

# Ako optimálne plánovať pri variabilných časových možnostiach?

PAVOL FÜLÖP

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

**Abstrakt.** Efektívne, časovo optimalizované plánovanie je kľúčovým atribútom úspešne zvládnutého tímového projektu. Špecifickou vlastnosťou tohto typu projektu sú však nerovnaké a nehomogénne časové možnosti členov tímu – študentov. V porovnaní s relatívne paušálnymi „časovými balíkmi“, s ktorými môže softvérová firma kalkulovať pri tvorbe krátkodobých aj dlhodobých plánov, je čas, ktorý má k dispozícii študent na školskú prácu, „roztrúsený“ medzi jednotlivé predmety štúdia. Aký model tvorby plánov zvoliť tak, aby v čo najväčšej možnej miere zohľadňoval tieto dynamicky sa meniace časové možnosti členov tímu? Táto esej bude pojednávať o plánovaní projektu s využitím smerníc a techník procesu „Team Software Process“, pričom bude diskutovať vlastnosti takéhoto systému plánovania vo vyššie uvedenom kontexte.

## Úvod - prečo je vlastne potrebné plánovať?

Asi každému, kto mal niekedy do činenia so softvérovým projektom o veľkosti programového kódu aspoň 10000 riadkov, musí byť jasné, že ak sa pustí do takéhoto projektu nerozmyslene, živelne, pričom odpovede na otázky typu „Čo a ako ďalej?“ pozná len na niekoľko hodín, prípadne dní dopredu, projekt sa nakoniec môže skončiť oveľa neskôr ako sa pôvodne predpokladalo, pričom bude vykazovať mizernú kvalitu a veľmi malú alebo žiadnu perspektívu – vlastnosti, ako udržovateľnosť, či rozširovateľnosť softvéru budú na minimálnej úrovni, a samozrejme bude aj výrazne predražený. Najpravdepodobnejšie sa však tento projekt vzhľadom na priebežne vysoko prekročený časový a finančný rozpočet ukončí predčasným krachom. Ak však na projekte pracuje celý tím ľudí, dopady intuitívneho prístupu k plánovaniu na výsledok ich spoločného úsilia sú vždy katastrofické. Systematické plánovanie softvérového projektu je teda nevyhnutnou, aj keď nie postačujúcou podmienkou jeho úspešnej realizácie.

Táto esej sa bude v podstatnej miere zaoberať problémom plánovania pre tímové softvérové projekty odohrávajúce sa v rámci jedného univerzitného kurzu. Aktérmi projektu sú študenti, ktorí ho realizujú a vyučujúci, ktorý simuluje rolu zákazníka, okrem toho však tiež konzultuje s tímom všetky možné problémy spojené s realizáciou

*Manažment v softvérovom inžinierstve, október 2006, s. 1-9*

projektu – od systému riadenia tímu až po rôzne technologické problémy. Projekt je riešený v závislosti od konkrétnej školy 12 až 24 nominálnych semestrálnych týždňov. Cieľom eseje je analyzovať aspekt variabilných časových možností členov tímu pri plánovaní tohto projektu. Pod variabilnými časovými možnosťami sa myslia nerovnaké a nehomogénne množstvá času, ktorí si študenti dokážu vyhradiť pre prácu na tomto projekte.

### **Plánovanie softvérového projektu ako súčasť manažmentu v softvérovom inžinierstve**

Petr Paleta v [6] vysvetľuje, že v terminológii, ktorú zvyčajne používa odborná literatúra, má softvérový projekt tri dimenzie: **kvalitu produktu, čas dokončenia a vynaložené náklady**. Tieto tri veličiny alebo ohraničenia sú navzájom úzko prepojené, žiadnu z nich nemôžeme zmeniť bez toho, aby sa to prejavilo na niektorej z ostávajúcich dvoch. Ak je napríklad z nejakého dôvodu nutné skrátiť termín dodávky produktu, je potrebné buď zredukovať niektoré z jeho pôvodne plánovaných rysov alebo vložiť doňho viac úsilia, čo sa nevyhnutne odzrkadlí na finančnom rozpočte projektu. Veľmi častým prípadom bývajú obidva ústupky súčasne. Vypracovanie kvalitného projektového plánu, jeho dôsledné dodržiavanie, monitorovanie a aktualizácia vytvárajú ideálny predpoklad pre to, aby sa projekt v trojdimenzionálnom priestore ohraničení čo najviac priblížil k svojmu optimu.

