

DOKÁŽEME NAPLÁNOVAŤ NÁŠ PRVÝ TÍMOVÝ PROJEKT?

Dva krát meraj a raz strihaj. Hej, počkať, zobrali sme meter?

Roman Burger

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
roman.arnold.burger[zavináč]gmail[.]com

Abstrakt. *Nie je ľahké začať robiť niečo, čo sme doteraz nerobili. Nie je ľahké sa naučiť niečo nové. A najmä, nie je ľahké zistiť, či to robíme správne. Mali by sme si preto dávať zvlášť pozor a dať si záležať, aby sme skutočne pochopili podstatu vecí. V tejto eseji rozoberiem moje skúsenosti s tvorbou softvéru v začínajúcom tíme. Ukážem, ktoré veci sú náročné, nech už ste akokoľvek dobrý, a tiež ukážem, ktoré veci sa dajú ľahko obísť s vhodným prístupom.*

Kľúčové slová: *plán, tím, učenie, problémy, odhady, vývoj*

Úvod

Že vraj znak múdrosti je, ak sa najprv zamyslíme pred tým, než niečo spravíme. Niečo na tom naozaj bude. Dokážeme sa tak vyhnúť plno problémom, prípadne prísť s elegantnými riešeniami. Kto nám však zaručí, že sa zamyslíme správne? Že na to používame správne prostriedky?

Keď zostavíte svoj prvý tím a pokúsite sa vytvoriť už aj nejaký väčší projekt, asi sa začnete obhliadať po skúsených tímoch po nejaké to „know-how“. Skoro určite vám budú tvrdiť, že plánovanie projektu je jeden z najdôležitejších a najkritickejších procesov pre váš projekt. A zatiaľčo je to vskutku pravda, už nie tak často počuť, ako by ste mali začať. Niet sa čo čudovať, keďže vo svojich tímoch pracujú už nejaký ten čas a ich rady tým určite budú ovplyvnené.

Čo obnáša plánovanie?

Aby sme si rozumeli, musíme sa najprv dohodnúť, čo budeme rozumieť pod pojmom plánovanie. Plánovanie si môžeme predstaviť ako určenie si cesty a označenie míľnikov na nej. Je to cesta úsilia v našom projekte, kde míľniky sú konkrétne výsledky práce.

Ako si teda túto cestu určíme? V každom momente na našej ceste potrebujeme vedieť kade ďalej, čo ďalej. To znamená, že potrebujeme mať cestu úsilia rozdelenú na dostatočne jednoduché a jasné úlohy - potrebujeme dekomponovať problém. Aby sme mohli z týchto úloh vytvoriť celú cestu, musíme vedieť ako dlho budú trvať, čiže kedy môže na ceste začať ďalšia úloha. Potrebujeme teda odhadnúť náročnosť všetkých úloh. Ak už vieme aké úlohy treba a koľko budú trvať môžeme ich usporiadať do kompletnej cesty. Aby sme si však boli istí, že usporiadané úlohy sú počas projektu stále usporiadané správne, potrebujeme monitorovať dodržiavanie vytýčenej cesty a reagovať na vychýlenia.

Problémy pri plánovaní

Brooks [2] píše, že viac projektov zlyhalo kvôli zlému plánovaniu než kvôli zvyšným dôvodom dohromady. Brooks uvádza 5 hlavných dôvodov, prečo je to tak:

- Slabé techniky odhadov.
- Techniky odhadov si mýlia úsilie na projekte so stavom projektu.
- Nereálna predstava pre odhady.
- Dodržiavanie plánu je nedostatočne monitorované.
- Zlá reakcia na chyby v pláne.

Prvé tri dôvody súvisia s odhadom, zvyšné dva s manažovaním plánu. Môžu teda tieto problémy ohroziť práve náš projekt, ak si na ne dáme pozor? Mohli by sme sa predsa naučiť techniky odhadovania a naštudovať si dôležité rady v rámci manažmentu plánu.

