Problémy a otázky plánovania

ADAM VALENT

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
valent05[zavináč]student[.]fiit[.]stuba[.]sk

Abstrakt. Plánovanie vedie k efektívnemu uskutočneniu mnohých aktivít, či už v bežnom živote, obchode, alebo pri tvorbe softvérového projektu. Podrobnou analýzou je možné získať významný prínos, na rozdiel od bezhlavého napredovania bez účinnej stratégie. Táto práca sa zaoberá najčastejšími problémami, ktoré vznikajú pri plánovaní, aké bariéry musia plány prekonať, popisuje prístupy k plánovaniu projektov a aké aspekty sa podieľajú na neúspešnosti projektov. Obchod a IT spolu súvisia a preto je potrebné naplánovať využívanie zdrojov podniku tak, aby nedošlo k ich spotrebovaniu ešte pred dokončením projektov. Orientácia len na požiadavky zákazníkov môže viesť ku spomaleniu, alebo až zastaveniu ostatných projektov. Študenti riešia okrem tímového projektu aj iné projekty v rámci predmetov a preto je potrebné stanoviť si časové možnosti a priority, pretože vplyv pridávania projektov pôsobí na efektívnosť práce celého tímu. Plánovanie je preto v tomto smere značne dynamické.

Čo je to projekt

O rôznych projektoch počujeme na každom kroku, avšak každá skupina ľudí poníma slovo projekt pod rôznym významom. Považujem za potrebné pre začiatok uviesť význam tohto slova na správnu mieru. Projekt vždy zamestnáva skupinu ľudí a taktiež aj ovplyvňuje iné skupiny ľudí. Je vždy spojený s rizikom neúspechu, keďže každý projekt je vždy jedinečný a nikdy presne nevieme, čo sa môže v priebehu jeho realizácie udiať. Podľa definície projektu [7] je projekt jedinečný proces zložený z rady koordinovaných a riadených činností s dátumom začiatku a ukončenia, uskutočňovaný pre dosiahnutie cieľa, ktorý vyhovuje špecifickým požiadavkám, vrátane obmedzení daných časom, nákladmi a zdrojmi. Aby sme dokázali projekt úspešne zrealizovať, je potrebné ho riadiť a riadenie je úzko späté s plánovaním projektu.

Rozhodnutie či plánovať, alebo neplánovať myslím, že nie je potrebné rozoberať. Treba si uvedomiť, že plánovanie sa snaží znížiť neurčitosť, zlepšiť výkonnosť, zaistiť lepšie pochopenie cieľov projektu a vytvoriť základňu pre sledovanie a riadenie práce. Najmä ak ide o projekty väčších rozmerov, zavrhnutie plánovania by znamenalo pracovný chaos. Zoberme si ako príklad situáciu z bežného života, teda prácu

robotníkov na stavenisku. Ich práca musí byť koordinovaná a vopred naplánovaná, v opačnom prípade by sa budova nikdy nepostavila a každý by robil všetko a zároveň nikto by nespravil nič. Výsledok práce by bol teda nulový. Pri plánovaní sa definujú požiadavky na zdroje, požiadavky na prácu a definuje sa kvalita a kvantita práce. Plánovanie je intenzívne hlavne v počiatočných fázach projektu, v priebehu uskutočňovania a riadenia sa plány upravujú podľa potreby. Dobrým príkladom plánovania môže byť napríklad správanie sa včiel priamo v živočíšnej ríši. Každý jedinec má svoju vopred danú úlohu a ich vzájomná kooperácia vedie ku spoločnému úspechu a prežitiu celej včelej rodiny. Plánovanie obvykle predstavuje malú časť z celkovo vynaloženej práce, ale môže predstavovať pre projekt značné riziko. To však neznamená, že je nutné vytvárať vyčerpávajúce plány, ale práve naopak, plánovanie by malo byť tak podrobné, ako je to potrebné, a nie tak, ako je to možné.