Čo si teda máme predstaviť pod pojmom „plánovanie softvérového projektu“? Podľa [1], plánovaním v softvérovom projekte nazývame súbor činností manažérskeho charakteru, ktoré pripraví úspešnú realizáciu projektu. Tento iteratívny proces zahŕňa aktivity ako plánovanie procesu životného cyklu vytváraného softvéru, určenie výsledných produktov projektu, odhad potrebného úsilia, odhad finančných a časových nákladov, alokáciu zdrojov, manažment rizík, kvality a manažment samotného plánovania.

Proces plánovania vychádza z informácií získaných na základe predchádzajúcej špecifikácie požiadaviek a štúdie vhodnosti. Následne sa na základe charakteru projektu - jeho funkcionálna, či technická zložitosť, stupeň inovácie, či požiadavky na kvalitu - vyhodnotia možné aplikovateľné procesy životného cyklu softvéru a vyberie sa najvhodnejší z nich. Projekt sa potom hierarchicky dekomponuje na menšie úlohy, pričom sa jasne definujú výstupy každej úlohy tak, aby boli splnené požiadavky špecifikácie. Následne sa vykoná detailný odhad vynaloženého úsilia, a nákladov, určí sa časový harmonogram projektu. Na základe tohto harmonogramu sa pre každú úlohu priradia zdroje tak, aby sa optimalizovala produktivita na individuálnej, tímovej, aj organizačnej úrovni. Detailne sa analyzujú riziká a určia potrebné procesy riadenia kvality. Ako bolo vyššie uvedené, proces plánovania má iteratívny charakter, je teda nevyhnutné, aby sa určili takisto procesy riadenia tohto procesu, jeho monitorovanie a pravidelné revízie [1].

Opísaný proces plánovania softvérového projektu je zrejme dostatočne všeobecný na to, aby sa mohol použiť ako šablóna pri riešení prakticky ľubovoľného softvérového

projektu. Zároveň je však príliš všeobecný na to, aby sa mohol priamo uplatniť. Akým konkrétnym spôsobom je možné aplikovať ho pre školský tímový softvérový projekt?

### **Tímový softvérový projekt – malý projekt, malý tím**

Akademicko-pedagogický rámec školského tímového softvérového projektu dáva tomuto typu projektu osobitný charakter, nie príliš vídaný v priemyselnej praxi. Hoci býva spravidla veľmi dobrou, ak nie optimálnou prípravou na „skutočný svet“, v zásade určuje niekoľko viac, či menej podstatných rozdielov, ktorými sa spôsob realizácie tohto projektu a jeho ohraničenia líšia od reálnych projektov, s ktorými sa študenti neskôr vo svojej profesii stretávajú.

#### **Problém distribúcie časových zdrojov študenta**

Jedným z týchto rozdielov je aj problém distribúcie časových zdrojov študenta, čo je v porovnaní s reálnou praxou veľmi špecifický jav. Klasická softvérová firma má k dispozícii pevné množstvo človekohodín týždenne, s ktorým môže viac-menej bezpečne rátať (ak si odmyslíme dovolenky pracovníkov, ktoré sú však tiež zvyčajne plánované; choroby, prípadne úrazy alebo náhly odchod pracovníka do inej firmy) pri tvorbe projektových plánov. Množstvo času, ktoré majú jednotliví študenti k dispozícii pre tímový softvérový projekt, je však nerovnaké v závislosti od schopností konkrétneho študenta a aj množstva a náročnosti úloh, ktoré si vyžadujú ostatné predmety zapísané študentom v danom semestri – nie všetci členovia tímu majú totiž zapísané tie isté predmety. Tiež je potrebné zdôrazniť, že rozloženie tohto „časového balíka“ medzi jednotlivé predmety býva v priebehu semestra pomerne nehomogénne, keďže ťažisko pozornosti a úsilia študenta sa presúva od jedného predmetu k druhému podľa toho, ako si to jednotlivé predmety počas semestra vyžadujú, pričom často dochádza k „nárazovým odberom“ z tohto balíka.

