

Dekompozícia projektu ako cesta k vytváraníu plánov

MICHAL JAJCAJ

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
michal.jajcaj@gmail.com*

Abstrakt. Esej sa zaoberá plánovaním v softvérových projektoch. Tak ako aj v iných projektoch, tak aj v oblasti softvérových projektov vznikajú problémy rôzneho druhu. Práca na ich odstránení by v optimálnom prípade mala začať už pred samotným samotným zahájením projektu vytvorením jednej z najzákladnejších častí – plánu projektu. Ten dopĺňajú procesy, ktoré definujú postupy a metódy ako plán dosiahnuť. Práca rozoberá rozloženie projektu na menšie časti, ktoré sa dajú jednoduchšie a prehľadnejšie sledovať. Vytvorený plán umožňuje sledovať postup pri riešení cieľov projektu a zároveň napomáha pri rozdeľovaní zdrojov potrebných na dosiahnutie jeho cieľov. Tieto zdroje môžu byť na jednej strane finančné alebo materiálne a na druhej strane ľudské, medzi ktoré je potrebné naplánované úlohy rozdeliť.

Úvod

Softvérový projekt je v mnohých pohľadoch podobný iným projektom, napríklad zo stavebníctva, alebo z iných oblastí. V mnohom je však aj odlišný. Tá hlavná vec, ktorá ich tak rozlišuje, je to, že softvérový projekt sa dá pred jeho samotným vytvorením vizualizovať alebo predstaviť iba v obmedzenej podobe. A tento neúplný a nedokonalý pohľad na budúci softvérový projekt má slúžiť tak pre zákazníka, ako aj pre tím vývojárov, ktorí sa budú podieľať na jeho riešení. Podobne, ako pri už spomínanom stavebníctve, ani pri softvérových projektoch sa nezaobídeme bez rôznych zdrojov, či už finančných, materiálnych a ľudských. Dôležitú úlohu popri tomto všetkom hraje čas. To, ako všetky tieto ingrediencie efektívne využiť v obmedzenom čase a tak dospieť k cieľu, je plán.

Pohľad na softvérový projekt

Na softvérový projekt sa dá hľadieť z rôznych úrovní. Z toho najvyššieho pohľadu pravdepodobne uvidíme celý projekt ako menší, či väčší celok, ktorý je potrebné vyriešiť zatiaľ neznámymi cestami a spôsobmi. To akým veľkým ho vnímame môže závisieť od rôznych okolností, ako sú skúsenosti s riešením podobných projektov, znalosť danej problémovej oblasti atď. V každom prípade je tento prvý pohľad na problém veľmi zjednodušený.

Keď sa však začneme o problém viac zaujímať, môžeme zbadáť, že náš projekt sa začína postupne rozpadávať na väčší počet menších častí. Toto približovanie sa je prítomné pri plánovaní asi všetkých typov projektov, v tom našom, softvérovom, je však nevyhnutné. Ak by sme zostali vnímať projekt ako jeden celok, bolo by prakticky nemožné ho naplánovať, odhadnúť jeho trvanie, cenu (všeobecne vzaté aj všetky iné náklady) a zabezpečiť jeho kvalitu. Tieto tri posledne menované atribúty projektu sú však kľúčové pre obe strany – pre zákazníka aj dodávateľa. Ďalším dôvodom pre toto rozkladanie pôvodného problému na menšie časti je myšlienka, že je jednoduchšie naplánovať to, čo bude robiť jeden človek nasledujúci týždeň alebo dva, ako naplánovať to, čo bude robiť 5-6 ľudí, alebo dokonca celý tím nasledujúcich pár mesiacov. O tomto zjemňovaní pôvodného problému nehovoríme náhodne, ide o jeden zo základných kameňov, na ktorom je postavené plánovanie softvérových projektov.

Dekompozícia ako cesta k plánu

Projektová úroveň

V dokumente *A Project planning and development process for small teams* [3] sa hovorí o troch úrovniach plánovania softvérového projektu. Prvá úroveň je projektová úroveň a jej cieľom je určenie základných cieľov plánovaného projektu. Pri tomto sa na projekt pozeráme z vysokej úrovne. Na tejto úrovni sa stanovujú globálne ciele projektu, to, čo má byť výstupom projektu. Tieto ciele môžu byť už určené v špecifikácii projektu ak sa jedná o vývoj takého softvéru, o ktorom máme dopredu dostatočné informácie. Výstupom tohto procesu je stručný náčrt projektu. Ten by mal obsahovať nasledujúce časti:

Názov projektu.

