

Plány do nepohody

JAROSLAV BÁLIK

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

Balki[.]balkovic[zavináč]gmail[.]com

Abstrakt. V tvrdom konkurenčnom boji manažéri prijmu aj zákazku, pre ktorú nie sú vo firme vytvorené ideálne podmienky. A preto v reálnych projektoch nie je často jednoduché vytvoriť ucelený plán, ktorý by s prijateľnou mierou neurčitosti odrážal skutočnosť. Dôvodov môže byť viacero. Či už časová tieseň, neistá, niekedy len prisľúbená pracovná sila, alebo nedostatočná znalosť domény projektu. V eseji analyzujem stratégie, vďaka ktorým je možné zmierniť alebo úplne eliminovať problémy, ktoré sú spôsobené samotnou povahou projektu. Ďalej sa zamýšľam nad problematikou preplánovania, respektíve zmeny plánu. Teda kedy je nepresný plán chybný a kedy je nepresnosť integrálnou súčasťou plánovania. Na záver sa pokúšam sformulovať zásady, ktorých by sa mal manažér držať, keď chce vytvoriť flexibilný plán a zároveň si udržať pracovníkov.

Nežijeme vo vzduchoprázdne

Dobrá manažér na základe svojich skúseností vie primerane odhadnúť zdroje a čas potrebné pre úspešné splnenie cieľov projektu. Avšak žiaden projekt nie realizovaný vo vzduchoprázdne. O pracovnú silu musí niekedy manažér bojovať s inými projektmi, ktoré majú vyššiu strategickú prioritu. Prípadne členov tímu zdieľa, alebo je nútený nasadiť menej skúsených ľudí, ktorí sa na projekte takpovediac „učia“. Kvôli obchodným záujmom môžu byť pred manažéra projektu nastolené šibeničné termíny. Takéto okolnosti čoraz častejšie zvyknú komplikovať vytváranie plánov pre projekt. Vo svojej eseji by som rád zanalyzoval známe techniky plánovania a ich použitie v nie celkom priaznivých situáciách. V tejto práci čerpám najmä z článkov autoriek Esther Derby a Johnny Rothman, ktoré sú známe svojou publikačnou činnosťou v oblasti extrémneho programovania.

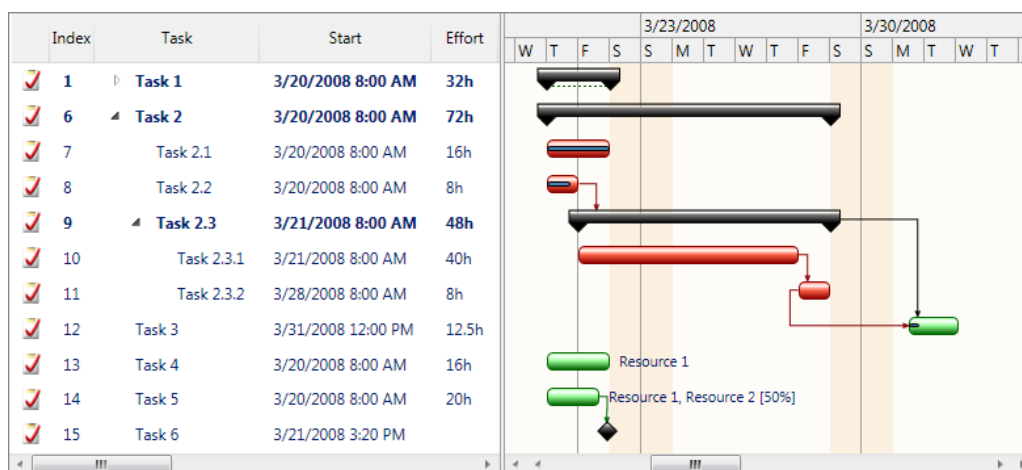
Na úvod začnem citátom. Všimnite si, že pre kuchára aj pre programátora platí to, že naliehanie zákazníka môže ovplyvniť plánované splnenie úlohy, ale nemôže ovplyvniť jej samotné splnenie. Ak sľúbime, že omeleta bude hotová za dve minúty, môže sa zdať, že postupujeme svižne. Ale keď nie je uvarená za dve minúty, zákazník má na výber dve možnosti – počkať si, alebo zjesť ju surovú. Pri softvéri si zákazníci