#### **Efekt zosilnenej nevýhody**

Neodškriepiteľným faktom reality vývoja softvéru je existencia niekedy veľmi výrazných rozdielov v produktivite práce medzi programátormi, s čím musí nakoniec počítať každá seriózna softvérová spoločnosť. Táto skutočnosť je v prípade študenta – riešiteľa tímového projektu ešte umocnená tým, že si svoje sily musí rozložiť do viacerých predmetov štúdia. Produktívnejší študent má v porovnaní s menej produktívnym kolegom teda dvojnásobnú výhodu - nielenže za každú jednotku času, ktorú si dokáže alokovať pre riešenie tímového softvérového projektu, urobí toho viac, ale spravidla má tohto času aj viac k dispozícii, keďže rýchlejšie zvláda aj požiadavky iných predmetov jeho štúdia.

#### **“Tvrdé“ termíny**

Priemyselná kultúra podporuje zadávanie tak „mäkkých“ ako aj „tvrdých“ termínov pre daný projekt. V krajných prípadoch sa môžu posunúť dokonca aj „tvrdé“ termíny.

Naproti tomu povaha akademickej kultúry umožňuje zadávať iba „tvrdé“ termíny pre ukončenie projektu, či dosiahnutie niektorého z určených míľnikov. Dôvod je zrejmy – Predmet musí byť vždy absolvovaný v rámci daného semestra alebo dvoch, podľa jeho rozsahu [2].

Dobré plánovanie v školskom tímovom softvérovom projekte by malo vo významnej miere brať do úvahy aj tieto vlastnosti prostredia, v ktorom je tímový projekt realizovaný. Je síce pravdou, že zodpovedným prístupom k študijným povinnostiam je možné do určitej miery zkorigovať vplyv týchto špecifik na priebeh projektu (najmä vyššie spomenuté „nárazové hrany“), z principiálneho hľadiska však tento vplyv nie je možné vylúčiť.

### **Proces „The Team Software Process“**

“The Team Software Process” (TSP) je úplne definovaný a merateľný proces, ktorý môžu tímy použiť pre plánovanie práce, vykonávanie týchto plánov a plynulú zdokonaľovanie svojich procesov vývoja softvéru [5]. Už prvé skúsenosti s týmto procesom ukázali, že jeho použitie zlepšuje kvalitu a produktivitu inžinierskych tímov, zároveň im pomáha presnejšie spĺňať záväzky týkajúce sa nákladov a časových plánov [4].

Autorom TSP je Watts Humphrey, bývalý viceprezident IBM [8], svetová autorita pre kvalitu softvéru a zlepšovanie softvérových procesov. V roku 1996 vyvinul prvú verziu TSP procesu. Pôvodný proces TSP0 bol navrhnutý čo najjednoduchšie, ako to len bolo možné. Humphrey ho potom vyskúšal na dvoch tímoch, vyhodnotil výsledky a následne vylepšil jeho smernice. Počas nasledujúcich troch rokov vytvoril a testoval postupne deväť ďalších verzií. Po tom, ako bolo jasné, že proces spĺňa svoj účel, snažil sa ho opäť zjednodušať a zefektívňovať. Výsledok tejto snahy je, že najnovšie verzie TSP sú rozsahom podstatne menšie ako tie pôvodné, pričom proces nestratil nič na svojej užitočnosti. [4]

### **Princípy a účel TSP**

Hlavnou myšlienkou TSP je, že inžiniersky tím je schopný riadiť svoju prácu sám efektívnejšie a kvalitnejšie ako keby túto prácu plánoval manažment zhora. Podľa [3] však tím musí byť dostatočne disciplinovaný, zároveň musí rozumieť celkovému kontextu projektu, takže môže

