

Prostriedky pre riadenie projektu, alebo význam nástrojov na sledovanie úloh a problémy s nimi späté

PETER NOSKO

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

Abstrakt. Ľudia, ktorí sú súčasťou softvérových projektov väčšinou využívajú nejakú formu sledovania a manažmentu úloh, či už pomocou papierových lístkov, Outlooku, alebo softvéru na to určeného. Použitie softvérových nástrojov na tento účel môže značne zjednodušiť prácu manažmentu, avšak niekedy prináša aj nové riziká. Táto esej bude pojednávať o dôležitosti takýchto nástrojov, ako aj požiadavkách na ne. Okrem toho esej rozoberá problémy súvisiace s použitím podporných prostriedkov pre riadenie projektu a uvažuje aj o možnostiach prevencie týmto problémom za použitia agilných metód vývoja.

Úvod

Nesmierne dôležitým aspektom každého, nielen softvérového projektu, je jeho riadenie. Hlavnými súčasťami tejto aktivity sú sledovanie projektu a jeho plánovanie. Plánovanie projektu obnáša detailné naplánovanie konštrukcie softvéru, plán zaručenia kvality, ako aj plán samotného vývoja [4]. Pod sledovaním projektu rozumieme zaznamenávanie priebehu projektu, ktorého súčasťou je napríklad pridelovanie úloh pracovníkom, ako aj kontrola správneho vykonania daných úloh v zaznamenanom časovom rámci. Pri softvérovom produkte je toto obzvlášť dôležité, najmä preto, lebo výsledok činnosti softvérových vývojárov nie je hmatateľný a ani ho nie je jednoduché znázorniť. Ďalším problémom je aj to, že je potrebné manažovať nielen jednotlivcov, ale tím ako celok, a to aj s ohľadom na rozdiely v rámci tímu a prípadne aj požiadavky iných tímov. Pre to, aby bolo možné dosahovať žiaduce organizačné výsledky, ako zvýšenie celkovej firemnej výkonnosti, je potrebné vnieť do manažmentu ľudí a úloh systém a známe postupy a metódy riadenia ľudských zdrojov a úloh.

Je možné, že pri projekte na ktorom sa podieľa iba niekoľko málo ľudí sa dá riadiť ich aj prostredníctvom papierových odkazov, ale pri väčších projektoch býva

zanedbanie potreby riadenia úloh pomocou sofistikovanejších prostriedkov jednou z príčin neúspechu projektu.

Riadenie

Pri riadení softvérového projektu je cieľom naplnenie požiadaviek s dostupnými zdrojmi a prostriedkami za dohodnutý čas, teda v danej cene a s požadovanou kvalitou. Toto vyžaduje systém monitorovania a koordinácie etáp, z ktorých samotný projekt pozostáva. V rámci etapy je potrebné zaznamenávať zmeny, ktoré sa počas nej vyskytnú a analyzovať ich kvôli neskoršiemu použitiu pri plánovaní a zasahovaní do priebehu projektu. Tento proces je tvorený niekoľkými ďalšími procesmi, ktoré delíme na základné a podporné. Základné procesy majú na starosti hlásenie postupu projektu a totálny manažment zmien. Úlohou podporných procesov je riadenie zmien spadajúcich pod konkrétne činnosti manažmentu.

Nástroje

Problémy a chyby

Z mojej skúsenosti vyplýva, že obvyklé problémy a chyby pri použití nástrojov sú tieto:

- Používanie nástroja na iné účely ako tie, na ktoré bol navrhnutý
- Používanie nesprávneho nástroja na danú činnosť
- Používanie veľkého počtu rôznych nástrojov, ktorých integrácia je obtiažna