môžu tiež podobne vyberať. Kuchár má ďalšiu možnosť, môže zvýšiť teplotu. Avšak nakoniec nič nezachráni – výsledkom jeho snaženia je omeleta z jednej strany spálená a z druhej surová [1]. Autor tu poukazuje na fakt, že každý projekt sa dá uskutočniť za takmer ľubovoľný čas, ale ak či už vedome, alebo nevedome plány poddimenzujeme, musíme upustiť od istých kvalitatívnych alebo kvantitatívnych parametrov.

A čo je vlastne ten „plán“ ?

Spýtajme bežného človeka na ulici otázku: „Čo je to plán ?“. Chvíľku sa zamyslí a povie: „Je to predstava o tom, ako si zorganizujeme svoju budúcnosť.“. Každý máme svoje intuitívne chápanie slova plán. Vieme, že keď pečieme tortu, najprv musíme pridať múku, vajíčka, mlieko, cukor, prášok na pečenie a iné prísady. Potom vymiesime cesto, vložíme do zahriatej rúry a necháme upiecť korpus. Korpus rozkrojíme natrieme lekvárom a potom polejeme polevou. Necháme vychladnúť a vložíme do chladničky. Potom prípadne prizdobíme šľahačkou a marcipánom. A môžeme podávať. Kebyže nejaký krok urobíme v nesprávny čas, napríklad pridáme polevu na neupečené cesto, dostali by sme možno požívateľný výsledok, ale nebola by to torta. Je jasné že jednoduchú piškótovú tortu zvládne gazdina v domácnosti, ale sedemposchodovú svadobnú tortu už musí vytvárať tím ľudí.

Ale čo by mal obsahovať plán softvérového projektu? Mnohí manažéri pod plánovaním rozumejú „len“ rozdelenie úloh medzi pracovníkov, závislosti, obmedzenia a ich časovú nadväznosť. Navrhnu prepracovaný Ganttov diagram **Obr. 1**, a sú s ním spokojní. V diagrame jasne vidno následnosť jednotlivých úloh a podmienky ich začatia, respektíve ukončenia, podľa toho, v akom sú vzťahu s ostatnými úlohami.



Obr. 1. Ganttov diagram.

Takýto prístup má ale svoje slabiny. Predstavme si, že nám niektorá úloha mešká a ešte k tomu od nej závisí plnenie iných úloh. Náš pekne zostavený diagram dostáva trhliny a musíme všetky naše starostlivo vytvorené čiarky preusporiadať. Avšak, môže sa stať, že nerozumíme podstate úlohy, aké pohnútky nás k jej vytvoreniu viedli. Potom ťažko vymyslíme preusporiadanie plánu, respektíve substitučnú úlohu. Preto je rozumné si tieto pohnútky zaznamenať a každú úlohu rozumne opísať niekoľkými súvislými vetami. To isté by sme mali spraviť aj pre samotný projekt. Teda napísať plán slovne, aby sme aj po čase vedeli, prečo sme samotný projekt vytvorili. Ganttov diagram je však dobrá pomôcka, netreba ju zavrhať, zide sa nám pri zostavovaní časového rozvrhu. Tu je príklad, ako by mohol vyzeráť plán projektu[4]:

- **Vízia projektu:** Vysvetlenie, prečo projekt existuje.
- **Požiadavky projektu:** Minimum, ktoré treba urobiť. Tieto požiadavky použijeme na definovanie spôsobu, akým môžeme robiť kompromisy. Napríklad máme požiadavku „Odovzdať nejakú časť do 1. júna, aby sme si to mohli vyúčtovať“. Požiadavku na produkt môžu byť vágne (“nejaká časť”), ale termín je nám jasný (1. jún).
- **Ciele projektu:** Čo by sme radi urobili.
- **Kritéria úspechu:** Čo považujeme za úspech, respektíve, za akých podmienok môžeme uvoľniť produkt.
- **Návratnosť investícií:** Ak ju vieme spočítať.
- **Organizácia projektu:** Zahŕňa dátumy, kedy budú dostupní členovia tímu a kedy môžu začať s jednotlivými úlohami, čiže ganttov diagram. Ďalej tu spomenieme typ životného cyklu projektu, niektoré črty metodiky našej firmy.
- **Zoznam rizík:** Spomenieme tie, ktoré môžeme predvídať, a tiež spôsob, ako sa s nimi vyrovnáť.

