

Plánovanie a spokojnosť

MATEJ LABAŠ

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

Abstrakt. V procese tvorby softvérového projektu hrá plánovanie činnosti a práce jednu z najpodstatnejších úloh. Nie je nič horšie ako pracovať na projekte niekoľko mesiacov a zistiť, že sa za ten čas skoro nič nevykonalo a termín odovzdania sa neúprosne blíži. Preto by som chcel v nasledovnom texte poukázať na možnosti plánovania v projekte a hlavne také, ktoré prinesú spokojnosť ako vývojárom, tak aj zákazníkom. To zahŕňa proces rozdelenia práce na časové úseky, rozdelenie projektu na moduly a v neposlednom rade rozhodnutia, kedy a čo ukázať zákazníkom, prípadne vydať. Tieto procesy je potrebné vykonať tak, aby bol plán dostatočne motivujúci pre tím, a výstupy práce tímu zas dostatočne uspokojujúce pre zákazníkov.

Úvod

Každý softvérový projekt začína v podstate rovnako. Zákazník zadá firme alebo tímu úlohu, špecifikuje svoje požiadavky. Tím tieto informácie musí zanalyzovať a navrhnúť pre ne riešenie. Následne prichádza na rad samotná tvorba navrhnutého systému. A to býva najčastejšie kameň úrazu.

Každý, kto už niekedy pracoval na vývoji softvérového systému v tíme mi dá za pravdu, že bez poriadneho plánu to jednoducho nejde. Nielen že členovia tímu nevedia, čo skôr robiť a aké úsilie danej časti vynaložiť, zároveň ani nemajú žiadnu motiváciu vôbec pracovať. Menežment a zákazníci tiež nemajú šancu monitorovať priebeh vývojového procesu.

V tomto článku by som preto rád rozobral niektoré časti plánovacieho procesu, ktoré považujem za dôležité a niektoré postupy, ktoré ma zaujali. Prejdem od základov plánovania ako takého, až po spravovanie plánu ako dynamickej časti projektu podliehajúcej rôznym faktorom a meniacej sa počas vývoja.

Ako začať

Ako tvorca plánu musí človek zvážiť všetky dostupné informácie. Máme zadanú úlohu, tím, ktorý ju má vykonať, a časový horizont, do ktorého má byť práca dokončená. Všetky tieto ukazovatele nám predurčujú smery, ktorými sa náš plán môže uberať. Ak máme menší tím, ale projekt má veľa častí, odrazí sa to prirodzene na čas a podobne. Ďalšie faktory sú rôzne obmedzenia, hlavne technického charakteru a s tým súvisiaceho rozpočtu pre projekt – nie vždy sa dajú zabezpečiť všetky technické podmienky, ktoré by boli najvhodnejšie.

Preto je treba na začiatku urobiť nasledovné rozhodnutia:

Rozdelenie členov tímu

Tím by sa mal skladať z dvoch hlavných typov členov. Tým prvým typom sú manažeri (alebo *riadiaci tím*), to jest ľudia, ktorí sa starajú o komunikáciu s klientami a vývojármi, spravujú plán projektu, revidujú a akceptujú výsledky jednotlivých častí vývoja. Sú zodpovední za identifikovanie cieľov projektu, vybranie členov tímu pre jednotlivé role, mali by rozhodnúť o potrebných zdrojoch (technických či finančných) a dokázať ich sprostredkovať, sledovať prácu a pokroky tímu, poskytovať výstupy pre klientov prípadne nadriadených za každú úspešne ukončenú časť plánu.

Druhým typom sú vývojári (alebo *projektový tím*), čo sú ľudia implementujúci riešenia. Mali by byť rozdelení do niekoľkých skupín pracujúcich samostatne na rôznych častiach projektu, v závislosti od jeho rozsiahlosti. Napríklad jedna skupina sa stará len o dizajn, kým ďalšia o vnútorné funkcie a podobne [1]. Jednou z charakteristík vývojárov je, že čím majú viac času na vykonanie určitej úlohy, tým ju budú viac odkladať. Na druhej strane, ak majú času málo, bývajú frustrovaní a to sa tiež odrazí na ich výkonnosti. Preto je takéto veci dobré sledovať pri plánovaní časových jednotiek.

Rozdelenie práce

Keď máme vývojárov tímu rozdelených na skupiny, ktoré majú každá nejaké silné stránky, je treba im zadať podúlohy z hlavného projektu. Tu treba brať do úvahy práve to, čo každá skupina vývojárov dokáže najlepšie urobiť, aby sme maximalizovali výkon. Čím viac sa tímu darí, tým viac sú aj členovia motivovaní a lepšie sa im pracuje.

