

Tvorba plánov v softvérovom projekte, rozdelenie úloh, plnenie a aktualizácia plánov projektu

RASTISLAV MASARYK

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

`rastislav[.]masaryk[zavináč]hlohovec[.]net`

Abstrakt. V tejto eseji chcem vyjadriť môj názor na problematiku tvorby plánov v softvérovom projekte a na rozdelenie úloh v tíme. Existuje viacero spôsobov, ako vytvoriť plán projektu. V tejto eseji sa pokúsim priblížiť jednoduchý vodopádový model plánovania. Pri riadení softvérového projektu nie je dôležité len vytvorenie dokonalého plánu, ale aj správne rozdelenie úloh jednotlivým členom tímu. Toto rozdelenie je veľmi dôležité aj pre dodržiavanie plánu a na vytvorenie spokojnosti a dobrej pracovnej atmosféry v tíme. Preto treba zohľadniť schopnosti, skúsenosti a preferencie členov tímu. Členovia tímu by sa preto mali aktívne podieľať nielen na tvorbe plánu, ale aj na rozdeľovaní úloh. Ani ten najdokonalejší plán sa však v praxi nedá dodržať úplne na 100%. Preto je dôležité navrhnúť plán tak, aby sa v prípade potreby dal ľahko modifikovať a aktualizovať. Taktiež je veľmi dôležité sledovať plnenie plánu a v závislosti na okolnostiach ho aktualizovať. Striktné trvanie na dodržiavaní plánu môže v tíme viesť k nepochopeniu a takto znemožniť ďalší postup v projekte.

Úvod

Tvorba plánov v softvérovom projekte je jednou z najťažších aktivít v modernom vývoji softvéru. Bez realistického a objektívneho plánu projektu nemôže byť proces vývoja softvéru zvládnutý efektívne. Prekročenia plánu aj o 100 - 200% sú bežnou záležitosťou. Niektoré softvérové projekty nič nepriniesli, lebo manažéri mali problém s pochopením procesu tvorby softvéru, ktorý je definovaný v pláne projektu [1].

Veľa ľudí si myslí, že plánovanie je strata času, že keby sa čas potrebný na tvorbu plánu vynaložil na „skutočnú“ prácu na projekte, bol by projekt hotový oveľa skôr. Podľa môjho názoru však toto môže byť pravda len pre veľmi malé projekty, vykonávané v tíme možno tak 2-3 ľudí. Pre každý väčší projekt je podľa mňa potrebné vypracovať dôkladný plán.

Projekt je väčšinou príliš veľký na to, aby sa dal naplánovať celý naraz. Preto plánovanie väčšinou postupuje po malých krokoch. Niektoré veľké projekty sa dokonca plánujú tak, že sa rozdelia na čo najviac nezávislé časti, ktoré sa potom plánujú samostatne [4]. Výsledný projektový plán obsahuje rozvrh činností a termínov, ktoré majú viesť k požadovanému cieľu. Plán je však iba odhadom trvania jednotlivých činností a ich postupnosti. Preto je potrebné, aby bol plán flexibilný a aby sa dal kedykoľvek aktualizovať.

Plánovanie je veľmi úzko späté s riešeným problémom, ako aj s povahou členov tímu, podieľajúcich sa na projekte. Preto ani v odbornej literatúre nenájdeme presný návod na to, ako zostrojiť plán. Po prečítaní niekoľkých článkov o tvorbe plánu sa v nasledujúcom texte pokúsím priblížiť moju predstavu o tom, čo by mal plán obsahovať a taktiež o úlohách členov tímu.

Práca na projekte by sa mala rozdeliť na etapy a každá z týchto etáp by mala mať „hmatateľný“ výstup. Po úspešnom dokončení všetkých etáp projektu by samozrejme aj celý projekt mal byť úspešne ukončený. Jednotlivé etapy projektu by mali mať jednoznačné chronologické poradie, teda malo by byť vždy jasné, ktorá etapa projektu nasleduje a ktoré etapy projektu môžu byť vykonávané zároveň. Každá etapa by mala mať určený svoj výstup, ako aj dobu trvania. Etapy sú ešte ďalej rozdelené na moduly. Každý člen tímu pracuje na svojom module, kým manažér riadi vývoj celej skupiny modulov ku konečnému cieľu. Celý tento postup rozdeľovania na etapy a moduly vychádza z predpokladu, že riadiť malý projekt je oveľa jednoduchšie ako riadiť veľký projekt [4].