Takto zostavený plán poskytneme všetkým stranám zúčastneným na projekte, aby sa každý mohol k nemu vyjadriť a prispieť tak svojimi skúsenosťami. Už od začiatku by mal byť plán podrobený niekoľkým iteráciám, kým sa nedospeje k vzájomnému konsenzu. Treba si však uvedomovať, že zákazník bude tlačiť na skracovanie termínov, niekedy až za únosnú mieru. Manažéri, programátori a analytici už väčšinou odhadujú reálnejšie, ale sú optimisti a zvyknú podceňovať časový rozvrh [1]. K tomuto optimizmu ich často vychovali nadriadení, lebo tí neradi počujú slovné spojenia: „Nedá sa“, alebo „To nestihneme“.

Náš pôvodne nahrubo zostavený plán sa postupným procesom iterovania zjemní, avšak musíme brať ohľad na to, aby sme nezabíehali do zbytočných podrobností. Prehnané detailný plán by nás mohol odvieť od podstaty problémov, ktoré svojim projektom chceme riešiť.

Ak nám nie je dostatočne jasná doména projektu, nevieme hneď na začiatku povedať, ako bude vyzeráť plán jednotlivých etáp. Avšak môžeme pristúpiť k zjemňovaniu tej etapy, ktorú práve vykonávame. A keď už sa dostávame do obrazu, čo bude potrebné pre úspešné zvládnutie projektu, môžeme zvyšok projektu začať

zjemňovať, respektíve pristúpiť k preplánovaniu ďalších etáp [5]. Čím viac vieme, tým lepšie odhadujeme, preto sa nemusíme báť meniť svoje odhady [3].

Kritická cesta

Kritická cesta je najdlhší nedeliteľný rad činností, ktorý určuje samotné trvanie dĺžky projektu. V každom projekte je možné identifikovať viacero kritických ciest. Jedna kritická cesta je určená samotným zoznamom úloh. Ďalšia kritická cesta je tvorená ľuďmi ktorí pracujú na projekte. Dostupnosť výrobných prostriedkov tvorí ešte jednu kritickú cestu. Skutočná kritická cesta vedie cez všetky tri zložky, teda cez zoznam úloh, dostupnosť pracovníkov a výrobných prostriedkov [5].

Kritická cesta vyplývajúca zo zoznamu úloh je jasne viditeľná, dá sa vyčítať z gantovho diagramu. Ak však nemáme včas dostupnú pracovnú silu, napríklad človek, ktorého nám „zapožičajú“ z iného projektu je zaneprázdnený, môže sa kritická cesta presunúť na iné úlohy. To isté platí pre prisľúbené výrobné prostriedky. Napríklad ak dodanie databázy druhou firmou oneskorí migráciu dát do nového systému.

Ak ideme plánovať projekt, ktorého doména je nám menej známa, môžeme kritickú cestu iba veľmi matne odhadovať. Veľmi často sa stáva, že sa kritická cesta presúva z jedného radu úloh na druhý.

Avšak aj my môžeme ovplyvniť kritickú cestu, respektíve ju meniť počas projektu. Dá sa ovplyvniť najmä to, kto vykoná akú úlohu, za aký čas [5].

Keď sa nedarí

Každý z nás asi pozná ten pocit, keď sa nedarí a zdá sa, že sa všetko pokazilo. Na človeka sa hrnie hriba čiastkových úloh, v ktorej sa len ťažko dá zorientovať. Pracujeme na plné obrátky, ale zdá sa nám, že z tej hřby neubúda. Pokúšame sa riešiť viacej úloh naraz, ale čím viac ich naraz riešime, tým dlhšie nám to trvá.