Softvérová kríza

Už v 60-tych rokoch sa začali objavovať problémy s vývojom rozsiahlejších programov. Tieto problémy vyústili k zavedeniu pojmov softvérové inžinierstvo a softvérová kríza na konferenciách v rokoch 1968 a 1969. Táto kríza sa prejavovala a stále sa prejavuje neúnosným predlžovaním a predražovaním projektov, nízkou kvalitou výsledných produktov, problematickou údržbou a nízkou produktivitou práce programátorov. O trvaní krízy a o miere jej závažnosti sa môžeme presvedčiť zo štúdie Standish Group Report [9] z roku 1995. Uvádzam tabuľku preukazujúcu prekročenie času vývoja.

Tab.1. Prekročenie času vývoja projektu [9]

Prekročenie času o	Projektov
Menej než 20%	13,9 %
21- 50 %	18,3 %
51-100 %	20,0 %
101-200 %	35,5 %
201-400%	11,2 %
Viac ako 400%	1,1 %

Z uvedenej štúdie vyplýva [6], že priemerný projekt stál v porovnaní s pôvodným plánom o 89% viac, jeho vývoj trval 2,22 krát dlhšie a výsledný produkt poskytoval len 61% funkčnosti. Ak to zhrnieme, môžeme povedať, že softvérový projekt bol takmer 7 krát horší ako sa pôvodne plánovalo. Softvérová kríza bola z môjho pohľadu potrebná pre výskum programovacích praktík, životného cyklu softvérového projektu, metodológií pre vývoj softvéru, vďaka čomu je možné dnes vyvíjať neporovnateľne zložitejšie systémy ako v minulosti. Táto skutočnosť vedie v konečnom dôsledku ku zvýšeniu efektivity práce, sledovaniu kvality softvéru, zníženiu nákladov a lepšiemu využitiu zdrojov.

Príčiny zlyhania projektu

Problémy, ktoré môžu nastať je dôležité identifikovať a uvedomiť si ich už pri ich vzniku, v opačnom prípade môže byť ich dopad na projekt katastrofálny. Pokiaľ si však možné problémy uvedomíme vopred a budeme s nimi počítať pri plánovaní, ich riešenie je možné hľadať tak dlho, pokiaľ sa nejaké prijateľné nenájde. Extrémnou situáciou by bolo odmietnutie projektu hneď na začiatku, avšak bolo by to stále efektnejšie, ako zistiť až po vynaložení nemalých časových a peňažných prostriedkov, že projekt nie je možné dokončiť. Vybral som niektoré udalosti [9], s ktorými je možné sa počas vývoju softvérového systému stretnúť a predchádzajú zastaveniu projektu. Niektoré z nich sú potrebné, niektoré kritické, iným sa dá vyhnúť. Číslo v zátvorke udáva percento zastavených projektov, kvôli danému problému na vzorke 365 firiem a 8380 aplikácií.

- Neúplnosť alebo nejasnosť požiadaviek (13,1 %)
- Nedostatok záujmu a podpory zo strany používateľa (12,4 %)
- Nedostatok zdrojov, t.j. podhodnotený rozpočet a krátke termíny (10,6 %)
- Nerealistické očakávania (9,9 %)
- Malá podpora od vedenia dodávateľa alebo odberateľa (9,3 %)
- Zmena požiadaviek a špecifikácie (8,7 %)
- Nedostatočné plánovanie (8,1 %)
- Vyvíjaný systém už nie je potrebný (7,5 %)

Obecne existujú viac ako 1 alebo 2 príčiny zlyhania projektu, ktoré môžu byť spôsobené kombináciou technických, plánovacích alebo obchodných rozhodnutí. Väčšina týchto príčin je úzko spätá s plánovaním a treba pri plánovaní vytvoriť taký plán, ktorý bude vedieť dynamicky reagovať aj na vzniknuté problémy.

Dovolím si tvrdiť podľa [1] a z osobnej skúsenosti, že jednou z dôležitých príčin zlyhania je taktiež komunikácia. Či už komunikácia medzi členmi tímu, ktorá môže viesť k nepochopeniu úloh, alebo samotná komunikácia medzi zákazníkom a manažérom, prípadne medzi používateľmi a projektantmi. Komunikácia zohráva dôležitú úlohu pri plánovaní a distribuovaní informácií, preto ju nemožno pri projekte podceniť.