Cieľ projektu. Toto je tá časť, ktorá obsahuje stručné informácie o ciele projektu. Dôležité je, aby ciele boli vyjadrené merateľnými atribútmi. Pre úplnosť informácií môže obsahovať odkazy na špecifikáciu projektu.

Štádiá projektu. Usporiadany zoznam štádií, ktorými bude projekt prechádzať na ceste k jeho vyriešeniu. Jednotlivé štádiá by mali obsahovať termíny ich dokončenia.

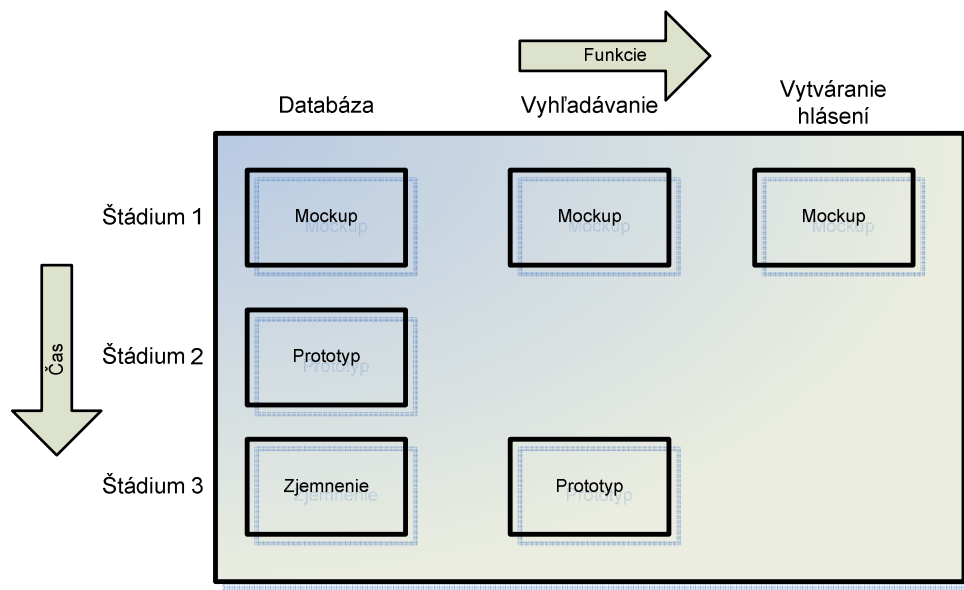
Členovia tímu. Zoznam ľudí pracujúcich na danom projekte. Zároveň sa uvedie, aké majú úlohy jednotliví členovia tímu. V podmienkach podobných tým našim na

tímovom projekte, kde sú jednotlivé tímy relatívne malé (4-6 ľudí), vznikne len pár rolí, ktoré si členovia rozdelia. Podľa [3] je možné úlohy rozdeliť na technické a sociálne. Medzi tie technické patria úlohy implementátorov, doménových špecialistov a úloha technického revízora. Technické úlohy tvoria hlavnú činnosť na samotnej práci na projekte. Implementátori vytvárajú samotný program písaním zdrojových kódov. Doménový špecialista sú odborníkmi na danú oblasť, nie je nutné, aby ovládali programovanie. Zúčastňujú sa na riešení svojimi vedomosťami a taktiež kontrolujú výstupy práce iných osôb. Technickí revízori môžu byť samotní implementátori. Navzájom revidujú výstupy iných implementátorov. Účelom tejto úlohy je v prvom rade zhodnotiť prácu inej osoby, poprípade navrhnúť vylepšenia, čím sa prispieva ku kvalite celkovej práce. Dôvodom, ktorý sa mi však zdá tiež dôležitý je to, aby sa aj ostatní členovia tímu oboznámili s aktuálne riešeným problémom a tak získavali nové vedomosti.

Okrem týchto technických úloh vznikajú pri riešení projektu aj takzvané sociálne úlohy. Dôležitou rolou je úloha vedúceho tímu. Táto osoba je priamo zúčastnená na riešení projektu. Jej výber závisí od rôznych okolností, v niektorých tímoch môže byť okamžite známe, kto je touto osobou, v inom tíme sa táto osoba postupne vyprofiluje. Vedúci tímu má viaceré dôležité úlohy. Myslím však, že nie je potrebné aby všetky tieto úlohy vykonával sám. Skôr sa prikláňam k tomu, aby týmito úlohami delegoval iných členov tímu a iba dohliadal na ich vykonávanie a sám vykonával iba niektoré z nich. Úlohy, o ktorých hovoríme, sú: organizovanie a vedenie stretnutí tímu, vytváranie a archivovanie zápisov zo stretnutí a iných dokumentov, udržiavanie a aktualizovanie stavu riešenia projektu. Vykonávanie týchto môžu zabezpečovať členovia tímu vykonávajúci už aj iné technické úlohy, alebo osoby špeciálne určené len pre tieto úlohy. Úlohou, o ktorej si myslím, že by mal vykonávať sám vedúci tímu je samotné plánovanie projektu. K tomuto názoru ma privádza to, že vedúci tímu na to, aby sa vedúcim stal, musel preukázať isté schopnosti. Či už sú to vedomosti z problémovej oblasti, alebo schopnosti riadenia tímu, je táto osoba pravdepodobne najvhodnejšia na plánovanie projektu.

