

Podporné prostriedky na riadenie softvérového projektu, ich dôležitosť a vlastnosti

PETER CSÉFALVAY

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
gomera@ynet.sk*

Abstrakt. Účelom článku je oboznámiť čitateľa o tom, prečo by spoločnosti zaoberajúce sa vývojom softvéru mali používať podporné prostriedky na riadenie softvérového projektu. Čitateľ sa môže tiež dozvedieť, o dôležitosti používania takýchto nástrojov pri vývoji nového softvéru a o tom, ako takéto nástroje znižujú náklady na vývoj softvéru. Článok sa tiež podrobnejšie zmieňuje o jednotlivých vlastnostiach takéhoto softvéru a zdôrazňuje, ktoré vlastnosti je potrebné sledovať pri jeho výbere a následnom nasadení v spoločnosti. Zameriava sa predovšetkým na kľúčové vlastnosti softvéru, ktoré by mal každý takýto softvér obsahovať, ale zmieňuje sa aj o ostatných rozšírených vlastnostiach. Po prečítaní článku by mal čitateľ pochopiť dôležitosť používania takýchto nástrojov pri vytváraní nového softvéru a o vlastnostiach, ktoré by mali takéto nástroje obsahovať.

Prečo používať podporné nástroje?

V súčasnosti je veľmi dôležité, aby softvérové projekty boli dokončené načas a aby sa nepresiahol ich pôvodný rozpočet. Spoločnosti zaoberajúce sa vývojom softvéru si nemôžu dovoliť, aby dodali svoj softvér neskoro. Mohlo by sa stať, že by prišli nielen o jedného zákazníka, ale aj o budúcich potencionálnych zákazníkov tým, že by stratili svoju dobrú povesť.

Avšak skutočná situácia je iná. Veľa projektov nebýva dokončených načas a veľa z nich je stornovaných ešte pred ich dokončením. Podľa Gartner Inc. sídliacej v Stamforde až polovica softvérových projektov nesplní ciele a polovica prekročí rozpočet. Aberdeen Group Inc. z Bostonu informuje, že 30% projektov je stornovaných ešte pred dokončením a 90% projektov je dokončených neskoro [3]. Môže sa zdať, že 90% softvérových produktov dodaných neskoro je priveľa a že toto číslo je prehnané, ale stačí si len spomenúť na firmu Microsoft a jej nový operačný systém (Windows Vista), ktorý má prísť na trh až budúci rok. Pritom pôvodné (prvotné) vyhlásenia boli, že operačný systém bude dokončený v polovici roku 2005. Avšak tento termín sa

niekoľkokrát posunul. Ako vidíme, ani veľké a známe spoločnosti nedodávajú svoje produkty vždy načas a podľa plánu.

Riadenie vývoja softvérového projektu nie je jednoduché a stretáva sa s rôznymi problémami. Veľa softvérových projektov je neúspešných a veľa z nich je stornovaných. Za týmto všetkým stojí predovšetkým zlé riadenie softvérového projektu [2].

Slabý manažment softvérového projektu zapríčiňuje nie len stornovanie projektov, ale aj presiahnutie rozpočtu projektu, neskoré dokončenie projektu, zlú kvalitu produktu, zmeškané príležitosti presadiť svoj produkt na trhu, slabú morálku tímu a v neposlednom rade stratený zisk.

Preto keď si uvedomíme, že každý rok sa na vytvorenie nových softvérových produktov použijú miliardy dolárov, pochopíme, že používanie podporných nástrojov na riadenie softvérového projektu je nevyhnutné.

Treba si však uvedomiť, že kúpenie si niektorého z podporných nástrojov na riadenie softvérového projektu nie je postačujúce. Zvyčajne takáto kúpa nie je úspešná, ak kúpený nástroj nie je spätý s kultúrou danej spoločnosti. Podľa Crawforda spoločnosti trávajú veľa času a mŕňajú veľa peňazí nakupovaním vecí, o ktorých si myslia, že im zlepšia manažment, ale nikdy sa nezaoberajú zadefinovaním toho, čo dobrý manažment je [3]. Firmy by sa v prvom rade mali zamyslieť nad tým, ako zlepšiť manažment a až potom by mali rozmýšľať o tom, či je potrebné kúpiť nejaký podporný nástroj alebo nie a ak áno, aké by mali mať naň požiadavky.