Častým rozhodnutím v prípade časovej núdze pri riešení projektu je pridávanie ďalších ľudí [8]. V knihe s názvom „The Mystical Man-Month“ od pána Brooksa sa uvádza, že ak máme projekt, ktorý dokončia dvaja ľudia za 6 mesiacov, nie je možné jednoduchým pridaním ďalšej osoby znížiť trvanie projektu na 4 mesiace. Pridaním človeka do projektu na mesiac neznamená, že sa zredukuje práca o jeden človeko-mesiac. Treba si uvedomiť, že ak človek na podobnom projekte ešte nepracoval, tak istú časovú réžiu mu zaberie naštudovanie problematiky, komunikácia a kooperácia s tímom. Brooksov zákon hovorí, že pridanie ľudských zdrojov do meškajúceho projektu ho omešká ešte viac. S týmto tvrdením sa celkom nestotožňujem, závisí to však od

mnohých faktorov projektu. Ak je potrebné dokončiť úlohu projektu s ktorou sa už človek, ktorý má byť do projektu už v minulosti stretol a pozná riešenie problému, nevidím problém v tom, aby tento človek nedokončil dané riešenie ešte skôr, ako ktorýkoľvek iný člen tímu, ktorý pracuje na projekte od začiatku. V prípade potreby dokončiť grafickú stránku systému, je taktiež možné pridať ďalšieho človeka, čo prácu značne urýchli.

Dôležité faktory pre úspech softvérových projektov

Teraz si pre zmenu uvedieme niekoľko faktorov, ktoré prispievajú k úspešnosti softvérových projektov [7]:

- zainteresovanosť členov tímu
- záujem a zapojenie používateľov
- jasne a presne definované požiadavky na systém
- dobré plánovanie
- realistické očakávania
- správna dekompozícia úlohy
- kompetentnosť zúčastnených

Kvalita ľubovoľného výrobku je určovaná tromi základnými faktormi a to cenou, časom a splnením požiadaviek. Tieto faktory si však navzájom konkurujú a preto sa stretávame so zložitosťou, prispôsobivosťou, nestálosťou a neviditeľnosťou softvérového produktu o ktorých je možné nájsť z dostupných zdrojov veľa informácií, preto ich nebudem bližšie rozoberať.

Čo majú úspešné projekty spoločné?

V otázkach riadenia IT projektov nie je nič dôležitejšie, ako bohaté, hlboké a praktické skúsenosti. Podľa už spomínanej spoločnosti Standish Group sú v zozname faktorov ovplyvňujúcich úspešnosť IT projektov hodnotené skúsenosti projektového manažéra vyššie, ako definovanie jasných obchodných cieľov. Dobrý manažér na správnom mieste dokáže výrazne zvýšiť šance na úspech projektu. Až ohromujúcich 97 percent úspešných IT projektov má jedného skúseného projektového manažéra. Dobrý IT projektový manažér dokáže organizovať zdroje, pripravovať a dokumentovať detailné pracovné postupy, schvaľovať vykonanú prácu a riadiť všetko. Od zadefinovania rozsahu projektu cez jeho náklady, časový harmonogram, kvalitu, tímový výkon až po riziká a to všetko pri paralelnej komunikácii so všetkými zaangažovanými subjektmi. V menších tímoch je preto užitočné, aby funkciu manažéra projektu vykonával človek, ktorý má už skúsenosti s prácou v tíme a prešiel viacero oblastí pri tvorbe softvérového produktu. Na druhej strane ak tento člen tímu má skúsenosti, avšak nedokáže jednať s ľuďmi a má značné problémy s komunikáciou, môže byť táto výhoda skôr prekážkou

ako pomocou. Pre manažéra nie sú dôležité hlboké technické znalosti, ale skôr umenie pracovať s ľuďmi. Čiže aj dobré plány môžu zlyhať pri nesprávnej interpretácii členom tímu.