Prvým krokom k vytvoreniu plánu je už spomínané rozdelenie projektu na štádiá (vhodným pomenovaním pre túto menšiu časť je aj *fáza*, tá však bude neskôr použitá v inom kontexte). Toto rozdelenie sa deje smerom zhora dole, čiže rozkladáme celok na menšie časti. Každé štádium projektu je charakterizované tým, čo má byť jeho výsledok a akú úlohu bude tento zohrávať v celom systéme. Jednotlivé štádiá následne prechádzajú štyrmi všeobecnými fázami. Prvou fázou je rýchle vytvorenie vizuálnej prezentácie toho, čo by malo byť výstupom daného štádia. Ide o tzv. *mockup*. Tento krok je dôležitý hlavne z pohľadu komunikácie projektového tímu so zákazníkom, pretože až v tejto fáze zákazník môže vidieť ako približne bude produkt naozaj vyzeráť a je dôležité, aby sa k tejto otázke vyjadril v čo najskoršej fáze riešenia projektu. Po zhodnotení a prípadnom doplnení a upravení cieľov a požiadaviek nasleduje fáza vytvorenia prototypu. V tejto fáze sa už vytvára funkčná časť systému. Tá je znova zhodnotená používateľmi a prechádza do tretej fázy, nazvanej *zjemnenie*, kde sú už doladené detaily a aplikované poznatky a zistenia používateľov. Poslednou fázou je fáza údržby, ktorá slúži na ďalšie prispôbovanie a úpravy funkcionality, odstraňovanie zistených problémov atď. Tieto štyri fázy sa nedajú aplikovať na každé

štádium riešenia projektu, ich výber môže závisieť od rôznych faktorov, akými sú rozsiahlosť aktuálne riešeného štádia, jeho zložitosť, či vôbec možnosť nejakého vizuálneho výstupu daného štádia, ak sa napríklad jedná o implementáciu algoritmu. Na konci prvej úrovne plánovanie by malo byť známe časové a aj funkčné rozloženie častí riešeného projektu. Graficky sa dá znázorniť spôsobom zobrazeným na obrázku č. 1.



Obr. 1. Zobrazenie fáz riešenia projektu podľa času a funkcií.

Úroveň samostatných štádií

Po vytvorení hrubého plánu celého projektu sa prejde na úroveň plánovania jednotlivých štádií. Podobne ako pri projektovej úrovni, aj tu sa na štádia projektu pozerá ako na samostatné celky, ktoré sa ďalej dekomponujú na časti nazývané moduly. Názov modul nie je v tomto prípade označením podprogramov, ale jednotiek práce. Výhodou tohto prístupu je napríklad jednoduchšie sledovanie postupu prác. Termíny pre vypracovanie modulov sa na tejto úrovni pohybujú v niekoľkých dňoch až týždňoch, takže je aj jednoduchšie včas spozorovať vznik oneskorení a následne na to upraviť plán tak, aby tieto oneskorenia mali čo najmenší vplyv na celkový termín ukončenia aktuálneho štádia a tým aj celého projektu.

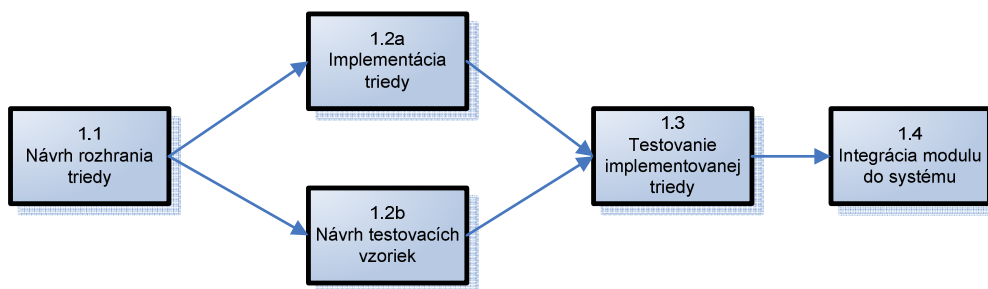
Myslím, že výhodou takéhoto rozkúskovania práce je aj to, že osoby pracujúce na týchto moduloch môžu častejšie odovzdávať časti vykonanej práce na zhodnotenie ostatným členom tímu, čo môže zvyšovať ich motiváciu v ďalej práci. Na druhej strane však treba dávať pozor na to, aby časté odovzdávanie malých častí a s ním spojená ich

zhodnotenie ostatnými členmi nezaberali príliš veľa času a nevytvárali tak nežiaduce oneskorenia.