Použiť alebo nepoužiť podporné nástroje v malom tíme?

Existuje veľké množstvo podporných nástrojov na riadenie softvérového projektu. Tieto nástroje obsahujú veľké množstvo funkcií. Avšak veľa z týchto funkcií nie je v malom tíme potrebných, sú potrebné len pre veľké tímy vývojových pracovníkov.

Preto si treba položiť otázku, či je pre tím potrebné použiť niektorý z podporných nástrojov. Túto otázku si kladie snád väčšina manažérov. Niektorí manažéri majú pocit, že takéto nástroje je zbytočné používať v malom tíme ľudí a že iba zbytočne zdržiavajú vývoj softvéru. Malý tím má zvyčajne od dvoch do osem ľudí, a preto sa manažérom môže zdať, že takýto tím je jednoduché riadiť a je jednoduché sledovať, čo kto kedy spravil.

Opak však môže byť pravdou. Podľa môjho názoru je dôležité, aby sa aj v malom tíme používali podporné nástroje. Tieto nástroje sa môžu využiť hlavne na pridelovanie úloh jednotlivým vývojovým pracovníkom a tiež na sledovanie, ako prebieha vývoj softvéru. Manažér tímu tak má ucelený prehľad o tom, ako projekt napreduje. Môže tak jednoznačne určiť, či sa projekt stihne načas dokončiť. Ak vidí, že projekt napreduje pomalšie ako má, môže tím rozšíriť o ďalšieho, príp. ďalších vývojových pracovníkov, a tak doceliť, aby bol softvér dokončený načas. Takto má tiež prehľad o tom, kto čo urobil a teda môže zistiť, či v tíme nemá človeka, ktorý si neplní úlohy. Tohto človeka môže z tímu vylúčiť a nahradiť ho iným, ktorý dokáže plniť úlohy načas.

Ak sa aj manažér rozhodne, že použije nejaký podporný nástroj, je dôležité, aby si vybral taký, ktorý bude vyhovovať celému tímu. Preto je potrebné, aby projektový manažér rozumel tomu, čo tím robí a čo potrebuje. Ak vyberie nástroj len preto, aby spoločnosť používala nejaký nástroj, dosiahne len to, že nástroj nesplní svoj účel a bude len tím spomaľovať pri práci.

Manažér musí tiež zobrať do úvahy, aké náročné je používať daný nástroj. Ako vieme, používatelia sa neradi učia nové veci a vôbec nemajú radi, ak sú programy komplikované a ťažkopádne. Používateľ sa danému programu zvyčajne venuje len pár minút a ak za ten čas nezistí, ako sa dá s programom efektívne pracovať a že mu program pomôže pri práci, odsúdi ho a bude chcieť používať iný, resp. žiadny. Preto je potrebné, aby manažér rozanalyzoval situáciu a vybral nástroj tak, aby vývojovým pracovníkom nebol nespôsobil problém.

Vlastnosti podporných nástrojov

Ako už bolo spomenuté, podporných nástrojov na riadenie softvérového projektu existuje veľa. Podporné nástroje môžeme vo všeobecnosti rozdeliť na niekoľko druhov: nástroje na plánovanie, nástroje na rozdeľovanie zdrojov, nástroje na spoluprácu a komunikáciu a nástroje na dokumentáciu.

Čo je ale potrebné, aby podporné nástroje obsahovali? Pre manažéra je dôležité, aby mal nástroj na tieto špecifické úlohy pri vývoji softvéru: odhadovanie, plánovanie, sledovanie, manažment rizík, porovnávanie [4]. Z toho teda vyplývajú aj jednotlivé požiadavky, ktoré by mal podporný nástroj obsahovať:

- podpora viacerých projektov, tímov, členov tímov
- úlohy, ich plánovanie, pridelovanie a monitorovanie
- vizualizácia projektu
- repozitár (miesto, kde sú uložené a udržiavané dáta)

Podporné nástroje môžu ďalej obsahovať:

- wiki, diskusné fórum
- Ganttov diagram
- kalendár
- matica zodpovednosti
- generovanie súhrnných správ (tzv. reporty)
- nástroje na vykazovanie práce
- stretnutia
- notifikácia e-mailom, na icq klienta a pod.