Životný cyklus projektu

Projekty bývajú rôznych typov, pričom každý jeden je jedinečný. Jeho životný cyklus závisí od typu, veľkosti a rôznych faktorov, ktoré vstupujú do hry počas práce na projekte. Každému projektu vyhovuje iný model životného cyklu, pričom voľbou nesprávneho modelu dávame projektu len veľmi malú pravdepodobnosť úspechu. Ak plánujeme projekt, ponúkajú sa nám v podstate 4 základné modely[2] životného cyklu: vodopádový, inkrementálny, špirálovitý a evolučný.

Najjednoduchším modelom je *vodopádový*, čiže sekvenčný životný cyklus. Všetky aktivity projektu je možné usporiadať do určitých po sebe idúcich sekvencií. Výstupom jednej fázy je vstupom do druhej fázy, a preto je potrebné všetky činnosti prvej fázy ukončiť a zdokumentovať predtým, než sa začne ďalšia fáza. Tento model nezodpovedá reálnym požiadavkám projektu, kedy sa jednotlivé etapy projektu prekrývajú, alebo sú usporiadané v inom poradí. Niekedy sa stáva, že zákazník si vyžiada viacero scenárov najmä z analytickej fázy projektu a na základe predložených dokumentov sa rozhodne o ďalšom postupe.

Inkrementálny model je podobný iteratívnemu modelu. Na základe špecifikácie celého systému sa v inkrementálnom modeli stanovia ucelené časti systému - podprojekty, ktoré sa vytvárajú a odovzdávajú používateľovi postupne. Štruktúra systému môže byť navrhnutá lepšie ako pri použití iteratívneho modelu, pričom neskoršie upravenie požiadaviek zákazníka nemá negatívny dopad ako pri vodopádovom modeli. Tieto podprojekty môžu byť zo strany zákazníka akceptované aj skôr, avšak s finálnym zavedením systému do prevádzky sa čaká na dokončenie všetkých projektov.

Pokiaľ zákazník ešte nevie sformulovať a predložiť svoje požiadavky, je vhodné použiť *špirálovitý model*. Nepresná špecifikácia môže však znamenať neúspech projektu, preto sa v počiatočných fázach zbierajú údaje v prostredí zákazníka a vytvárajú sa možné návrhy, aby sa odhalili všetky riešenia. Na základe získaných poznatkov sa potom definuje predmet projektu. Pri použití tohto modelu vznikajú spustiteľné verzie veľmi skoro, takže chyby je možné odhaliť hneď na začiatku, čo umožňuje znížiť cenu ich odstraňovania. Tento model je však náročný na riadenie a ťažko sa dá z neho získať celkový prehľad o projekte. Tento model je možné využiť vo veľkých organizáciách, kde zákazník nevie pevne stanoviť požiadavky na dlhšie obdobie. Opakovaním jednotlivých etáp sa však projekt môže značne predražiť.

Vhodnejším životným cyklom je *evolučný model*. Ide o hybridný model, ktorý umožňuje priebežne prispôsobovať produkt požiadavkám zákazníka. Tím rozdelí riešenie do viacero častí na základe očakávaní zákazníka, ktoré majú pre projekt prínos. Jednotlivé časti sa vyvíjajú a dodávajú zákazníkovi separátne. V priebehu riešenia bežne dochádza k zmenám zvoleného životného cyklu. Môžeme naraziť na problém aktuálnosti špecifikácie, keďže prebehla na začiatku. V prípade zmien si ale