Úroveň modulov

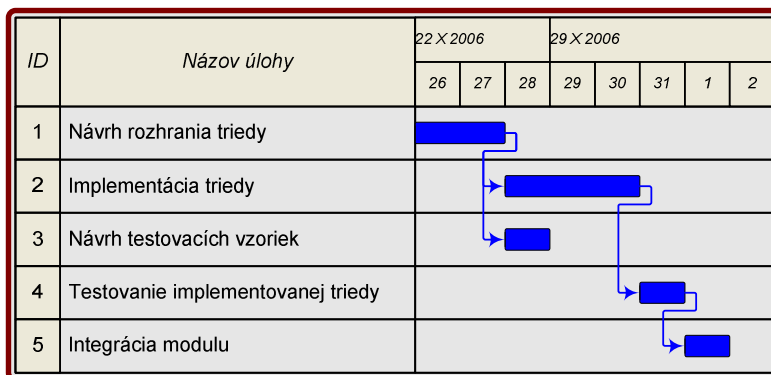
Po dekompozícii štádia na moduly sa vykoná odhad ich zložitosti. K tomuto odhadu je možné použiť buď informácie z predchádzajúcich projektov, ak také boli, alebo nechať priamo ohodnotiť modul členmi tímu jednoduchými atribútmi jednoduchý, stredne náročný alebo náročný. Inými metrikami môžu byť aj odhadovaný počet riadkov zdrojového kódu, počet tried, ktoré musia byť implementované a rôzne iné. Odhadovanie nákladov je vo všeobecnosti zložitý proces v ktorom vystupuje veľa rôznych premenných a existuje k nemu viacero prístupov. Ideálne z hľadiska presnosti odhadu by bolo plánovanie posunúť do čo najneskoršej fázy riešenia projektu. Tu však nastáva rozpor s tým, čo už bolo povedané, že plánovanie je začať už pred samotným riešením problému a tak je tento prístup nepoužiteľný.

Výsledkom dekompozície je zatiaľ neusporiadaná množina modulov. Tie však majú medzi sebou závislosti, či už z funkčného alebo časového hľadiska. Na znázornenie týchto závislostí je vhodné využiť niektorú z ich grafických reprezentácií. Z môjho pohľadu najjednoduchším spôsobom je použiť graf siete úloh. Uzmi tohto grafu sú jednotlivé moduly a hrany znázorňujú závislosti medzi nimi. Príklad takéhoto grafu sa nachádza na obrázku číslo 2.



Obr. 2. Príklad reprezentácie modulov v jednom štádiu riešenia projektu so znázornením závislostí medzi modulmi.

Ako je možné vidieť na obrázku 2, tento graf neobsahuje informácie o časoch a termínoch riešenia jednotlivých modulov. Plánovanie je však aj o konkrétnych termínoch, v ktorých má byť či už celý projekt, alebo jeho časť spracovaná. Preto je vhodné do grafu vložiť aj túto informáciu. Získame tak graf, kde na horizontálnej osi je čas a na osi vertikálnej sú jednotlivé moduly a štádia získané v procese dekompozície. Takýto graf sa nazýva aj Ganttov graf. Pomocou Ganttovho grafu je možné vyjadriť časové závislosti, ako aj závislosti medzi jednotlivými modulmi či štádiami projektu. Tento graf je aj veľmi výhodným pri sledovaní stavu plnenia úloh, pretože je z neho jednoduché vidieť, aké následky bude mať nesplnenie úlohy v danom čase, ktoré ďalšie moduly budú ovplyvnené a bude nutné ich preplánovanie. Príklad Ganttovho grafu je na obrázku číslo 3.



Obr. 3. Ganttov graf. Jednoduchý príklad znázornenia postupnosti riešenia modulov s časovou informáciou.