Pre úspech každého projektu je veľmi dôležitý aj tím ľudí, ktorý na ňom pracuje. Je potrebné zvoliť nielen expertov na všetky aspekty projektu, ale aj zabezpečiť, aby členovia tímu dokázali spolu efektívne pracovať. Podľa môjho názoru je veľmi dôležité, aby ľudia v tíme pracovali spolu na dosiahnutí spoločného cieľa, nie každý sám za seba.

Prístupy k tvorbe plánu

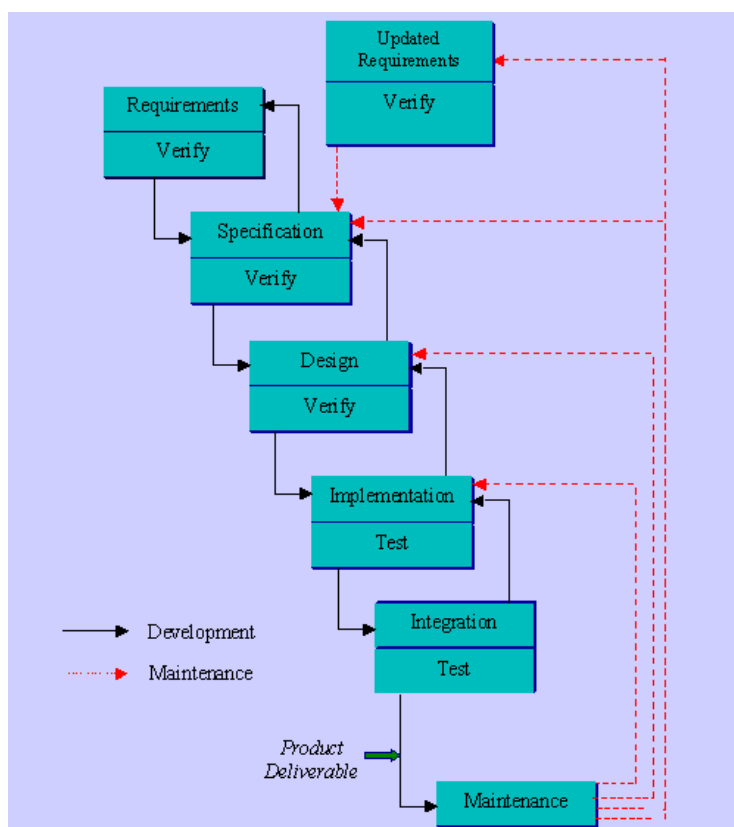
Pri tvorbe plánu treba brať do úvahy, ako vyzerá životný cyklus vytváraného softvérového produktu. S ním totiž priamo súvisí rozvrh činností a čas potrebný na ich ukončenie. V tejto eseji sa pokúsím priblížiť hlavne vodopádový model plánovania a taktiež spomeniem „Architecture-Centered Software Project Plannig“.

Životný cyklus softvéru je často reprezentovaný ako jednoduchý *vodopádový model*, kde vývoj softvéru napreduje cez 7 jasne definovaných etáp [5]. Tento model je zobrazený na obrázku č. 1. Z tohto modelu jasne vyplýva postupnosť jednotlivých etáp, ostáva teda už len určiť dobu ich trvania.

Jednotlivé etapy vodopádového modelu:

1. Vo fáze „*analýza požiadaviek*“ je špecifikovaný problém a sú identifikované obmedzenia
2. Vo fáze „*špecifikácia*“ je vytvorená detailná špecifikácia systému. Tento dokument by mal jasne definovať funkciu produktu.