- porozumieť obchodným cieľom a cieľom samotného produktu
- vyprodukovať vlastný plán pre splnenie týchto cieľov
- vytvárať si vlastné záväzky
- riadiť si svoj projekt
- konzistentne používať metódy a procesy, ktoré si vyberie
- riadiť kvalitu produktu

Úspešné samoriadiace sa tímy si vyžadujú schopných a vhodných členov. Tento fakt je kritický, pretože v konečnom dôsledku každý jeden programový riadok

softvérového modulu (okrem automaticky generovaného kódu napr. vývojovými prostrediami) pochádza od konkrétneho softvérového inžiniera. Jeho schopnosti, disciplína a záväzky určujú kvalitu tohto modulu a aj rýchlosť, s akou bol modul vytvorený. Softvérový produkt je zase kompozíciou jeho jednotlivých modulov. Tieto moduly sú navrhnuté, vytvorené, integrované, testované a udržiavané tímom softvérových inžinierov, ktorých schopnosti, disciplína a záväzky určujú úspech celého projektu [3].

### „The Personal Software Process“

Aby mohol softvérový inžinier v tíme vytvárať optimálne plány pomocou TSP, musí najskôr vedieť napláňovať svoju vlastnú prácu. Aby mohol plánovať svoju prácu, musí vedieť dobre odhadnúť jej rozsah, tiež však musí poznať svoje schopnosti. Poznanie svojich schopností znamená poznať mieru svojej produktivity a kvalitu práce a vedieť ju vyjadriť pomocou nejakých konkrétnych metrík..

V kurzoch PSP softvéroví inžinieri získavajú znalosti o tom, ako produkovať plány postavené na svojich vlastných historických dátach. Učia sa koncepty odhadovania rozsahu, úsilia a času, a produkcie plánov založených na princípoch „plánovanej hodnoty“ a „získanej hodnoty“ (spotreba plánovaného rozpočtu v závislosti na čase určuje „plánovanú hodnotu“ pre daný časový okamih projektového plánu; „získaná hodnota“ je časť rozpočtu, ktorá podľa plánu zodpovedá ukončeným úlohám do tohto okamihu) [5]. Tréning PSP pozostáva z riešenia niekoľkých postupne komplikovanejších softvérových problémov. Základný proces riešenia problému pomocou PSP sa skladá z troch fáz (podľa [7]):

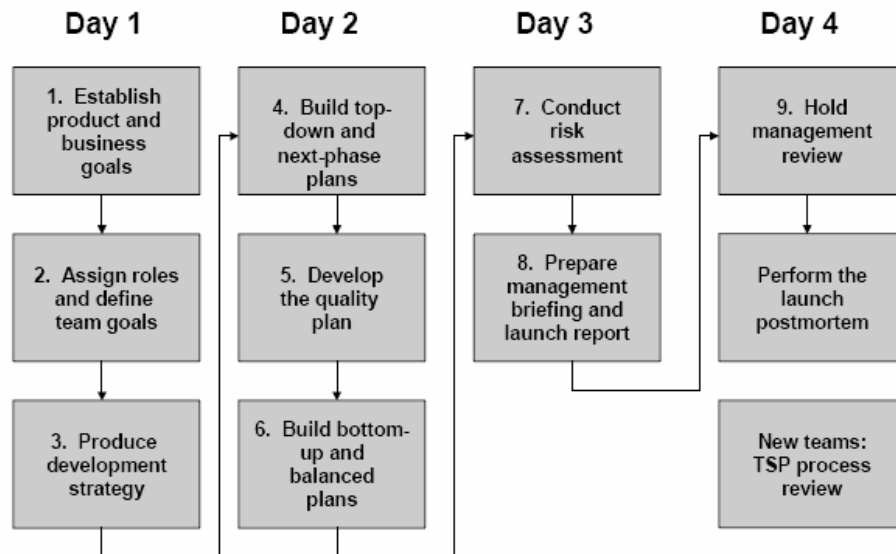
1. Produkcia plánu pre realizáciu riešenia
2. Vlastné riešenie
  - a) návrh programu
  - b) kontrola návrhu
  - c) napísanie kódu programu
  - d) kontrola kódu programu
  - e) kompilácia a odstránenie všetkých chýb
  - f) testovanie a odstránenie všetkých chýb
3. Postmortem fáza, v ktorej treba porovnať prevedenie riešenia s pôvodným plánom, zaznamenať dáta procesu, vygenerovať sumárnu správu a zadokumentovať všetky nápady, ako zdokonaľiť samotný proces.