Hoci prvé dva problémy sa na prvý pohľad môžu javiť veľmi podobné, ako uvidíme na príkladoch, sú medzi nimi rozdiely. Hneď prvý z uvedených problémov je pomerne častý a vyskytuje sa hlavne v menších projektoch s nižším rozpočtom, kde je snaha maximálne využiť nástroje ktoré sú k dispozícii. Nie je to však výhradne problém malých projektov, nájdeme ho aj v tých veľkých. Sám môžem uviesť príklad, kedy som bol svedkom toho, že vo veľkom medzinárodnom projekte sa na evidenciu niektorých úloh používal Excel hárok s položkami ako názov úlohy, popis, zodpovedný človek, odhadovaný časový rámec, splnené a zostávajúce úsilie. Takýto postup môže byť vhodný, ak sa pomocou neho manažuje iba malý počet ľudí a je pomerne malá pravdepodobnosť, že niektoré zmeny spôsobia nekonzistenciu dokumentu. Vo všeobecnosti je však viacero dôvodov, prečo je takýto postup nevhodný, medzi inými fakt, že MS Excel neumožňuje automatickú notifikáciu o zmene dokumentu, zaznamenávanie identity osoby, ktorá zmenu vykonala, ale predovšetkým to, že vo všeobecnosti je potrebné údaje v týchto dokumentoch ďalej analyzovať, archivovať, a pod. v inom systéme, čo znamená duplicitnú prácu.

Častým príkladom druhej z vyššie spomenutých chýb, vyplývajúcej z toho, že mnohokrát je diskusia o úlohe realizovaná prostredníctvom mailovej komunikácie, je

ukladanie elektronickej pošty napríklad do osobitného priečinka ako náhrada za záznamník úloh. Takáto praktika môže obzvlášť komplikovať a zneprehľadňovať osobný, ale aj tímový manažment. Do tejto kategórie problémov by sa dal zahrnúť aj problém ktorý som zažil s použitím príliš komplikovaného nástroja. V už spomenutom projekte bol istú dobu používaný nástroj na riadenie projektov Primavera. Ide o pokročilé, široko prispôsobiteľné riešenie, na ktorého efektívne použitie v praxi je potrebný vyškolený personál. Toto bolo do istej miery zanedbané a onedlho sa firma vrátila k alternatívnym riešeniam manažmentu projektu.

Tretí z problémov môžeme nájsť prevažne u projektov väčšieho rozsahu. Vyplýva z nejednotnej politiky ohľadom takýchto nástrojov v rámci firmy, absencii dohody v rámci celého manažmentu o používaní rovnakého nástroja, alebo prílišnej komplexnosti riadených procesov v organizácii, čo núti k použitiu viacerých - zväčša navzájom nekompatibilných - nástrojov. Napríklad v organizácii v ktorej som pôsobil boli požiadavky na zmenu kódu realizované prostredníctvom nástroja Rational ClearQuest, ktorý však vnášal do procesu mnoho byrokracie a v dôsledku týchto administratívnych prekážok stratu efektívnosti a flexibility pri riešení prioritných úloh. Manažéri si toto uvedomili a prioritné, prípadne rýchlo riešiteľné úlohy boli riadené pomocou „odľahčeného“ nástroja FixIt. Ďalej požiadavky na novú funkcionality obstarával už spomínaný Excel hárok, takže získať dobrý prehľad o stave úloh musela byť pre manažment jednoduchá úloha.

Taktiež, napríklad pri veľkých projektoch na ktorých sa podieľa viacero dcérskych divízií jedného koncernu, prípadne partnerských organizácií, vzniká množstvo ďalších problémov spôsobených nesúladom a/alebo absenciou vzájomnej kompatibility v používaných nástrojoch na manažment úloh a ľudských zdrojov.

Požiadavky

Aj na základe problémov v predošlej kapitole vieme zostaviť základné požiadavky na softvérový systém na sledovanie úloh.

