sa potvrdzuje tvrdenie, že manažment projektu musí byť schopný komunikácie na vysokej odbornej, rečnickej a psychologickej úrovni.

Požiadavky na projekt

Jedným zo základných podkladov pre projektové plánovanie sú požiadavky na projekt. Pri veľkých projektoch sa prejavuje efekt nedocenenia prvotnej špecifikácie požiadaviek. Väčšina z viac ako polovice projektov, ktorým sa výrazne menili požiadavky v priebehu projektu boli veľké (v kontexte prieskumu) projekty. Spôsobil to fakt, že veľké projekty sú omnoho viac náchylné na nedostatočnú špecifikáciu požiadaviek. Zmena čo len jednej požiadavky môže vyvolať zmenu obrovského množstva súvisiacich požiadaviek a znehodnotiť množstvo už vykonanej práce. Pri veľkých projektoch je taktiež nutné rátať s rôznymi aspektmi, ktorých dôležitosť sa

bohužiaľ zanedbáva. Ich dôležitosť ale narastá s veľkosťou projektu. Zjednodušene povedané, požiadavky a manažment rizík, ktorý sa vykonáva pri malých projektoch, nie je zďaleka postačujúci a priamo aplikovateľný na veľké projekty. Tento prístup je obdobný aj v iných oblastiach manažmentu ako napr. manažment bezpečnosti, manažment riadenia (vid'. [1]). Zároveň ale treba dbať pred stanovovaním požiadaviek, aby sa pochopil problém do čo najväčšej hĺbky.

Tab. 1 zobrazuje niektoré role v softvérovom projekte podľa článku [2].

Rola	Motivácia	Expertízna oblasť
Zákazník	Zmena s maximálnym úžitkom	Firemné a informačné systémové stratégie, trendy v priemysle
Používateľ	Zmena s minimálnym narušením	Firemný proces, operačné procedúry
Projektový manažér	Úspešné ukončenie projektu s danými zdrojmi	Projektový manažment, softvérový vývoj a proces dodania
Analytik	Špecifikácia požiadaviek načas a s daným rozpočtom	Metódy a prostriedky inžinierstva požiadaviek
Vývojár	Vytvorenie technicky vynikajúceho systému, použitie najnovších technológií	Najnovšie technológie, návrhové metódy, programovacie prostredia a jazyky
Sledovanie kvality	Vyhovenie procesovým a produktovým štandardom	Softvérový proces, metódy, štandardy

Tab. 1. Motivácia a expertízna oblasť niektorých rolí v projekte.

Z tabuľky vyplýva, že práve analytik má na starosti čo najlepšiu identifikáciu požiadaviek. Analytik používa metódy a nástroje inžinierstva požiadaviek. Na jeho pleciach preto stojí jedna z najdôležitejších a najťažších úloh v softvérovom projekte. Od jeho úsudku identifikovania správnych a vhodných požiadaviek vzhľadom na poskytnuté financie závisí osud celého projektu. Úspešní analytici poznajú do hĺbky sledovanú oblasť. Tento cieľ dosahujú dvoma cestami. Prvá cesta je vlastná skúsenosť v danej problémovej oblasti. Druhá v praxi omnoho viac používaná metóda je komunikácia s expertom v danej problémovej oblasti, teda najmä s používateľom, zákazníkom a/alebo zadávateľom softvérového projektu. Ďalej analytici musia mať prepracovaný systém procesu inžinierstva požiadaviek, aby boli schopní efektívne komunikovať s expertom v problémovej oblasti a získané poznatky vhodným a prehľadným spôsobom zaznamenávať. Podľa článku [2] úspešné projekty vynalozili na inžinierstvo požiadaviek až 28 % svojich zdrojov. S alokáciou zdrojov úzko súvisí prioritizácia jednotlivých požiadaviek. Musíme odstupňovať jednotlivé požiadavky podľa ich dôležitosti, aby sme na relatívne nepodstatnú požiadavku zo strany zákazníka zbytočne neplytvali zdrojmi. Toto všetko by sa nedalo naplniť, ak by

analytici nedocenili zásadnú vec. Analytici musia často a úzko komunikovať so zákazníkom.