3. Vo fáze „návrhu systému“ je systémová špecifikácia preložená do softvérovej reprezentácie.
4. Vo fáze „implementácia a testovanie“ dochádza ku programovej realizácii softvéru a jeho testovaniu.
5. Vo fáze „integrácia a testovanie systému“ sú pospájané všetky podsystémy do celistvého systému a testuje sa, či systém spĺňa všetky požiadavky. Po tejto fáze je softvér dodaný zákazníkovi na „akceptačné testovanie“.
6. Fáza „údržby“ je väčšinou najdlhšou fázou v životnom cykle softvéru. V tejto fáze je softvér aktualizovaný aby vyhovoval potrebám zákazníka a aby sa prispôboval zmenám prostredia. Taktiež sú opravované rôzne chyby nezachytené počas testovania a zvyšuje sa efektívnosť softvéru [6].

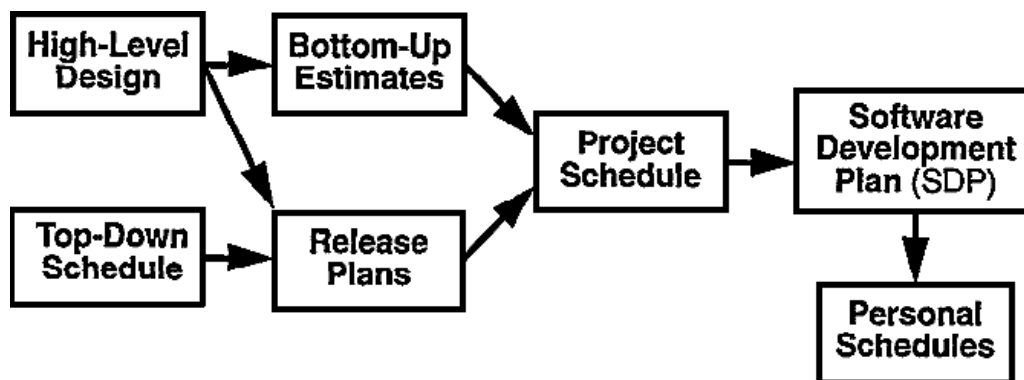


Obr. 1. Vodopádový model životného cyklu softvérového projektu [6]

Cieľom tejto eseje nie je do podrobností vysvetliť štruktúru vodopádového modelu životného cyklu softvéru. Chcel by som ale ukázať, že pri tvorbe plánu softvérového projektu je veľmi dôležité, aký model životného cyklu zvolíme. Podľa

môjho názoru nie je tento model veľmi dobrým riešením pre návrh systému, lebo pri jeho použití je aktualizácia a zmena v pláne veľmi problematická. Taktiež chyby, ktoré sa vyskytnú v niektorých vyšších fázach modelu sú prenesené do všetkých nižších fáz. V najhoršom prípade sú tieto chyby odhalené až pri akceptačnom testovaní.

Zaujímavým prístupom k tvorbe plánu je *plánovanie softvéru na základe architektúry* (Architecture-Centered Software Project Plannig - ACSPP). ACSPP je aplikovaný v skorom štádiu vývoja softvéru po tom, čo je dokončená analýza požiadaviek. Manažment veľmi často požaduje veľmi skoré odhady času a prostriedkov potrebných na vytvorenie nového softvérového produktu. V niektorých prípadoch sú tieto odhady použité na zistenie, či sa oplatí projekt vziať, alebo nie. Úlohou ACSPP je poskytnúť tieto odhady v čo najkratšej dobe a čo najpresnejšie [2]. Prístup ACSPP je znázornený na obrázku č. 2.