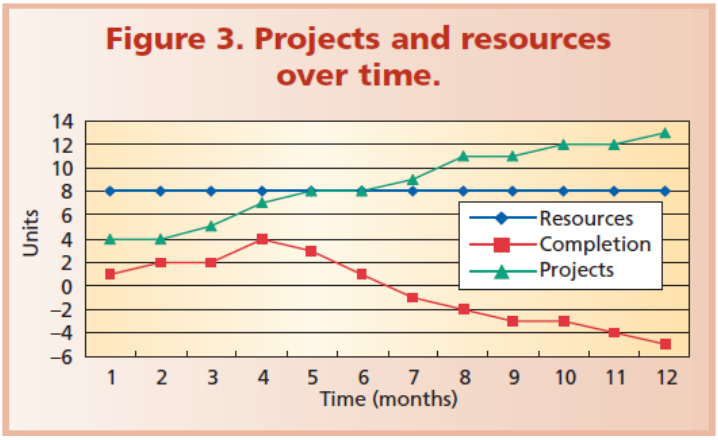
musíme uvedomiť, že nemeníme celý projekt, ale jednotlivé časti a ak sa na toto nezabudne pri analýze, z tohto problému sa stáva triviálna záležitosť.

Súvislosť obchodu a IT plánovania

Plán IT aktivít musí byť riadený obchodom, ale vedený IT oddelením, ktoré stanovuje jeho priority v plánovaní. Bez tohto kroku sa budú IT úlohy nachádzať v stave organizovaného chaosu. Úlohy pre daný mesiac sú jednoznačné, ale menia sa v závislosti od požiadaviek koncových používateľov. Ak sa tieto potreby nebudú riešiť, môžu nastať v IT portfóliu firmy problémy.

Na začiatku roka si podnik stanovuje zoznam IT cieľov a určuje ľudí zodpovedných za požiadavky zákazníkov. Zakaždým, len čo zmení zákazník požiadavky alebo nastanú zmeny v implementácii, je potrebné tieto požiadavky zapracovať do plánov. Keďže IT oddelenie má pridelené určité zdroje a stanovený pevný dátum pre dokončenie projektov, každé narušenie projektu spôsobuje zmenu portfólia a spotrebu zdrojov určených pre rozbehnuté projekty. Optimálne si IT oddelenie vie udržať vývoj projektov s obmedzenými zdrojmi a pevne stanoveným časovým horizontom, častokrát však zmeny požiadaviek zákazníkov spôsobujú problémy.

Ako je uvedené v príklade [3] na obrázku č.1, typickým príkladom môže byť riešenie 4 projektov počas roku, avšak podnik dokončí len 2 počas roka. Podnik pracuje na požiadavkách zákazníkov, zatiaľ čo vývoj ostatných projektov sa približuje nulovej hodnote. Toto je typický príklad, ako môže nastať špirála služieb orientovaných na zákazníka, ktorá vedie k neúspechu. Tým, že si podnik zoberie viacej projektov, v domnienke, že sa tým orientuje na zákazníka, v podstate kvalita služieb upadá nedodaním produktov načas.

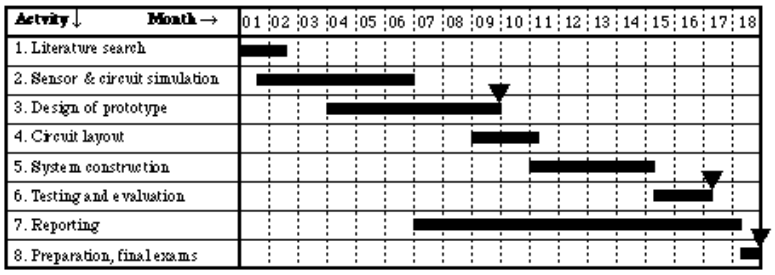


Obr. 1. Vplyv pridávania projektov na dokončenie úloh [3]

Je preto podľa mňa potrebné, aby v podniku existovalo schvaľovanie procesov a určovanie ich priority. Pridávaním ďalších aktivít sa rozdelia projekty na časti, zoradia sa podľa požiadaviek a ostatné aktivity môžu byť odklonené, alebo odložené na neskôr. Schvaľovanie procesov a určovanie priorít sa týka ako IT odvetvia, tak aj obchodu. S týmto problémom sa stretáva veľa tímov, keďže reálne sa vždy rieši viacero problémov a členovia tímu majú okrem práce na projekte taktiež aj iné povinnosti. Preto je užitočné si vytvárať osobný plán, vyhodnocovať dokončené úlohy, vedieť si stanoviť prioritu jednotlivých úloh a využiť tak dostupné časové možnosti a optimálne využiť všetky zdroje.

