

# Metódy a postupy plánovania v softvérovom inžinierstve

PETER SMOLINSKÝ

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

**Abstrakt.** V softvérových projektoch, rovnako ako v projektoch z iných oblasti ľudskej činnosti, sa už dlhodobo používajú plány. Pri softvéri špeciálne hovoríme o vývoji riadenom plánmi. Je však použitie takéhoto prístupu aj v tomto prípade vhodné? Vytvorenie presného plánu, ktorý zohľadňuje aj potenciálne zmeny, je náročné. Obzvlášť v prípade softvéru, ktorý je do veľkej miery nevizualizovateľný. Pre riešenie problémov so správnym odhadnutím dĺžky trvania projektu ako aj pre samotné vytvorenie plánu pozostávajúceho z jednotlivých vhodne dekomponovaných úloh sa používajú rôzne metódy. V tejto eseji sa s nimi bližšie oboznámime a zhodnotíme ich silné a slabé stránky. Navzájom vybrané metódy porovnáme a pokúsime sa nájsť tú vhodnú pre tvorbu plánov v softvérovom inžinierstve.

## Úvod

Ľudia už od pradávna rozmýšľajú dopredu nad niečím skôr ako to urobia. Toto im umožňuje lepšie sa pripraviť na danú činnosť a odhadnúť okolnosti, ktoré by ich mohli pri samotnej činnosti nemilo prekvapiť. Čím rozsahovo väčšie sú dané činnosti, tým sa ľudia začínajú viac zamýšľať nad nimi ešte pred ich vykonávaním. Snažia sa čím detailnejšie a presnejšie odhadnúť priebeh danej činnosti, tvoria plán. Systematický prístup k tvorbe plánov môžeme nazvať plánovaním. Plánovanie a plány majú svoje obrovské opodstatnenie aj v softvérovom inžinierstve, kde navyše je ťažké robiť presné odhady. V tomto odvetví vytvárame plány hlavne vzhľadom na jednotlivé projekty. Podstatné je však samotný plán vôbec vytvárať. Aj keď nebude presne splnený, ale výrazne sa tým zvýši úspešnosť projektu.

## Metódy plánovania

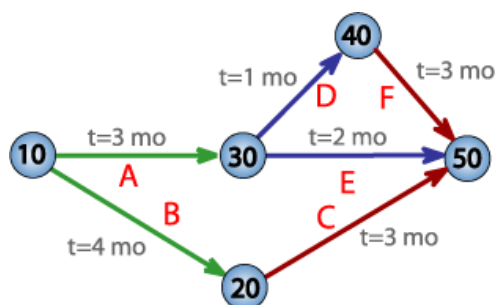
Keďže plánovanie nie je žiadnou novinkou, existujú už rôzne zaužívané metódy a postupy tvorby plánov. To, že plánovanie je skutočne dôležité, dokazujú aj štatistické výsledky, podľa ktorých zlé plány sú jedným z najväčších problémov dnešných manažérov. Chybou v týchto plánoch sú hlavne nereálne termíny. K neplneniu termínov vedú jednak požiadavky zákazníkov, ktorý samozrejme chcú čím skôr vidieť výsledok, ale aj nesprávne odhady a nepresnosti plánov. Práve preto sa pozrime aké možnosti riešenia týchto nepresností nám vybrané metódy plánovania ponúkajú.

## PERT

Program Evaluation and Review Technique skrátene PERT je plánovacia metóda na analyzovanie jednotlivých úloh a ich usporiadanie do orientovaného grafu. V grafe sú zachytené závislosti medzi jednotlivými úlohami a časy potrebné na ich vykonanie. Metóda si vie poradiť aj v prípade ak nevieme presne odhadnúť potrebný čas, ale vieme povedať jeho minimálnu a maximálnu hodnotu.

Od iných techník sa hlavne odlišuje prístupom orientovaným na udalosti, nie na začiatok a koniec činnosti. Jeho výhodou je jednoduché rozšírenie v neskorších etapách projektu. Nie je problém pridať ďalšie míľniky a činnosti do grafu. Týmto PERT rieši nejasnosti a nepresnosti dané na začiatku projektu.