Postupným rozkladaním pôvodného problému sme sa cez dve úrovne dostali do tretej, v ktorej už vystupujú iba relatívne malé a jednoduché časti, ktoré sme nazvali modulmi. Pred preštudovaním materiálov o plánovaní by som pravdepodobne v tejto fáze skončil, zdalo sa mi, že úlohy sú dostatočne zjemnené, malé a definované na ich úspešné vykonanie. Možno by to v istých prípadoch bolo takto dostačujúce, závisí to od veľkosti projektu, jeho zložitosti, rozsahu. Tým som chcel naznačiť, že aj na tejto tretej, poslednej, úrovni existuje akýsi mikro životný cyklus každého modulu, ktorý môže byť tiež naplánovaný. Plánovanie každého modulu je už v rukách samotného vývojára, ktorý je za modul zodpovedný. V zdrojoch [1] a [3] je tento životný cyklus modulu označovaný anglickým akronymom PADRE (Plan – Approve – Do – Review – Evaluate), čo by sa dalo voľne preložiť ako Plánuj, Schvaľuj, Vykonaj, Posúď, Vyhodnoť. Tieto fázy sa dajú chápať ako míľniky, ktorých úspešné dosiahnutie vedie k vytvoreniu modulu.

Plánovanie je fáza v ktorej člen tímu vytvorí plán na implementáciu a testovanie modulu a odovzdá tento plán inému členovi tímu na schválenie. Táto osoba má za úlohu prehliadnuť vytvorený plán a v prípade potreby ho upraviť tak, aby s obsahom plánu súhlasil celý tím. Následne sa plán dostane k osobe, ktorá ho implementuje. V závislosti od toho, čo je obsahom plánu, môže byť výstupom tejto fázy vytvorenie časti programu, vytvorenie dokumentácie alebo iná aktivita. Po implementovaní nasleduje znova fáza kontroly výstupu tohto procesu a jeho možné úpravy a doladenie. Je možné, že tieto dve fázy – implementačná a následná kontrolná, sa budú viac krát opakovať až kým nebude tím spokojný s výsledkom. Na záver je celý proces zhodnotený medzi členmi tímu a vedúcim tímu, pričom sa hľadajú možnosti, ako tento proces v budúcnosti spraviť efektívnejším. Aby celý tento proces plánovania nebol iba jednorazovou záležitosťou a zanechal po sebe cenné informácie do ďalších štádií projektu či iných projektov, je dobré zaznamenávať informácie o postupe pri práci na

moduloch. Jednoduchým nástrojom na toto môže byť tabuľka, kde sa zaznamenávajú termíny pôvodného plánu, termíny aktualizovaného plánu a konečné termíny, kedy boli jednotlivé míľniky v životnom cykle modulu dosiahnuté. Na konci každého štádia je na základe týchto informácií možné analyzovať, vývoj ktorých modulov sa nepohyboval v plánovaných termínoch. Môžu nastať situácie, kedy bolo na vývoj modulu prideleného málo prostriedkov a času a termín bol tak prekročený, ale aj opačný prípad, kedy bol čas potrebný na vývoj modulu nadhodnotený. V oboch prípadoch ide o nežiaduci jav, ktorý je však pomocou záznamov z minulosti možné do istej miery eliminovať.

Záver

Plánovanie softvérových projektov by sa dalo zhrnúť ako dekomponovanie pôvodného problému postupne na menšie a menšie časti. V tejto fáze postupujeme smerom zhora nadol. Vstupom môže byť špecifikácia softvérového projektu, výstupom je hierarchia štádií a navzájom poprepájaných modulov. Tým, že sme sa dostali na takúto nižšiu úroveň, je jednoduchšie odhadnúť potrebné náklady a čas. Následným zozbieraním týchto údajov smerom zdola nahor zistíme odhadované celkové náklady a čas potrebný na projekt. Ak zistíme, že údaje, ktoré sme získali sa výrazne líšia od tých, ktoré má projekt pridelené, je potrebné plán revidovať.

Použitá literatúra

1. Devedzic, V., Software Project Management, School of Business Administration, University of Belgrade, Yugoslavia, 1991.
2. Pressman, R.S., Software Engineering: A Practitioner's Approach, 5th edition, 2001.
3. Rettig, M., Simons, G., A Project Planning and Development Process for Small Teams. *Communications of The ACM* 36, October 1993, pp. 45-55.

Annotation

Decomposition as a way to project planning

Essay deals about planning in software projects. As in other projects, so in software projects the problems of many kinds arise in time. In ideal case, the effort for their elimination should begin before the whole project starts by creating one of the basic parts – the project plan. The plan is supported by processes which define methods how to fulfill the plan. This work deals about decomposing the whole project into smaller parts which are easier to manage and track their progress. Created plan allows tracking how the work is going and helps to divide resources needed for achieving its goals. These resources can be on one hand financial and material, on the other human sources. Both of these have to be assigned into planned tasks.