Samozrejme nemôžeme sa striktne držať toho, že všetko, čo by malo byť súčasťou nástroja aj musí byť a naopak, čo by mohlo, nemusí byť. O tom, čo by mal nástroj obsahovať a čo nemusí, rozhodne projektový manažér.

V ďalších kapitolách sa budeme podrobnejšie venovať niektorým zaujímavým vlastnostiam podporných nástrojov.

Pridelovanie a monitorovanie úloh

Ako vieme, pri plánovaní sa zadefinujú jednotlivé úlohy, ktoré je potrebné spraviť. Každý člen tímu má pridelený zoznam úloh, ktoré je potrebné spraviť. Môže sa jednať o naprogramovanie nejakej časti kódu, opravenie chyby, pridanie testu, napísanie dokumentácie a pod. Avšak kvôli sledovaniu projektu nie je žiadúce, aby každý člen tímu mal svoj vlastný zoznam úloh a pracoval s ním ako so zoznamom úloh, ktoré treba urobiť (tzv. todo list). Tak by nebolo možné sledovať, čo kto kedy urobil a manažér by tak nemal prehľad o projekte. Preto každá úloha by mala byť definovaná tým, čo sa má spraviť, kto to má spraviť a do kedy má byť úloha splnená.

Na tento účel slúžia nástroje na monitorovanie úloh (issue tracking tools). Tieto nástroje sú priamo špecializované na monitorovanie úloh a na pridávanie nových úloh tak, ako už bolo spomenuté. Typicky takýto nástroj obsahuje zoznam úloh, súhrnné správy, míľniky (milestones), správy o výdavkoch a pod. Avšak táto technika poskytuje slabé informácie o stave projektu – iba výdavky a plánovanie. Ak však nemonitorujeme zároveň výdavky, plánovanie, kvalitu a funkcionálnosť softvéru, tak nemonitorujeme projekt [4]. Ak monitorujeme iba, či sa úlohy plnia v stanovenom čase (monitorujeme teda iba funkcionálnosť produktu v čase), tak úlohy, ktoré sú označené ako „urobené“, sú zvyčajne nízkej kvality. Vývojoví pracovníci sa snažia zvyčajne vynaložiť čo najmenej úsilia na to, aby označili úlohu za splnenú. Avšak neskôr sa zistí, že práca, ktorá bola označená ako „hotová“, je v skutočnosti nedokončená a projekt tak začína postupne zaostávať za plánom. Preto je dôležité, aby sa zároveň monitorovala aj kvalita splnenia úlohy a zároveň aj celého projektu a nie len funkcionálnosť vyvíjaného produktu.

Ak je už projekt rozbehnutý, musí byť v každom okamihu vidieť, v akom stave sa projekt nachádza - musia byť hneď zrejmé náklady na projekt, splnenie jednotlivých úloh, funkcionálnosť vytváraného systému a pod. Ak manažér nemonitoruje projekt pravidelne, je to, ako keby ho ani neriadil [4].

Úloha manažéra je teda sledovať, ako projekt napreduje. Je potrebné predovšetkým sledovať, či sa úlohy plnia načas. Ak by sa neplnili, je pravdepodobné, že projekt sa nestihne dokončiť načas a softvérová firma bude musieť zaplatiť penále za omeškanie dodania softvéru. Tým sa jej samozrejme zvyšujú náklady a znižuje zisk, čo nie je žiadúce. Preto, ak manažér zistí, že úlohy sa nestíhajú plniť načas, do tímu zintegruje ďalšieho vývojového pracovníka. Ak sa jedná iba o jednu osobu, ktorá neustále nestíha plniť pridelené úlohy, je potrebné túto osobu nahradiť inou. Tak sa dosiahne, že projekt sa dokončí na čas.