Obr. 2. Architecture-Centered Software Project Plannig [3]

Malý tím začne tvoriť návrh systému na vysokej úrovni abstrakcie (High-Level Design). Paralelne s ním druhý tím vytvára rozvrh práce (Top-Down Schedule). Pri jeho návrhu sa využívajú znalosti z tvorby iných (podobných) systémov, napríklad z koľkých riadkov kódu sa skladajú konkurenčné systémy a pod. High-Level Design a Top-Down Schedule sú podkladom pre plán uvoľňovania verzií softvéru (Release Plans). High-Level Design je tiež podkladom pre Bottom-Up Estimates, kde členovia tímu odhadnú čas a úsilie potrebné na vytvorenie im pridelených softvérových komponentov. Tieto odhady sa potom porovnávajú s Release Plans a tak vzniká rozvrh projektu (Project Schedule). V tomto rozvrhu je už zohľadnený aj životný cyklus každého komponentu systému. Teda napríklad tvorba zložitej časti systému začne skôr (aj keď táto časť nie je potrebná na začiatku projektu), aby bolo dost' času napríklad na jej testovanie. Z rozvrhu projektu potom vzniká plán vývoja softvéru (Software Development Plan), od ktorého sa následne odvíjajú rozvrhy jednotlivých členov tímu (Personal Schedules) [2].

Na tomto prístupe sa mi páči, že na tvorbe plánu sa podieľajú všetci členovia tímu, nie len manažment. Preto si myslím, že takto naplánovaný projekt má veľkú šancu na dokončenie včas a s minimom závažných chýb.

Na záver tejto časti by som rád uviedol, že okrem uvedených spôsobov plánovania existuje množstvo ďalších metód na tvorbu plánov v softvérovom projekte. Pri výbere vhodného plánu treba zohľadniť hlavne veľkosť tímu, množstvo času a finančné zdroje. Mňa z uvedených spôsobov plánovania viac oslovil ACSPP, lebo podľa môjho názoru poskytuje väčšiu flexibilitu ako plánovanie pomocou vodopádového modelu.

Rozdelenie úloh v tíme

Tím je skupina ľudí pracujúcich spolu na dosiahnutí spoločného cieľa, za ktorý sú všetci zodpovední. Členovia tímu zdieľajú znalosti a skúsenosti pri plnení tímovej práce. Tímovú prácu pokladajú za prioritu, nie za niečo, čo ich ruší pri ich skutočnej práci.

Aby bol tím úspešný, musí byť vykonané veľké množstvo úloh. Tieto úlohy sú rozdelené medzi *členov tímu*, *vedúceho tímu*, *inštruktorov* a *sponzorov*.

Členovia tímu sú ľudia vybraní sponzorom a vedúcim tímu, ktorí zdieľajú svoje znalosti a skúsenosti počas práce na tímovom projekte.

Vedúci tímu riadi aktivity tímu, udržiava dokumentáciu a slúži ako spojenie so zvyškom organizácie.

Inštruktori sú ľudia, ktorí učia používať vedúceho tímu a členov tímu rôzne podporné nástroje a postupy a snažia sa tak uľahčovať tímovú prácu.

Sponzori sú manažéri, ktorých úlohou je posudzovať prácu tímu. Teda sa snažia identifikovať, čo sa dá zlepšiť[3].

Úlohy v tíme by sa dali rozdeliť na dve skupiny a to *vývojové úlohy* a *riadiace úlohy*. Členovia tímu vykonávajú vývojové úlohy. Riadiace úlohy sú vykonávané vedúcimi tímu, inštruktormi a sponzormi. Jednotlivé úlohy v tíme sú veľmi špecifické a závisia z veľkej časti od konkrétneho projektu. Je však veľmi dôležité, aby rozdelenie úloh vyplynulo prirodzene zo situácie tímu. O pridelovaní úloh by nemali rozhodovať len manažéri, ale mali by sa na ňom podieľať všetci členovia tímu.

Aktualizácia a zmena plánov

Kontrola plánov by mala byť pre členov tímu samozrejmosťou. Každý člen má svoj osobný plán pre časť projektu, na ktorej práve pracuje. Preto by mal sledovať a kontrolovať, či napreduje podľa plánu. Ak si je vedomý toho, že plán nestíha včas splniť, mal by to ísť oznámiť manažérovi, ktorý potom môže urobiť zmeny v rozvrhu projektu a navrhnúť riešenie, napríklad prideliť na danú činnosť viac ľudí.