Po absolvovaní kurzu PSP sú teda softvéroví inžinieri schopní [10]:

- riadiť kvalitu svojich projektov,
- vytvárať záväzky, ktoré sú schopní splniť,
- presne odhadovať a plánovať,
- redukovať počet chýb vo svojich produktoch.

### Plánovanie projektu pomocou TSP

Po vyformovaní tímu, pričom členovia tímu sú správne natrénovaní pomocou PSP, celý tím participuje na tzv. spúšťaní tímu. Proces spúšťania tímu je zobrazený na Obr.1. Pre každé z deviatich stretnutí existuje osobitný skript, ktorý popisuje aktivity stretnutí v dostatočných detailoch, takže trénovaný „kouč spúšťania“ môže tím sprevádzať po jednotlivých krokoch. Nasledovaním tohto procesu spúšťania sa tím dopracuje k detailnému plánu. Aby sa tím mohol stať kohéznou a efektívnou pracovnou jednotkou, všetci jeho členovia musia voči tomuto plánu cítiť záväzok, čo sa dá dosiahnuť jedine zahrnutím všetkých členov tímu do tvorby plánu, ktoré proces spúšťania tímu priamo anticipuje. [4]



Obr. 1. Proces spúšťania tímu [3]

V prvom kroku manažment projektu prezentuje tímu produktové a obchodné ciele. Na druhom stretnutí si tím rozdelí role (napríklad vedúci tímu, manažér plánovania, manažér vývoja, manažér kvality manažér podporných prostriedkov a iné), pričom každý člen tímu musí mať aspoň jednu rolu. Na treťom a štvrtom stretnutí sa určí celková stratégia projektu a jeho plán. V tomto kroku sa vyberá vhodný proces životného cyklu a alokujú sa podporné prostriedky a iné potrebné zdroje. Počas piateho stretnutia sa vytvorí plán kvality, ktorý okrem definície akcií poskytuje merateľný základ pre priebežné vyhodnocovanie kvality práce. Na šiestom stretnutí sa jednotlivé úlohy plánu rovnomerne prerozdedia medzi členov a vyrobí sa detailný plán. Po analýze rizík na siedmom stretnutí sa tím pripraví na prezentáciu plánu manažmentu, čo je obsahom stretnutia číslo osem. Na deviatom – poslednom stretnutí sa tento plán

prezentuje, ak sa však svojimi ohraničeniami nezhoduje s pôvodnými cieľmi manažmentu, je potrebné pripraviť alternatívne plány so zmenenými požiadavkami alebo pridanými ďalšími zdrojmi. [4]

Po spustení tímu je plán dôkladne monitorovaný pomocou metódy „plánovanej hodnoty“ a „získanej hodnoty“. Týždenné stretnutia tímu slúžia na prezentáciu stavu plnenia plánu jednotlivými členmi, ale aj komunikáciu všetkých členov tímu súčasne. V prípade potreby sa pristúpi k prerozdeleniu práce medzi jednotlivých členov tímu tak, aby celkový plán ostal nezmenený.

Manažment potrebuje byť pravidelne informovaný o stave prác, čo je kritickou zodpovednosťou vedúceho tímu. Ten sa musí postarať o to, aby tím vygeneroval každý týždeň správu o stave pokroku projektu.