- Vedúcemu tímu umožniť pohodlne prideliť úlohy, alebo notifikovať zverený tím o nových úlohách, ktoré je potrebné rozdeliť v rámci tímu
- Notifikovať vývojára o pridelenej úlohe, alebo mu jednoduchým spôsobom umožniť prevzatie nesplnenej a odovzdanie dokončenej úlohy.
- Celkovo prispieť k zjednodušeniu a zefektívneniu jednak medziľudskej, tak aj dátovej komunikácie
- Manažérovi uľahčiť sledovanie pridelených úloh a postup ich riešenia, získať prehľad o tom, do akej miery je naplnený plán, prípadne ako treba plán upraviť na základe rozdielu medzi plánom a skutočnou mierou jeho splnenia
- Pri vyskytnutí problému (sklzu oproti plánu, nedostatku zdrojov, nestabilite v tíme) umožniť skúsenému manažmentu čo najskôr problém spozorovať a pružne naň reagovať

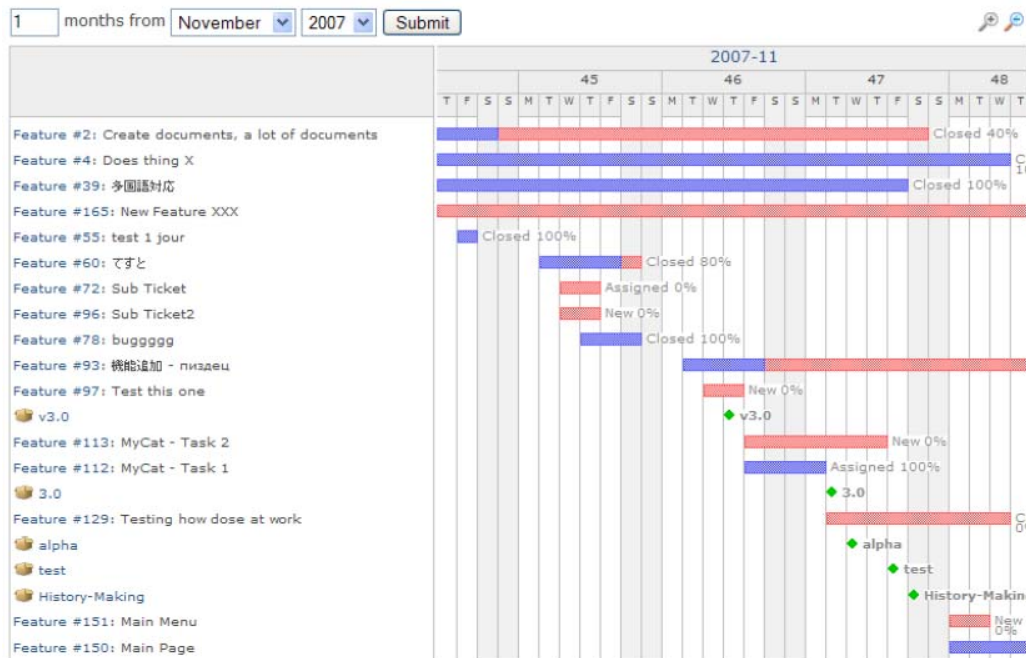
Reálna situácia a menej zřejmé výhody

Na trhu existuje veľa podporných softvérových prostriedkov, väčšina z ktorých viac-menej nejakým spôsobom spĺňa hore uvedené požiadavky. Bežne používané prostriedky majú spravidla klient-server architektúru, aby vedeli zabezpečiť funkcionality ako zdieľanie kalendára pre komunikáciu ohľadom tímových stretnutí a posledných termínov a pridelovanie a preberanie úloh (nielen) v rámci tímu [2]. Toto je výhodné aj z pohľadu dátovej komunikácie, kedy server môže obstarávať aj riadenie verzii. Zdieľaná databáza úloh môže, okrem iného, zase poskytnúť členovi tímu, ktorý má problém s riešením danej úlohy, informácie o tom, na čom pracujú jeho tímoví kolegovia, prípadne kto pracoval na niečom podobnom a mohol by poskytnúť kvalifikovanú radu. Toto je obzvlášť pravda v prípade testerov, ktorí takto môžu zistiť kto implementoval kód, ktorý majú testovať, a následne zistiť, aké testy už boli vykonané a aké je ešte potrebné vykonať.