Centrálneho správa

Nedocenenie požiadaviek je podľa prieskumu vážny problém pri vývoji informačného systému. Iba 60 percent projektov používalo centrálnu správu požiadaviek, chýb a nedostatkov. Z vlastnej skúsenosti viem, že správa požiadaviek, chýb a nedostatkov spôsobuje vážne komplikácie pri vývoji systému. Centrálna správa umožňuje jednoduchšie a najmä prehľadnejšie zadeľovanie úloh. Manažment projektu, ale aj členovia tímu rýchlo vkladajú do centrálnej správy požiadavky na novú funkcionality, opravu chýb, zmenu atď. Na druhej strane nielen člen tímu má jasný prehľad o jeho úlohe, o ďalšej práci, ktorú má vykonať. Prehľad o dosiahnutých výsledkoch a priebehu vytvárania projektu má taktiež manažment projektu a vrcholový manažment. Pracovník je poloautomatickým spôsobom upozorňovaný resp. oboznamovaný, ako má ďalej postupovať na projekte v rámci tímu a nemusí po každej splnenej úlohe žiadať o novú. Zároveň v takejto centrálnej správe je možné konzultovať jednotlivé požiadavky. Po uzavretí úlohy sa do centrálnej správy zapíše, ako bola uzavretá a nakoľko bola splnená. Centrálny manažment požiadaviek má aj množstvo ďalších výhod, o ktorých sa je možné dočítať napr. na stránke [3].

Manažment rizík a analýza po ukončení projektu

Úspešnosť softvérového projektu tiež závisí od manažmentu rizík a vykonaní analýzy po ukončení projektu. Manažment rizík je najviac zanedbávanou oblasťou manažmentu v softvérovom projekte. Väčšina projektov, ktoré nemali manažment rizík bolo neúspešných. Manažment rizík v priebehu projektu bol pritom podľa prieskumu tiež identifikovaný ako výrazný ukazovateľ úspechu alebo neúspechu projektu.

Iba jedna tretina softvérových projektov vykonala analýzu po ukončení projektu (angl. postmortem review). Podľa môjho názoru tento problém úzko súvisí s vedením podnikovej informačnej databázy otázok a riešení. Takáto podniková databáza uľahčuje prácu zamestnancom. Nemusia pri riešení každodenných otázok znovu „vymýšľať koleso“, ale ak sa už nachádza v takejto centrálnej databáze riešenie, použijú ho. Nevyhnutnou podmienkou využívania centrálnej databázy je jej neustála zmena a dopĺňanie v priebehu riešenia projektov a pri analýze po ukončení projektov. Tým sa tiež docieľajú lepšie zdieľanie vedomostí jednotlivých zamestnancov v rámci spoločnosti.

Záver

Zaručí nám niečo alebo niekto úspech projektu? Nie, nezaručí. Nevie o žiadnom softvérovom projekte, ktorý by poistila poisťovňa proti rizikám. Nevezme na seba riziko neúspechu. Je to pre ňu veľký hazard, lebo ako prieskum ukázal, iba niečo viac ako polovica projektov je úspešných. Tiež neexistuje jednotlivец, ktorý dokáže zaručiť úspech, ak s ním v tíme spolupracujú kolegovia, ktorí nie sú schopní splniť požiadavky na nich kladené. Dalo by sa teda povedať, že úspech je súhra okolností, ktorým je nutné zo všetkých síl napomáhať. Lebo nedostaví sa automaticky, ani veľkým rozpočtom, ani silnou vôľou. Preto sa musíme snažiť čo najlepšie pochopiť a spoznať príčiny úspechu a neúspechu softvérových projektov.

Použitá literatúra

1. Bieliková M.: *Softvérové inžinierstvo - Princípy a manažment*, Vydavateľstvo STU, Vazovova 5, Bratislava, 109-173.
2. Hofmann H. F., Lehner F.: *Requiremenst Engineering as a Success Factor in Software Projects*, IEEE Software, July/Aufust 2001, 58-66.
3. IBM Rational RequisitePro: *Introduction to Requirements Management*, <http://www.pts.com/wp2297.cfm>, 24. 3. 2005.
4. Polášek I.: *Objektovo orientovaný prístup a jeho vlastnosti*, <http://www.gratex.com/Download/OOANS01Postupnost.pdf>, 24. 3. 2005.
5. Verner J. M., Evanco W. M.: *In-House Software Development: What Project Management Practices Lead to Success?*, IEEE Software, Vol. 22, No. 1 (January/February 2005), 86-93.

Annotation

What guarantee success of software project

Almost every software project is special with in usage, development methodology, programming language, and so on. Therefore is hard applying knowledge from one project to another. Despite this, we should try find some useful factories that lead to success or failure of software project. Project management would be able to apply piece of knowledge from analyzing software project from past. What cause success or failure of software projects? What bad do project management? When we will able to better resolve some of causes of failure, then more projects will be success. I try to answer in my essay to these questions and reasons describe to details.