Zahlcovanie členov tímu nespočetným množstvom paralelne vykonávaných úloh je všeobecne veľmi zlá taktika. Okrem toho, že pracovník môže od množstva práce vyhorieť, prepínanie medzi jednotlivými úlohami si vyžaduje ďalší čas. Informatik, ako duševne pracujúci človek sa potrebuje do svojej úlohy zahĺbiť [2]. Keď sa potrebuje zanoriť niekoľkokrát denne, zaberie mu to značnú časť pracovného času.

Pri veľkom množstve činností, bývajú úlohy slabozadefinované. Riešiteľ sa musí zdĺhavo zamýšľať, čo sa od neho vyžaduje. Pokiaľ je takých úloh priveľa, treba sa spýtať:

- Kto úlohu zadal ?
- Prečo ju zadal ?
- Aký je prínos tejto úlohy, bez ohľadu na prioritu ?
- Vieme úlohu opísať aspoň tromi súvislými vetami ?

Často zistíme, že nevieme odpovedať na všetky otázky. Úlohy, ktoré sú zmätočné, sa dajú takýmto spôsobom eliminovať.

K tomuto problému však existuje aj lepší prístup. Ak členovi tímu predložíme úlohu operatívneho plánu (plánu na každý deň) rámcovo, on sám lepšie identifikuje, čím je potrebné sa zaoberať. Potom postupne, v spolupráci s ním plán zjemňujeme a sledujeme, ako mu sa darí rámcovú úlohu plniť.

Zmena plánu

Predstavme si situáciu, keď máme síce jasne zadefinované jednotlivé úlohy, poznáme termíny, ale proste sme v sklze. Môžeme si vybrať, či budeme:

- Vypúšťať funkcionality.
- Posúvať termíny.
- Nútiť členov tímu pracovať nadčas.
- Pridať ďalších ľudí do projektu.
- Preusporiadať plán.

Posunutie termínov zaručene nebude vyhovovať zákazníkovi, ak vytvárame projekt na zákazku. Ak vytvárame generický softvér, môže to ohroziť naše pozície na trhu.

Členovia tímu môžu viacej pracovať, aby sme nezaostávali za časovým plánom. Ale dlhodobou prácou nadčas môžeme spôsobiť uštvanie ľudí. Tí to potom fyzicky, alebo psychicky neunesú a odídu s firmy.

Pridanie ďalších ľudí do projektu pomôže len vtedy, ak vieme včas meškanie identifikovať, rádovo mesiace vopred. Inak, ak pridáme ľudí do spomaleného projektu, tak sa projekt ešte viac spomalí [1].

Ak však poznáme podstatu projektu a jednotlivých úloh, môžeme si ich preusporiadať, respektíve rozdeliť na viacero etáp. Pomôže nám to predísť nechcenému zdržaniu, alebo zvýšeniu nákladov[5]. Príklad pôvodného plánovania etáp jednoduchej aplikácie skladajúcej sa z fakturačného účtovného modulu ukazuje Tabuľka 1.

Tab. 1. Pôvodný plán.

Termín ukončenia	Názov úlohy
1.2.2008	Implementácia
10.3.2008	Automatické testy
5.4.2008	Integračné testovanie

Avšak vieme, že fakturačný modul sa dostane do sklzu lebo jeden z programátorov dostal chrípku. Členovia tímu začínajú byť nervózni, pristúpime teda k preplánovaniu časti projektu. Vieme že modul fakturácie môžeme rozdeliť na separátnu funkcionality,

teda odberateľské a dodávateľské faktúry, príklad zmeneného plánu uvádza **Tabuľka 2**.

Tab. 2. Upravený plán.

Termín ukončenia	Názov úlohy
1.2.2008	Implementácia účtovného modulu, Implementácia fakturačného modulu – časť dodávateľské faktúry.
10.3.2008	Automatické testy účtovného modulu, Automatické testy fakturačného modulu – časť dodávateľské faktúry. Implementácia fakturačného modulu – časť odberateľské faktúry.
5.4.2008	Automatické testy fakturačného modulu – časť odberateľské faktúry , Integrované testovanie systému.