Nemenej dôležitou súčasťou dnešných softvérových nástrojov je vizualizácia stavu projektu. Tento významný aspekt vývoja softvérového projektu je kľúčom ku kvalitnému manažmentu softvéru, pretože vďaka nemu je možné pružne reagovať na nedostatky a sklzy. Skúsenému manažérovi stačí výstup z niektorého softvérového nástroja na presnú identifikáciu rizika a zvolenie vhodnej stratégie jeho riešenia. Aj preto sa dnešné nástroje snažia do istej miery automatizovať proces znázornenia stavu projektu a to generovaním napríklad tzv. Ganttových diagramov, alebo tzv. „Burndown charts“, využívaných najmä pri agilných metodikách vývoja [1]. Ganttov diagram, znázornený na obr. 1. je diagram na ktorom sú znázornené jednotlivé úlohy a ich umiestnenie v čase. Na tomto diagrame, vytvorenom nástrojom RedMine, je znázornená aj časť splnenia konkrétnej úlohy pomocou farebného odlíšenia splnenej časti (modrá farba) a zostávajúcej časti (červená farba). Na obr. 2. je vyobrazený „Burndown chart“ [1], čo je graf znázorňujúci „spaľovanie“ vopred určenej metriky reprezentujúcej úsilie potrebné na dosiahnutie cieľa, v tomto prípade človekohodín.

Pri mnohých projektoch, najmä takých, v ktorých figuruje zákazník ako priekupník a nie koncový užívateľ, sa môže zákazník zaujímať o priebežné plnenie úloh. V takomto prípade slúži elektronická evidencia úloh aj ako podklad pre zákazníka.

Gantt



Obr. 1. Ganttov diagram

Burndown for "Test Sprint 15.8.2008"

Please define the duration of Sprint

Start Date: 2008-07-28 00:00:00

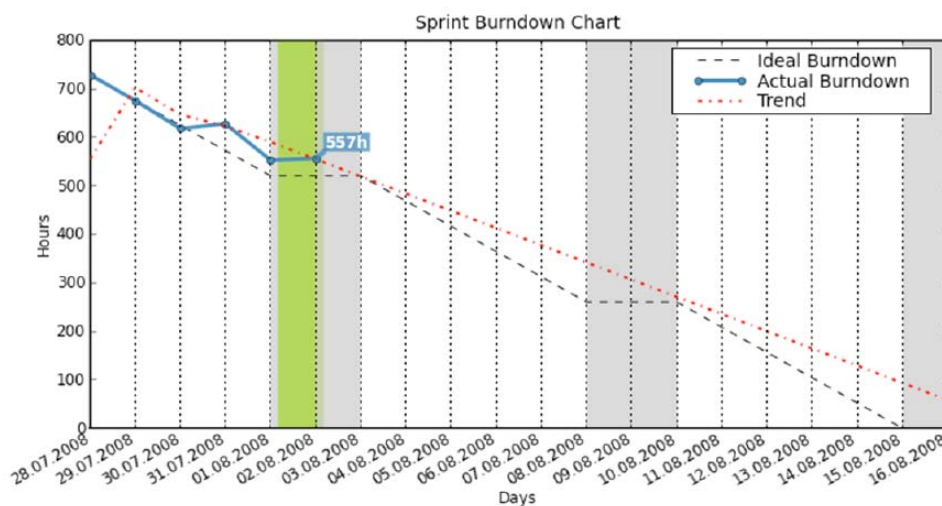
End Date: 2008-08-16 16:04:20

Duration: calendar days , 15 working days

[Change](#)

The **Burndown Chart** in Agilo gives you the actual status of the Sprint.

The team has a real time perception on what is going on, and it can react fast.