Vhodnou a silnou technikou odhadovania je aj odhad, založený na štatistických dátach z predchádzajúcich projektov [5]. Nech si však vyberieme ľubovlnú techniku (rozdiely v nich však nie sú až také veľké), jedno majú všetky techniky odhadu spoločné, vyžadujú si skúsenosti. Ak napríklad chceme odhadnúť zložitost danej úlohy podľa nejakej všeobecnej metriky, musíme mať v tíme človeka, ktorý má s daným typom úlohy skúsenosti. Toto nemusí byť nutne problém, ak by sme vedeli zohnať tých správnych ľudí. Bohužiaľ však nevieme, ako dobre budú spolupracovať. V tímových projektoch nie sú totiž aj tie jednoduchšie úlohy otázkou len jedného človeka, len na jedno odpoľudnie. Nedokážeme preto dobre odhadnúť časovú náročnosť danej úlohy v rámci nášho projektu, nedokážeme odhadnúť potrebné úsilie práve nášho tímu.

Jedným z jednoduchých, ale zato efektívnych spôsobov, ako sa vyhnúť problémom pri plánovaní je, paradoxne, počítanie s problémami pri plánovaní. Dosiahneme to naplánovaním si rezervy v pláne, napríklad pomocou dvoch plánov [1]. Získame tak čas reagovať na nepredvídané udalosti a hlavne dáme rezervu pre naše odhady. Na prvý pohľad to vyzerá rozumne, čo ak ale ani rezerva nebude stačiť. Vieme jej veľkosť odhadnúť? Problém je, že chceme použiť odhad na to, aby sme zmiernili dopad slabého odhadu. To už asi nezní tak rozumne.

Nechcem však povedať, že je zbytočné používať dobré techniky, len preto, že pre ne nemáme najlepšie predpoklady. Dobré techniky ostanú dobrými technikami. Dokonca nevravím ani to, aby ste prestali robiť odhady. Odhadujte, plánujte, riadte váš projekt. Ak však nie ste dostatočne presvedčený o správnosti odhadov, nenechajte ľudí spoliehať sa na ne (a najmä sa na ne nepoliehajte vy sami).

Ako obísť to, čo sa dá obísť

Vieme teda, že naše odhady v našom prvom projekte asi nebudú ani na úrovni našej vlastnej spokojnosti. Odhady sú dôležitou súčasťou plánovania, čo však s dekompozíciou a monitorovaním dodržiavania plánu? Musia tiež trpieť v novom tíme?

Použitím vhodných prístupov k vývoju softvéru si môžeme tieto činnosti značne zjednodušiť a zautomatizovať. V tomto smere vhodným prístupom je iteratívna, inkrementálna metodika vývoja softvéru. Podstata tejto metodiky je v dodávaní čo najmenších (alebo podľa dohody) funkčných inkrementoch systému. Často sa totiž stretávame s problémom, kde sa požiadavky zákazníka menia v priebehu vývoja. Dokonca nie je ojedinelé, že svoje požiadavky ani nevie poriadne vyjadriť, dokým neuvidí funkčnú predstavu vývojárov. Je preto vhodné postupovať po malých kúskoch, kde sa môže po každom kúsku predstaviteľ zákazníka vyjadriť, ako sa mu výsledok páči. Nazývame to vývoj orientovaný na zákazníka. Ako nám to však pomôže?

Pokiaľ chceme dodávať v pravidelných časových úsekoch (napríklad metodikou Scrum) malé funkčné inkreментy, musíme vedieť rozdeliť základné požiadavky na dostatočne malé úlohy. Zároveň to znamená, že pri každom dodaní sa okamžite ukáže akýkoľvek sklz v pláne [3]. Ak teda vieme vyvíjať softvér, vieme si takýmto implicitným spôsobom zjednodušiť a zabezpečiť aj plánovanie použitím vhodnej metodiky.