Repozitár

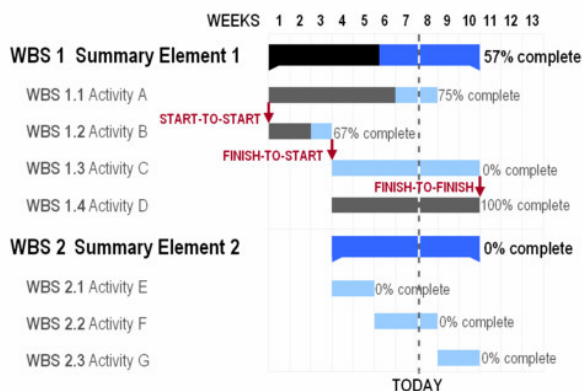
Repozitár je dôležitou súčasťou podporných nástrojov. Ako už bolo spomenuté, je určený na ukladanie a správu dát. Dáta sú uložené centrálne, zvyčajne na nejakom serveri. Dátami môžu byť rôzne zdrojové súbory alebo dokumentácia. Tieto dáta sú výstupom jednotlivých úloh a je potrebné ich monitorovať, ako už bolo spomenuté v predchádzajúcej kapitole. Musí sa monitorovať predovšetkým kvalita týchto výstupov.

V repozitári sa nenachádzajú len posledné (aktuálne) verzie súborov, ale aj verzie predošlé. Zvyčajne je možné jednotlivé verzie navzájom porovnávať. Tak môže projektový manažér monitorovať, ako projekt napredoval v čase. Zároveň vie tiež určiť, kto a kedy urobil akú zmenu v danom súbore. Tak sa dá skontrolovať, ako si jednotliví členovia tímu plnia svoje úlohy.

Vizualizácia projektu

Je ťažké si predstaviť, ako by sa dal projekt vizualizovať. Samotný projekt sa vizualizovať nedá, avšak dá sa vizualizovať stav, v akom sa projekt nachádza. Dost' často sa vizualizuje, ako sa splnil plán dokončenia jednotlivých úloh v čase, príp. aj v akom stave dokončenia sa nachádzajú.

Predovšetkým sa dá vizualizovať časový plán, v akom poradí a v akej náväznosti na seba sa budú jednotlivé úlohy plniť. Na tento účel sa najčastejšie používa Ganttov diagram, ktorý je znázornený na Obr. 1. Ganttov diagram zobrazuje dátum, kedy sa má nejaká činnosť alebo úloha začať a kedy má skončiť, taktiež aj dĺžku trvania jednotlivých činností, ale zvyčajne nezobrazuje závislosti jednotlivých činností (úloh) [1]. Ganttov diagram znázornený na Obr. 1 znázorňuje nie len to, kedy sa má ktorá činnosť začať a skončiť, ale aj náväznosti jednotlivých činností a taktiež aj stav dokončenia jednotlivých činností (úloh). Ganttov diagram je relatívne jednoduchý na čítanie a preto sa dost' často používa v manažérskych prezentáciách.



Obr. 1. Ganttov diagram.

Vizualizácia projektu sa dá tiež dosiahnuť pomocou kalendára, ktorý pre každý deň zobrazí zoznam úloh, ktoré sa majú splniť do daného dňa.

Ľudské zdroje

Už pri plánovaní softvérového projektu je potrebné myslieť na ľudské zdroje. Je potrebné mať v tíme ľudí, ktorí sú schopní vytvoriť softvér a sú ho schopní dokončiť načas. Nájsť teda schopných ľudí je dôležitá úloha. Ale na to, aby bol projekt úspešný, treba viac. Treba, aby členovia tímu vedeli navzájom spolupracovať, komunikovať, vymieňať si jednotlivé výstupy úloh a pod. Na to sú potrebné podporné nástroje, ktoré slúžia na riadenie softvérového projektu. Bez nich by aj tí najlepší členovia tímu nemali šancu uspieť pri vývoji softvéru. Je preto potrebné, aby sa takéto nástroje sústreďovali aj na manažment ľudských zdrojov. Manažér by tak mal k dispozícii informácie o jednotlivých členoch tímu, ale aj o externých vývojových pracovníkoch, ktorých by v prípade potreby mohol pridať k tímu.