Obr. 2. „Burndown chart“

Alternatívny prístup

Hoci aj pri agilnom spôsobe vývoja sa tiež zvyknú používať softvérové nástroje na manažment úloh, nie je to podmienkou úspechu. Mnoho úspešných projektov realizovaných v tzv. „Scrum-e“ sa zaobišlo len s informačnými kartičkami (tzv. „index cards“), zopár veľkými nástenkami a lepiacimi papierikmi (tzv. „stickies“). Jedným z dôvodov prečo odpadá potreba rigorózneho sledovania úloh je, že všetci členovia Scrum tímu, vrátane Scrum Master-a, získavajú na každodenných stretnutiach (tzv. „Daily Scrum“, alebo „Daily standup“) prehľad o tom, na čom pracujú ostatní. Toto umožňuje nielenže lepšiu spoluprácu členov v tíme, ale aj skoré odhalenie „flákačov“ - čo je jeden z dôvodov sledovania úloh u zamestnancov organizácie. Informačné kartičky s názvami úloh zaisťujú túto potrebu sledovania úloh do dostatočnej miery tak, že každá kartička s úlohou je počas šprintu premiestňovaná medzi nástenkami podľa jej stavu - napr. nezačaté, začaté, otestované, a pod. [3]. Takýto prehľad o tom, aká časť úloh pre šprint je dokončená a aká časť ešte ostáva, prispieva k motivácii a sústredenosti členov tímu. Vizualizáciu priebehu šprintu poskytuje „Burndown

chart“, ktorý môže byť priebežne zostavovaný na základe vyššie spomenutých informačných kartičiek.

Záver

V eseji som sa nezaoberal konkrétnymi nástrojmi na riadenie softvérových projektov, ale všeobecne na ich problematiku. V prípade záujmu môže čitateľ nájsť rozbor vhodných kandidátov v [1], alebo [5]. Softvérové nástroje pre riadenie projektov sú takmer nutnosťou pre každého moderného manažéra, ale môžu byť veľmi užitočným pomocníkom aj pre vedúceho vývojárskeho tímu, ako aj pre člena takéhoto tímu. Pri agilnom prístupe k softvérovému vývoju je pri menších projektoch pre tím ako taký možné pracovať efektívnejšie bez nejakej formy elektronického sledovania úloh, avšak je zrejmé, že pri väčších projektoch sa ani agilné metodiky vývoja nevyhnú použitiu softvérového nástroja. Čo sa týka výberu konkrétneho nástroja, záleží na špecifických požiadavkách organizácie, avšak čoraz dôležitejším faktorom pri výbere by mala byť integrácia s existujúcimi systémami fungujúcimi v organizácii a množstvo funkcionality bez ujmy na jednoduchosti použitia.

Použitá literatúra

1. BARLA, Michal. Softvérové nástroje pre manažment v tíme. 2007
2. BLANDFORD, A. E. - Green, T. R. G.: Group and Individual Time Management Tools: What You Get is Not What You Need. *Personal and Ubiquitous Computing*, Vol. 5, No. 4 (December 2001) 213 - 230.
3. KNIBERG, Henrik . *Scrum and XP from the Trenches : How we do scrum*. [s.l.] : Dixie Press, 2007. 127 s. <<http://www.infoq.com/minibooks/scrumpxpfromthetrenches>. ISBN 978-1-4303-2264-1>.
4. MCCONNELL, Steve. Tool Support for Project Tracking. *IEEE Software* [online]. 1997, vol. 14, no. 5 [cit. 2008-10-18], s. 118-120. Dostupný z: <<http://www.stevemccconnell.com/ieeesoftware/bp11.htm>>.
5. MÜHLBAUER, Rudolf, et al. Comparison of Project management software., 2007. 43 s. Dostupný z: <<http://klambauer.info/pms.pdf>>.

Annotation

Support tools for software project management with an emphasis on task management

People participating in software projects usually take advantage of some form of task tracking and management, whether it is using stickies, MS Outlook, or software designed for this purpose. Usage of such software tools can significantly simplify the work of management; however, it can sometimes introduce new hazards. This essay deals with the importance of these tools, as well as the requirements on them. Furthermore the essay analyses the problems relating

to the use of support tools of project management and considers options for avoiding these problems by using agile methods of development.