Keď vo všeobecnosti nie je možné pre inžiniera detailne si naplánovať prácu pre viac ako 3 až 4 mesiace vopred, a samotný projekt má často omnoho väčší rozsah, je potrebné rozdeliť projekt na niekoľko fáz, pričom na začiatku každej ďalšej fázy sa zrealizuje nové spustenie tímu.

## **Aplikácia niektorých konceptov TSP do školského tímového projektu**

Východiskami pre realistické plánovanie v softvérovom projekte sú najmä znalosti

- cieľov projektu a jeho kontextu
- schopností a skúseností členov tímu
- ohraničení projektu – finančných, časových, kvalitatívnych, iných...

Každé z týchto východísk je kľúčové. Ak by napríklad tím poznal ciele aj ohraničenia projektu, ale zároveň by nemal dobrú vedomosť o svojom potenciáli z hľadiska zručností, znalostí, produktivity a kvality, ale aj schopnosti tímovej spolupráce jednotlivých členov tímu, nemôže vytvoriť funkčný plán pre realizáciu projektu – s veľkou pravdepodobnosťou by tento plán podceňoval, v zriedkavejších prípadoch preceňoval časovú náročnosť projektu. Podobne, ak si je tím vedomý všetkých termínov a aj seba samého, ťažko naplánuje niečo, o čom nemá presnú predstavu.

Podľa TSP si je každý z členov tímu dobre vedomý všetkých cieľov projektu a aj kontextu, v akom sa tento projekt nachádza (situácia na trhu), čo jasne vyplýva z definície prvých dvoch krokov procesu spúšťania tímu. Tiež na základe predchádzajúceho tréningu PSP veľmi dobre pozná samého seba a aj ostatných, ak sú tréningy podobne. Vďaka fungujúcej komunikácii s manažmentom sú tímu jasné všetky ohraničenia projektu, tím sa však nesnaží za každú cenu „natlačiť“ do týchto ohraničení, ak to nie je možné, lebo vie, že by to z hľadiska výsledku projektu nemalo žiadny zmysel. Namiesto toho generuje alternatívy plánov s dosiahnuteľnými „konfiguráciami“ ohraničení a manažment má len dve možnosti. Prijatť niektorú z alternatív, prípadne po diskusii s tímom mierne pozmeniť (ale nie sprísňovať) nové ponúkané ohraničenia, alebo...rozlúčiť sa s projektom!

Kontext školského tímového projektu je zrejmý – je potrebné absolvovať daný predmet. Menej jasné je to so znalosťou študenta seba samého. Ak by fakulta ponúkala kurz PSP ako povinný predmet štúdia, pri riešení tímového projektu by sa mohli

vytvárať oveľa agresívnejšie plány v zmysle funkcionality a kvality, ktoré by si túto agresivitu mohli dovoliť, keďže by stáli na reálnych základoch definovaných metrick.

Pevné ohraňenie v školskom tímovom projekte existuje len jedno, dodržať časový harmonogram. Finančné ohraňenie v tomto prípade (zvyčajne) neexistuje a pokiaľ zadávateľ „nebazíruje“ na dodržaní všetkých jeho predstáv do bodky, špecifikácia požadovanej funkcionality a kvality výsledku sa dá upraviť na prijateľnú mieru ako pre zadávateľa, tak aj pre realizátorov projektu.

Dĺžka trvania priemernej fázy projektu podľa TSP približne zodpovedá dĺžke akademického semestra. Ak by sa teda tímový projekt realizoval v priebehu dvoch semestrov, spúšťanie tímu by mohlo nastať vždy na začiatku semestra. Stretnutia pri spúšťaní tímu sú dosť nahustené, čo však napríklad v druhom týždni semestra nemusí predstavovať zásadný problém. Týždenné stretnutia tímu sú taktiež vyhovujúce.