Otázka, ktorá by nám však mala ihneď napadnúť je, či je možné alebo vhodné použiť túto metodiku pre nový tím? Existujúce štúdie ako [3] a [4] ukazujú, že áno. Dokonca väčšina členov tímov tvrdí, že sa im agilná metodika páčila budú ju chcieť aplikovať aj vo svojich ďalších projektoch.

Trénovania plánovania

Ak teda začíname pracovať v novom tíme, nemusíme vôbec plánovať? To určite nie je správna implikácia. Je zrejmé, že ak by sme sa nikdy neučili plánovať, nenaučíme sa plánovať vôbec. Treba mať však stále na pamäti, že plánovanie je proces, ktorý má svoju cenu a svoju hodnotu. Ak hodnota pre náš tím nie je adekvátna cene, určite tým utrpí náš projekt.

Najrozumnejším prístupom sa ukazuje analyzovanie a zhodnocovanie po ukončení úsekov. Schválne píšem o úsekoch. Analyzovať a zhodnocovať by sme mali po ukončení každého projektu, každej iterácie, každej úlohy. V podstate každého úseku, u ktorého je šanca, že sa bude rovnaký typ znova vykonávať. Robíme to preto, aby sme mohli náš postup (naš proces) zlepšiť a nabudúce tak môžeme byť lepší, rýchlejší alebo lacnejší. Práve preto je ukončenie úseku pre učenie sa nových znalostí vhodná doba. Ak sa teda bavíme o tom, že naše plánovacie schopnosti sú značne obmedzené (napríklad slabými

4 Roman Burger

odhadmi), je práve čas medzi dvomi úsekmi vývoja ten správny čas na to, aby sme zlepšili naše plánovanie.

Rozdiely v procesoch plánovania medzi dvomi projektami by sa mali meniť až dokým nie je naše plánovanie takmer dokonalé. Nie je však vylúčené, aby zmeny nastávali aj častejšie, napríklad po každej iterácii na projekte. Navyše nič nám nebráni skúšať si plánovať a najmä odhadovať samých pre seba a učiť sa tak. Ako som však spomínal v kapitole problémov plánovania, nemali by sme nechať iných spoliehať sa na naše odhady pokiaľ si za nimi nebudeme stáť.

Záver

V tejto eseji som sa snažil priblížiť vývojárom, ktorí by chceli začať pracovať v tíme, že nie je také jednoduché začať. O to viac by sa mali dávať pri vývoji pozor a najmä by sa mali snažiť vždy pochopiť podstatu vecí. Ak sa chceme naučiť niečo nové, musíme tomu najprv porozumieť. Inak oklameme iba samých seba.

Použitá literatúra

1. Armour, P.G.: The Business of Software To Plan , Two Plans. Communications of the ACM, 48(9):15–19, 2005.
2. Brooks, F.P.: The mythical man-month. In Proceedings of the international conference on Reliable software, pages 193–, New York, NY, USA, 1975. ACM.
3. Rising, L., Janoff, N.: The Scrum software development process for small teams. Software, IEEE, 17(4):26–32, 2000.
4. Schild, J., Walter, R., Masuch, M.: ABC-Sprints: adapting Scrum to academic game development courses. In Proceedings of the Fifth International Conference on the Foundations of Digital Games, pages 187–194. ACM, 2010.
5. Spolsky, J.: Evidence Based Scheduling. Dostupné na internete: <http://www.joelonsoftware.com/items/2007/10/26.html>

Annotation

Can we plan our first team project?

It is not easy to do something we haven't done yet. It is not easy to learn something new. And especially, it is not easy to determine if we are doing it the right way. Therefore we should be extra careful and diligent to understand the core of things. In this essay I will discuss my own experience in software development in newly assembled team. I will show which processes in software planning are hard at first even if you consider yourself skilled and I will also show which processes could be done easier with the appropriate approach.