Každá skupina teda dostane na starosti modul alebo moduly – sú to časti projektu, ktoré sa dajú vyvíjať nezávisle, aby sa neskôr spojili do jedného celku. Je to aplikovanie metódy *divide and conquer*, ktorá je všeobecne známa a podporovaná. Každý modul je reprezentovaný aj na časovej osi, to znamená že má určený čas, kedy by mal byť dokončený, aby sa mohlo začať pracovať na nasledujúcom (ktorý na ňom môže byť napríklad závislý).

Pri rozdeľovaní projektu na časové jednotky je dobré sledovať pár dôležitých bodov. Môže to byť obtiažne, ale treba objektívne zhodnotiť dôležitosť modulu a tak mu priradiť adekvátny čas. Ak je modul príliš rozsiahly, je dokonca lepšie ho rozdeliť na ešte menšie časové jednotky, aby sa zachoval prehľad o jeho vývoji. Na konci každého

časového obdobia musí skupina vývojárov odovzdať hlásenie o postupe, aby sa dalo urobiť vyhodnotenie doterajšieho postupu a tieto informácie aplikovali na ďalšiu prácu.

Dynamický plán

Po dokončení úvodných krokov začína práca na projekte. Ak máme dobrý plán, všetko by malo ísť ako po masle. Stále je tu však veľký priestor na rôzne zmeny počas vývoja. Človek nikdy nevie, čo sa môže stať a zrazu príde zmena, ktorá ovplyvní časť plánu. Takúto situáciu treba následne promptne riešiť tak, aby sa zachovala kontinuita práce na projekte.

Preto sme si v úvode rozdelili náš plán na časti. Každá časť je navrhnutá tak, aby sa jej vývoj a výsledky dali hodnotiť nezávisle, a takýmto spôsobom postupovať plánom. Ak by sa stala udalosť, ktorá výsledok časového bloku alebo modulu zmení tak, že nebudú podľa očakávaní, dá sa v tomto prípade plán upraviť.

Takýto spôsob vykonávania plánu (možno označiť skratkou PADRE [1]) obsahuje tri hlavné úrovne – úroveň projektu, úroveň časového bloku a úroveň modulu. Všetky tri úrovne sa rozdeľujú na tri najdôležitejšie fázy – a to *plánovanie*, *vykonanie* a *vyhodnotenie*. Pohľad z každej úrovne umožňuje sledovať úroveň pod ňou z dostatočného nadhľadu, aby sa dali zbadáť problémy a takisto ich aj adekvátne riešiť.

Úroveň projektu je nadradená nad úroveň časových blokov. Preto v časti plánovania projektu robíme práve toto rozdelenie. Na tejto úrovni je tiež dôležité navrhnúť štandardy a odporúčania, ktoré treba dodržiavať pri tomto projekte, aby sa zachovala kvalitná práca. Tento pripravený plán sa predloží tímu vývojárov, ktorý ho schváli alebo navrhne prípadné zmeny.

Vykonaním na tejto úrovni sa rozumie vykonanie každého časového bloku. Menežeri počas tohoto vykonávania zabezpečujú potrebné zdroje a odstraňujú prekážky.

Na konci je dôležité celý projekt vyhodnotiť a nájsť prípadné vylepšenia plánu, výsledku práce a tiež tvorivého procesu.

Úroveň časových blokov je nadradená modulom. Každý časový blok môže obsahovať viac modulov, na ktorých sa dá pracovať paralelne. Ako som už spomínal vyššie, každý modul je pridelený nejakej skupine vývojárov, ktorá na ňom počas daného obdobia pracuje. Tento plán a rozdelenie je samozrejme pred aplikovaním treba schváliť na vyššej úrovni.

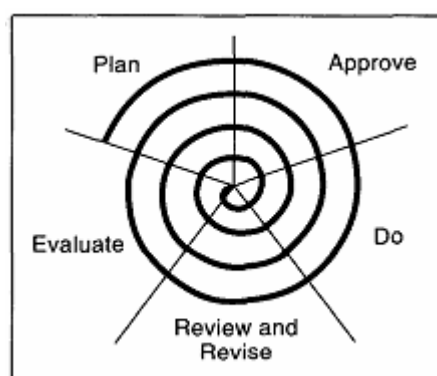
V časti vykonania sa vývojári venujú modulom, ktoré im boli pridelené – vykoná sa úroveň každého modulu. Po dokončení kompletného plánu tohoto časového bloku je treba výsledok ohodnotiť a schváliť – vedením a hlavne zákazníkmi.