Je však proces TSP vhodný aj z hľadiska témy tejto eseje? V prvom rade, jedným zo základných konceptov procesu je možnosť prerozdelenia práce medzi jednotlivých členov tímu podľa momentálnej potreby. Najvyťaženejší členovia tímu by takto mohli byť dočasne odbremenení, čím by získali viac priestoru pre zvládnutie ostatných študijných povinností. Celkové časové náklady projektu by sa dali oveľa presnejšie stanoviť vďaka lepšiemu sebapoznaniu členov tímu. No a dobrý plán vytvorený vďaka kvalitnému procesu a kvalitným ľuďom by pomohol lepšie homogenizovať distribúciu času študenta, čo by mohlo výrazne zvýšiť celkovú úroveň projektu.

Aby bol každý člen tímu krátkodobo ľahko nahraditeľný aj vzhľadom na jeho manažérsku rolu v tíme, je potrebné, ak by sa určitým spôsobom zdvojovali jednotlivé role. Okrem štandardnej role by teda člen tímu robil „pravú ruku“ alebo konzultanta členovi s inou rolou, čím by pri relatívne malom navýšení záťaže zabezpečil aspoň pomerne solídnu náhradu jeho role v prípade nutnej potreby.

## Záver

Koncept plánovania tak, ako ho prezentuje proces TSP sa javí ako veľmi dobrý podklad pre vytvorenie úspešného procesu plánovania školského tímového projektu. Nepredvídané vplyvy aspektu variabilných časových možností študenta na priebeh projektu a jeho výsledok by mohli byť za pomoci tohto procesu výrazne zredukované. Kurzy tímových softvérových projektov by sa takto mohli posunúť o ďalší krok bližšie k realite priemyselnej praxe.

## Použitá literatúra

1. Bourque, P. et al.: *Guide to the Software Engineering Body of Knowledge*, 2004 version, IEEE Computer Society, 2005
2. Conn, R.: A Reusable, Academic-Strength, Metrics-Based Software Engineering Process for Capstone Courses and Projects, In: *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, 492-496, 2004.



3. Davis, N., Mullaney, J.: The Team Software Process in Practice: A Summary of Recent Results, *Technical Report CMU/SEI-2003-TR-14*, SEI, CMU, 2003.
4. Humphrey, W.S.: The Team Software Process, *Technical Report CMU/SEI-2000-TR-023*, SEI, CMU, 2000.
5. McAndrews, D. R.: The Team Software Process: An Overview and Preliminary results of using disciplined practices, *Technical Report CMU/SEI-2000-TR-15*, SEI, CMU, 2000.
6. Paleta, P.: *Co programátory v škole neučí aneb Softwarové inženýrství v reálné praxi*, Vydání první, Brno, Computer Press, 2003. 337s.
7. Pomeroy-Huff, M. et al.: *The Personal Software Process Body of Knowledge*, version 1.0, *Special Report CMU/SEI-2005-SR-003*, SEI, CMU, 2005.
8. *Watts Humphrey – Wikipedia, the Free Encyclopedia* [on-line]. Wikipedia, the free encyclopedia, [cit. 2006-10-24]  
URL: < [http://en.wikipedia.org/wiki/Watts\\_Humphrey](http://en.wikipedia.org/wiki/Watts_Humphrey) >
9. *Biography - Watts S. Humphrey* [on-line]. Carnegie-Mellon Software Engineering Institute, [cit. 2006-10-24]  
URL: < <http://www.sei.cmu.edu/tsp/watts-bio.html> >
10. *Personal Software Process* [on-line]. Carnegie-Mellon Software Engineering Institute, [cit. 2006-10-24]  
URL: < <http://www.sei.cmu.edu/tsp/psp.html> >

## Annotation

### *How to schedule optimally with variable time resources?*

Effective, time-optimized scheduling is a key attribute of a successfully accomplished team project. A specific attribute of this kind of project is that team members – students have unequal and inhomogeneous individual time resources they can use for the work on project. In comparison with relatively flat „time packages“ that a software company can calculate with while making both short-time and long-time schedules, the time available for the student’s schoolwork is “scattered” throughout individual courses. Which planning model should be used so that it takes into consideration these constantly changing time resources of the team members? This essay deals with project planning by using guidelines and practices of the process „Team Software Process“, mainly discussing properties of the planning systems in the context above.