Ganttov diagram

Častou formou plánovania je používanie Ganttovho diagramu[5]. Je to grafická metóda, ktorej ideou je zobraziť dátumy na jednej osi a úlohy so zdrojmi na druhej. Graf znázorňuje kedy má byť úloha splnená v porovnaní s aktuálnym stavom ostatných úloh. Z diagramu sa dá ľahko vyčítať, čo všetko bolo v procese projektu už dokončené, odhadnúť, či má daná činnosť meškanie, alebo prebieha načas a zistiť v akom stave sa práve nachádza. Ganttov diagram je v súčasnosti najpoužívanejšia forma prezentácie projektových modelov pre plánovanie a riadenie rozsiahlych projektov.



Obr. 2. Príklad Ganttovho diagramu [5]

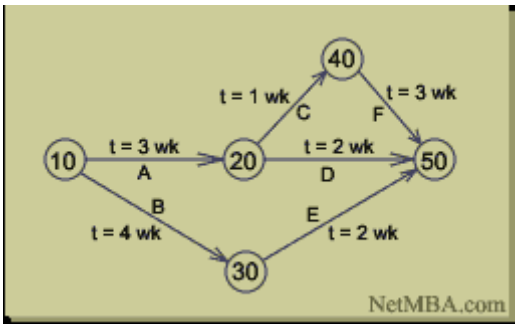
Pert diagram

Ďalším spôsobom plánovania je za pomoci využitia PERT (Program Evaluation and Review Technique – PERT) diagramu[4]. Tento diagram využíva metódu kritickej cesty a štatistických pravdepodobností na výpočet priemerného trvania jednotlivých úloh. Pri vytváraní diagramu sa určujú činnosti, ich míľniky a nadväznosti. Je to uzlovo definovaný graf, ktorý graficky prezentuje závislosti úloh. Obvykle sa do PERT diagramu zadáva optimistická, najpravdepodobnejšia a pesimistická doba trvania úloh. Očakávaný čas trvania potom vypočítame podľa vzorca:
Očakávaný čas = (Optimistický odhad + 4 x najpravdepodobnejší odhad + Pesimistický odhad) / 6

Kritická cesta sa určuje spočítaním časov v každej sekvencii činností a určením najdlhšej možnej cesty. Dostaneme tak čas potrebný na dokončenie projektu. Využitie

tohto diagramu nájde uplatnenie u špecialistov na sieťovú analýzu. Jeho hlavnými výhodami sú:

- dátumy začiatkov a ukončenia činností
- predpokladaný čas dokončenia projektu
- pravdepodobnosť dokončenia pred špecifikovaným dátumom
- určenie činností kritickej cesty, ktoré majú priamy dopad na čas dokončenia
- určenie aktivít, ktoré nemajú pevne stanovené časové obmedzenie a môžu tak požičať zdroje pre kritické aktivity



Obr. 3. Príklad diagramu PERT [4]

Každá z metód ponúka svoje výhody a má taktiež aj svoje nevýhody. Najlepším spôsobom by bolo vybrať si výhody z oboch a teda obe metódy skĺbiť. Pokiaľ sa v Ganttovom diagrame znázornia aj míľniky, vznikne nám ideálny diagram na vizualizáciu pre projekty. Neposkytuje nám také podrobné časové informácie o jednotlivých činnostiach ako PERT diagram, ale informácie ktoré je možné z tohto diagramu získať, sú dostatočné. Možným riešením by podľa mňa bola vizualizácia základných úloh v Ganttovom diagrame a na identifikáciu časových intervalov týchto úloh a určenie kritickej cesty vytvoriť vizualizáciu podúloh v PERT diagrame. Osobne preferujem vytvorenie Ganttovho diagramu z dôvodu väčšej prehľadnosti v tabuľke, nakoľko sa z nej dajú ľahko vyčítať činnosti prebiehajúce paralelne, trvanie činností a ich začiatok a koniec.