Na záver časového bloku je treba zhodnotiť všetku urobenú prácu na moduloch v tomto období a prípadne nájsť chyby a vylepšenia.

Úroveň modulov je najnižšia úroveň. Tu sa priamo implementujú riešenia vývojármi. Naplánuje sa detailný návrh implementácie a návrh testov pre daný modul

a po odsúhlasení sa modul podľa neho naprogramuje. Po ukončení práce na module je samozrejme zhodnotený a nájdené prípadné nedostatky a príležitosti na zlepšenie.

Tento postup by sa dal jednoducho označiť ako životný cyklus plánu. Na obrázku 1 je ukázaný takýto cyklus v špirálovom znázornení. Jasne je vidieť, že počas vývoja sa každou fázou prejde niekoľko krát. Týmto postupom si tím zaručí, že v každom okamihu bude mať jasný prehľad o tom, čo sa urobilo a čo nasleduje. A prehľad je práve to, čo je dôležité pre každého člena tímu, keďže čím jasnejšiu má predstavu o tom, čo má robiť, tým lepšie to vykoná a bude mať aj oveľa väčšiu motiváciu.



Obr. 1 Špirálový životný cyklus plánu [1]

Zákazníci a verzie

V predchádzajúcej kapitole sme si ukázali, ako pomôcť tímu. No dôležitou časťou plánovania je aj komunikácia so zákazníkmi. Nie je nič horšie, ako v polovici projektu zistiť, že zákazník chcel v skutočnosti niečo úplne iné ako ste navrhli a implementovali.

Preto je treba dôkladne naplánovať, kedy predložiť ktorú časť projektu na posúdenie samotným zákazníkom. Je dobré nájsť také body plánu, ktoré sú od seba časovo približne rovnako vzdialené, a zároveň spájajú čo najviac blokov – či už časových alebo modulových. V tomto časovom bode zákazníkovi predložíte viac ukončených častí, ktoré s ním treba do detailov prebrať kvôli spätnej väzbe.

Ak sa projekt už dostane do štádia, kedy ho tím dáva k dispozícii na testovanie priamo používateľom, je dobré plánovať každú verziu tak, aby bola aplikácia celistvá. Je preto dôležité držať sa podobného pravidla ako predtým – nájsť taký bod plánu, kde je čo najviac častí ukončených.

Ďalšia vec je, že nie vždy idú udalosti tak, ako by sme očakávali. Nájdenie nečakaného problému, zmena verzie niektorého z podporných systémov, zmena hardvéru... Toto všetko môže zapríčiniť to, že sa budú musieť urobiť zásahy do projektu. A tieto zmeny je treba otestovať, prípadne predložiť zákazníkovi v termíne, ktorý

nezapadá do plánu. Našťastie, štruktúra plánu akú sme si navrhli vyššie nám dovoľuje zasahovať do jednotlivých častí projektu separátne a riešiť podobné situácie bez problémov [2].

Úpravy plánu

Predstavme si, že sme uprostred vývoja projektu. Všetko beží ako má a podľa plánu, ktorý sme si navrhli na začiatku. Ale nečakane príde telefonát, že schôdzka so zákazníkmi nedopadla príliš uspokojivo. Niektoré z hlavných častí špecifikácie boli zmenené a je na nás, aby sme na ne reagovali.

Toto je len jeden príklad, kedy je potreba zobrať plán a urobiť v ňom zmeny. Iné dôvody môžu byť napríklad zlý odhad pri rozdelení práce, ktorý spôsobí predĺženie vývoja modulu – čo samozrejme posunie časový harmonogram. Ďalšie, napríklad personálne zmeny v tíme, čo sa bohužiaľ nedá len tak predvídať, alebo aj napríklad vývoj nových technológií, ktoré môžu vývoj zrýchliť, ale naopak aj spomaliť.

Iný dôvod zmien v pláne je priamo práca vývojárov. Často sa stane, že keď sú dva moduly navzájom prepojené a ich tvorcovia pracujú podľa vlastných plánov, vyskytnú sa konflikty medzi nimi. Zrazu obe skupiny vývojárov potrebujú prístup k modulu, ktorý je prepojený s ich vlastným, a vyžadujú v ňom zmeny. Takýto konflikt bráni vývojárom v práci a znižuje ich produktivitu. Preto je potrebný systém na kontrolu zmien plánov v projekte [3].