Jednou z možností ako uskutočniť plánovanie ľudských zdrojov, je použiť maticu zodpovednosti, ktorá je zobrazená na Obr. 2. Role v tíme a zodpovednosti jednotlivých osôb sa môžu reprezentovať práve pomocou matice zodpovednosti. V riadkoch sa nachádzajú zodpovednosti a v stĺpcoch sa nachádzajú členovia tímu, ktorí sú za danú úlohu alebo činnosť zodpovední. Vo veľkých projektoch môže matica zodpovednosti byť na rôznych úrovniach. Napr. na vyššej úrovni matice zodpovednosti sa definuje, ktoré skupiny ľudí sú zodpovedné za ktorú časť a na nižšej úrovni sa v rámci skupiny priradia role a zodpovednosti za špecifické činnosti príslušným členom tímu [1].

PERSON PHASE	A	B	C	D	E	F	...
Requirements	S	R	A	P	P		
Functional	S		A	P		P	
Design	S		R	A	I		P
Development		R	S	A		P	P
Testing			S	P	I	A	P

P = Participant A = Accountable R = Review required
I = Input required S = Sign-off required

Obr. 2. Matica zodpovednosti [1].

Komunikácia a kolaborácia medzi členmi tímu

Pri vytváraní softvéru je dôležité, aby jednotliví členovia tímu medzi sebou komunikovali. Pri tom nemusí ísť o rozhovor medzi dvoma alebo viacerými ľuďmi. Podporné nástroje by mali podporovať efektívnu komunikáciu medzi jednotlivými členmi tímu alebo medzi vývojovými pracovníkmi a zákazníkom. Pod pojmom komunikácia sa nerozumie len internetový rozhovor (chat, instant messenger a pod.), ale aj iné formy.

Už sme spomenuli napr. použitie repozitárov. Tu si užívatelia môžu navzájom vymieňať jednotlivé artefakty (výstupy) jednotlivých činností alebo úloh. Taktiež sa tu nachádza dokumentácia k projektu, čo je tiež nepriama komunikácia medzi členmi vývojového tímu.

Inou možnosťou komunikácie je napr. použitie diskusných fór, kde členovia tímu môžu riešiť jednotlivé problémy.

Iným spôsobom je napr. Wiki (kolaboratívna tvorba webových stránok), kde jednotliví členovia tímu môžu postupne upravovať webové stránky, a tak vytvárať spoločnú dokumentáciu.

Záver

Cieľom práce bolo oboznámiť čitateľa s podpornými nástrojmi na riadenie softvérového projektu. Zistili sme, že tieto nástroje nie sú dôležité len vo veľkých tímoch, ale taktiež aj v malých tímoch. Podporné nástroje slúžia nie len na to, aby sme mali prehľad o tom, v akom stave vývoja sa softvér nachádza, ale aj na zníženie nákladov projektu, dodržanie termínu dokončenia, zlepšenie kvality softvéru, zvýšenie zisku a pod.

Dozvedeli sme sa, že dobrý podporný nástroj na riadenie softvérového projektu by mal obsahovať podporu viacerých projektov, tímov, členov tímu, musí vedieť vytvárať úlohy, sledovať ich splnenie, vizualizovať projekt a mal by obsahovať repozitár, kde budú jednotlivé verzie zdrojových súborov a dokumentácie.

Použitá literatúra

1. Duncan, W.R.: A Guide to the Project Management Body of Knowledge. Project Management Institute, 1996. ISBN 1-880410-12-5
2. Fox, L.T., Spence, J.W.: The Effect of Decision Style on the Use of a Project Management Tool: An Empirical Laboratory Study. *ACM SIGMIS Database*, Vol. 36, No. 2 (2005) 28-42.
3. Fox, P.: Tapping the Right Tools. *ComputerWorld*, Vol. 36, No. 17 (2002) 43.
4. McConnel, S.: The Software Manager's Toolkit. *IEEE Software*, Vol. 17, No. 4 (2000) 5-7.

Annotation*Project management tools, their importance and features*

The main purpose of this paper is to discuss why software companies should use project management tools. The reader can learn why these tools are important during software development and how these tools can reduce costs. This thesis also summarizes the properties of these tools and highlights which properties are important and which are not. After reading this paper the reader will know why project management tools are important and which features they should contain.