Záver

Projektový manažment je vynikajúci nástroj, ktorý prinúti ľudí postupovať systematicky aj na pôde, ktorá je cudzia. V mnohých prípadoch sa počas projektu študenti naučia väčšej samostatnosti a zodpovednosti.

V práci bolo popísané, prečo plánovanie využívať a aký je jeho priebeh. Problémy projektov spojené s riadením a plánovaním, ktoré sa objavili v minulosti čiastočne

pretrvávajú dodnes. Ich výskyt podnietil výskum programovacích praktík a metodológií, vďaka čomu je možné efektívnejšie využiť zdroje a znížiť náklady.

Príčiny, ktoré môžu viesť ku zlyhaniu projektu je možné identifikovať na začiatku projektu a ak s nimi počítame pri plánovaní, riziko zlyhania je možné značne znížiť. Naopak úspešné projekty majú viacero spoločných znakov, ktoré boli v eseji popísané a je dôležité dodržiavať určité zásady pri práci na projekte, ktoré prispievajú ku zlepšeniu kvality.

Ďalšou témou bol životný cyklus projektu, ktorý je pre každý projekt značne špecifický. V práci som sa zamyslel nad jednotlivými modelmi a popísal, kedy je a nie je vhodné daný model aplikovať.

Tak ako v podnikovej sfére je potrebné určovať priority a schvaľovať procesy projektu, tak aj v menších tímoch je potrebné tieto aktivity dodržiavať a vytvárať si plán pre optimálne využitie všetkých zdrojov. Na vizualizáciu plánov sa najčastejšie používa PERT a Ganttov diagram, pričom každý z nich má určité výhody.

Použitá literatúra

1. Amponsah, K.: Patterns of Communication and the Implications for Learning among Two Distributed-Education Student Teams. ACM Press, New York (2003), 20 – 27.
2. Golian, Ivan. 1998. Životný cyklus projektov: Je každý "happy"? http://www.ibispartner.sk/index.php?option=content&task=view&id=189
3. Hayles, A.R.: IT Quality enhancement and process improvement. IEEE Educational Activities Department, Vol. 9, No. 5 (2007) 13-14.
4. Internet Center for Management and Business Administration: PERT http://www.netmba.com/operations/project/pert/ (cit. 2008-10-15)
5. Kioskea - Computing community: GANTT diagram http://en.kioskea.net/projet/gantt.php3
6. Křena B., Kočí R.: Úvod do softwarového inženýrství. Brno: FIT VUT, 2006. 15-16 s. Studijní opora
7. Staníček, Z., Hajkr, J.: Řízení projektů zavádění IS do organizací. Tutorial Datakon 2005, elektronická správa http://www2.fiit.stuba.sk/~bielik/courses/msi-slov/reporty/datakon2005tutorial.pdf (cit. 2008-10-13)
8. Stellman, A.: Late projects, man-months and the software crisis. http://www.stellman-greene.com/2007/05/15/late-projects-man-months-and-thesoftware-crisis (cit. 2008-10-16)
9. The Standish Group Report: Chaos Report, elektronická správa http://www.projectsmart.co.uk/docs/chaos-report.pdf (cit. 2008-10-14)

Annotation

Planning problems and issues

Planning results in an efficient completion of many activities, whether in a daily life, business, or software project creation. In comparison to a rush progression with no efficient strategy, the exact analysis enables a huge asset. This paper deals with most common problems which arise during planning, barriers which have to be overcome by the means of plans. It describes possible approaches to the project planning and which aspects share to the projects' bad success. Business and IT are directly connected. Therefore, it is necessary to plan business resources so that they were not endangered by dissipation before the project is even finishes. Customer orientation may lead to slowdown or even stagnancy of other projects. Beside the team project, students solve other projects within classes and therefore, it is necessary to prioritize and set the time schedule, the more the team work efficiency may be influenced. Therefore, planning in this area is very dynamic.