Jeden zo spôsobov riešenia je vyjednávanie [3]. Členovia tímu predložia návrhy zmien plánu. Menežer zodpovedný za spravovanie zmien ich analyzuje, zistí vzťahy medzi dotýčnými časťami projektu a rozhodne čo ďalej. Ak nie je možné vykonať požadovanú zmenu okamžite, môže menežer návrh na zmenu zamietnuť. Ak ho akceptuje, zostaví sa zoznam konfliktov a vývojári vyjednávaním medzi sebou rozhodnú, čo bude najvhodnejšie v tejto situácii urobiť. Závisí to hlavne od dopredu naplánovaných postupov riešenia projektu. Takto upraví plán danej časti až do bodu, kedy neobsahuje žiadne konflikty.

Aby sme predišli problémom pri riešení možných zmien plánu, môžeme tieto faktory analyzovať už pri jeho samotnej tvorbe. Jenda z možností je simulácia plánu [4].

Pri tvorbe počiatočného plánu sa vytvorí niekoľko rôznych verzii. Hlavne keď na ňom pracuje viac ľudí, určite dôjde ku konfliktu záujmov a je treba zvoliť ten plán – alebo kompromis medzi plánmi – ktorý je najlepší. Tu nám práve dobre poslúžia simulácie. Na časti plánu, ktoré sa zdajú kritické, sa budú aplikovať rôzne situácie a menežeri sa pokúsia ich riešiť v súlade s daným plánom. Sú to hlavne situácie takého typu, aké som spomínal vyššie – vyžadujú priamy zásah do plánu. Riešenia týchto situácií sú zaznamenané, vyhodnotené a simulácie plánov porovnané. Vyhodnotia sa hlavné faktory, ako sú cena, kvalita výsledku a časová náročnosť.

Simulácie môžeme použiť aj pri už nutnej zmene plánu. Tak isto sa môže stať, že prichádzajú do úvahy viaceré alternatívy zmien. Preto znova aplikujeme na plán rôzne kritické situácie a uvidíme, ktorá verzia nám lepšie poslúži.

Záver

Ako plánovať, aby boli všetci zúčastnení spokojní? Ťažká otázka. V tejto eseji som sa pokúsil zhrnúť oblasti plánovania, ktoré by nemali byť opomenuté a ktoré určite dokážu pomôcť. Či už menežerom, starajúcim sa o čistý chod projektu, alebo vývojárom, ktorí chcú mať čo najväčší prehľad v tom, čo robia, alebo zákazníkom, ktorí očakávajú splnenie všetkých svojich požiadaviek.

Samozrejme, každý projekt je iný a dajú sa na neho aplikovať rôzne postupy. Preto je vždy výhodné nechať si otvorené možnosti vo vašom pláne, keďže svet sa mení a tak ako môže prísť situácia ktorá projektu pomôže, veľmi ľahko môže prísť situácia opačná. A tu treba ako schopný plánovač vedieť flexibilne zareagovať, aby projekt neskončil fiaskom.

Použitá literatúra

1. Rettig, M., Simons, G.: A project planning and development process for small teams. *Communications of the ACM*, Volume 36, Issue 10 (October 1993), ACM Press
2. Saliu, M. O., Guenther, R.: Bi-objective release planning for evolving software systems. *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering ESEC-FSE '07*, September 2007, ACM Press
3. Okamura, K.: Combining local negotiation and global planning in cooperative software development projects. *Proceedings of the conference on Organizational computing systems COCS '93*, December 1993, ACM Press
4. Setamanit, S., Wakeland, W., Raffo, D.: Project management: Planning and improving global software development process using simulation. *Proceedings of the 2006 international workshop on Global software development for the practitioner GSD '06*, May 2006, ACM Press

Annotation

Planning and satisfaction

Planning of work is one of the most essential tasks in the software project creation process. There is nothing worse than to work on a project for several months and notice, that there has been basically nothing done up to now and that the deadline is coming close. That's why I would like to show a few possibilities of planning in a project, and especially those, that bring satisfaction to developers and end customers alike. This includes the time scheduling, work distribution to modules and last but not least, deciding when and what to release to the customers. These processes are meant to be done in a way that will make the plan enough motivating for the team, and the results of the work also enough satisfying to the customers.