Príklad je síce jednoduchý a neodzrkadľuje celý plán projektu so všetkými etapami, ale názorne ukazuje, ako je možné pristúpiť k preplánovaniu a rozdelením etáp na jednotlivé časti. Takto môžeme postupovať nielen vtedy, keď projekt mešká, ale aj keď vieme, že sa nám uvoľnia zdroje.

Ako by teda mal vyzerat' flexibilný plán ?

Na záver, by som chcel ako zhrnutie sformulovať niekoľko rád, ktorých by sa mal manažér držať, ak chce vytvoriť flexibilný plán projektu a zároveň nechce, aby od neho utiekli všetci jeho pracovníci.

1. Keď píšeme plán projektu, mali by sme ho slovne zaznamenať na papier, aby sme pochopili jeho podstatu.
2. Ganttov diagram je dobrá pomôcka, avšak by nemal slúžiť ako jediný spôsob pre zaznamenávanie plánu.
3. Pri plánovaní netreba zabiehať do zbytočných podrobností. Príveľmi podrobne rozpracovaný plán nás dokáže odviesť od skutočnej podstaty projektu.
4. Ak je nám doména projektu nejasná, uskutočňujeme podrobné plánovanie etapy projektu, až keď nám je jasný priebeh predchádzajúcej etapy.
5. Preplánovanie v projekte si môžeme dovoliť, ak dobre poznáme podstatu jednotlivých úloh. Jedine vtedy vieme určiť , ktoré úlohy sú substituovateľné a ktoré úlohy sa dajú rozdeliť na viac etáp.
6. Kritickú cestu môžeme presúvať medzi úlohami. Tým, že preskupíme ľudí, ktorí majú jednotlivé úlohy vykonávať.

Vo svojej eseji som vychádzal z premisy, že svet nie je dokonalý. A plány, ktoré sa v praxi zostavujú, dosť často nebývajú práve ideálne. Postrehy, načrtnuté v tejto eseji nemajú ambíciu byť všeliakom na problémy spôsobené manažérmi, ktorí sa usilujú získať projekt za každú cenu. A to bez dostatočných zdrojov. Aj takéto projekty majú svoju šancu na prežitie, ak použijeme zmenu prístupu k plánovaniu. Niektoré projekty nezaniknú, ani keď sú očividne zle naplánované. V takomto prípade som sa pokúsil načrtnúť, cestu, ktorou je možné ísť, keď chceme zmierniť „utrpenie“ ľudí pracujúcich na projekte.

Tak trochu som zamlčal, respektíve nezdôrazňoval som, že išlo najmä o techniku iteratívneho plánovania. Avšak nebolo to mojím zámerom. Rady, ktoré som spomenul, sa dajú využiť aj pri iných technikách.

Použitá literatúra

1. Brooks, F.P., *The Mythical Man-Month: Essays on Software Engineering*. Anniversary Edition. Addison Wesley. 1995.
2. Esther Derby: *Where does your time go?*
<http://www.estherderby.com/wheredoes.htm> [cit. 2008-14-10]
3. Pressman, R.S., *Software Engineering: A Practitioner's approach*. 5th Edition. McGraw Hill. 2001.
4. Johanna Rothman: *Don't depend on a work breakdown structure to keep your project on target*. <http://www.ddj.com/architect/184414983> [cit. 2008-14-10]
5. Johanna Rothman: *Iterative software project planning and tracking*
<http://www.jrothman.com/Papers/7ICSQ97.htm> [cit. 2008-14-10]

Annotation

Planning in hard times

This essay focuses on topic of project planning in conditions, that are not ideal for software development. Author explains how to overcome obstacles, that might be caused by several reasons. These reasons might be – underestimation of efforts, insufficient knowledge of project domain, or time pressure. In such conditions, there is not easy to create comprehensive and reasonable project plan. In essay author tries to analyze strategies that can reduce, or eliminate problems, which are caused by inappropriate project planning. Author also discusses the topic of replanning and inexactness in planning. In the conclusion of essay are mentioned theses which can be useful for creating flexible project plan.