Myslím si, že plány by mali byť predmetom verejnej diskusie. Je veľmi dôležité, aby ľudia s plánom súhlasili a nepovažovali ho od začiatku za nespĺniteľný. Takéto zlé plány môžu výrazne naštvať morálku v tíme. Taktiež súťaživosť v tíme nie je síce v zásade zlá vec, ale ak sa z tvorby projektu stane súťaž, kto splní viac úloh v čo najkratšom čase, tiež sa to môže neblaho prejaviť nie len na kvalite výsledného produktu, ale aj na spokojnosti členov tímu.

Dohľad nad plánom zabezpečujú manažéri. V prípade, že plán treba aktualizovať, je vhodné poradiť sa o úpravách plánu s členmi tímu. Ideálne podľa mňa je, ak sa aktualizácie plánu prerokúvajú na spoločných stretnutiach tímu, kde každý môže prejaviť svoj názor a prispieť tak k vylepšeniu plánu.

Veľmi dôležitá je aj motivácia zákazníkov. Ak to vyžadujú okolnosti, tak je podľa môjho názoru vhodné, aby manažéri motivovali tím či už finančnými odmenami, dovolenkami, alebo inými formami benefitov.

Záver

V eseji *Tvorba plánov v softvérovom projekte, rozdelenie úloh, plnenie a aktualizácia plánov projektu* som sa zamerlal na tvorbu plánov v softvérovom projekte. Taktiež som v tejto eseji vyjadril svoj názor na rozdelenie pracovných úloh v tíme a na aktualizáciu plánu projektu. V eseji sú vysvetlené dva princípy plánovania, a to tradičný *vodopádový model* plánovania a *plánovanie softvéru na základe architektúry*. Podľa môjho názoru je vodopádový model v dnešnej dobe už prekonaný, hlavne z dôvodu, že je málo flexibilný. Nechcem tým povedať, že by bol zlý, ale na dnešné dynamické projektu sa podľa mňa nehodí. Ďalej sa v eseji dá nájsť kapitola o rozdelení úloh v tíme, ako aj o aktualizácii a zmene plánov projektu. Treba si uvedomiť, že aj keď sa plán zdá na začiatku projektu akokoľvek dobrý, tak sa v praxi nevyhne väčším i menším úpravám. Robiť úpravy v pláne nie je žiadna katastrofa, len treba zabezpečiť, aby bola každá úprava plánu prínosom.

Použitá literatúra

1. Simmons Ching-Seh Wu, D.B. Dept. of Comput. Sci., Texas A&M Univ., College Station, TX; In: Computer Software and Applications Conference, 2000. COMPSAC 2000. The 24th Annual International; Publication Date: 2000; Software Project Planning Associate (SPPA): a knowledge-based approach for dynamic software project planning and tracking
2. Paulish Daniel J., Nord Robert L., Soni Dilip; Siemens Corporate Research, Inc.; Experience with Architecture-Centered Software Project Planning
3. Scholtes Peter R., Joiner Brian L., Streibel Barbara J.; The Team Handbook
4. Rettig, M., Simons, G.: A Project Planning and Development Process for Small Teams. Communications of the ACM, Vol. 36, No. 10 (1993) 45-55.
5. Scacchi Walt, Institute for Software Research, University of California, Irvine, February 2001; Process Models in Software Engineering
6. http://scitec.uwichill.edu.bb/cmp/online/cs221/waterfall_model.htm (dátum prístupu: 15.10.2008)

Annotation*Software Project Planning, Assigning Tasks, Fulfilling and Updating the Project Plan*

This paper describes my opinion about software project planning, assigning tasks and fulfilling and updating the project plan. There are several ways how to plan a software project. My aim is to describe a traditional waterfall model of software project planning. Directing a software project, it is not important just to build a perfect plan, but also to assign a correct task to each team member. Correct task assigning is also important for keeping plan schedule and to maintain a good work atmosphere. That is why it is very important to take note of skills, experiences and preferences of team members. Team members should participate on project planning and also on tasks assigning. Even the greatest plan cannot be 100% fulfilled. That is why it is important to design a plan in a way that it can be easily actualized. Monitoring progress in a plan and changing the plan if needed is also very important. Strict persisting on keeping the plan schedule can lead to incomprehension and thus making the progress in project impossible.