Ukážka diagramu je na Obr. 1. Vrcholy grafu predstavujú tzv. míľniky alebo tiež nazývane udalosti. Hrany sú jednotlivé aktivity. Míľnikom je bod, ktorý označuje ukončenie resp. začiatok jednej alebo viac aktivít. Míľnik nemôže byť dosiahnutý pokiaľ nie sú splnené všetky aktivity vstupujúce do míľnika. Míľniky nijak nespotrebúvajú čas ani zdroje, skorej iba vyjadrujú dosiahnutie určitého stavu projektu. Aktivita predstavuje činnosť, ktorá naopak spotrebúva čas aj zdroje (ľudská práca, suroviny, stroje). Dá sa opísať ako úsilie dostať sa z jedného stavu projektu, míľnika, do druhého, nasledujúceho. Míľniky sú v diagrame označované násobkami čísla 10, čo umožňuje do budúcnosti v neskorších fázach projektu v prípade identifikovania nových míľnikov a potrebných činnosti ich jednoduché pridanie. Samotné aktivity sú označované písmenami a ohodnotené potrebným časom. V diagrame sa môžu vyskytovať aj aktivity s nulovou dĺžkou trvania. Nejedná sa o skutočné činnosti, ktoré by mal tím vykonať. Používajú sa iba na vyjadrenie vzťahu a závislosti medzi míľníkmi, ktoré nie je možné spojiť do jedného, lebo väčšinou logicky spolu nesúvisia a vyjadrujú iný stav projektu.



Obr. 1. Ukážka PERT diagramu

Pri výpočte doby trvania aktivity PERT zohľadňuje:

- *Optimistický čas (O)*: minimálna možná doba potrebná na splnenie danej úlohy, predpokladá sa, že všetko pôjde lepšie ako očakávame.
- *Pesimistický čas (P)*: maximálna doba potrebná na splnenie danej úlohy, predpokladá sa, že všetko pôjde oveľa horšie ako očakávame (ale nie katastroficky).
- *Najpravdepodobnejší čas (M)*: najlepší odhad času potrebného na danú úlohu, predpokladá sa, že všetko pôjde ako má podľa plánu.
- *Očakávaný čas (TE)*: najlepší odhad času potrebného na splnenie úlohy, tiež sa predpokladá, že všetko pôjde ako má podľa plánu, vyjadruje však skôr strednú hodnotu času potrebného na danú úlohu, ktorá je vypočítaná z predchádzajúcich hodnôt podľa vzorca (1), pri opakovaní úlohy by mal byť spotrebovaný rovnaký čas.

$$TE = (O + 4M + P) / 6 \quad (1)$$

PERT je hlavne určený na plánovanie veľkých komplexných projektov. Myslím si však, že je plne použiteľný aj na projekty menšieho stredného rozsahu, kde poslúži hlavne na určenie závislosti medzi jednotlivými aktivitami a následne nájdenie kritickej cesty.

## Kritická cesta

S PERT je veľmi úzko spätá metóda kritickej cesty. Kritickú cestu môžeme definovať ako postupnosť činností v projekte, pričom súčet dôb ich trvania sa rovná dobe celého projektu. Z pohľadu grafu je to cesta s maximálnou dobou trvania. O činnostiach na kritickej ceste hovoríme ako o kritických. Keď sa niektorá oneskorí, oneskorí sa aj celý projekt. PERT nám veľmi jednoducho podľa teórie grafov umožňuje nájsť kritickú cestu.

Kritická cesta slúži na ohodnotenie úzkeho hrdla daného projektu. Vďaka tomu nachádza veľké uplatnenie nielen pri manažovaní projektov, ale aj pri softvéri samotnom. Napríklad môžeme hovoriť o kritickej ceste v TPC/IP sieťovom modeli alebo pri paralelnom spracovaní nejakých dát. Osobne s použitím kritickej cesty pri softvérovom projekte zatiaľ nemám skúsenosti, avšak danú metódu hodnotím veľmi pozitívne. Hlavnú výhodu vidím v odhadnutí doby ukončenia projektu, čo pomaly vždy je dôležitý údaj aj pre samotného zákazníka. Nevýhodou môže byť nepresnosť daného odhadu, ktorá sa však zvýši použitím v kombinácii napr. s metódou PlanningLines [1].

## Ganttova schéma

Ganttová schéma je druh diagramu na zachytenie rozvrhu projektu. Používa notáciu vodorovných pásov, úsekov reprezentujúcich určitú činnosť zakreslenú na horizontálnej časovej osi. Jednoducho môžeme vyčítať začiatok a koniec činností, ich závislosti a paralelné vykonávanie. V nástrojoch určených na manažovanie projektu sa často do diagramu zakresľuje aj vertikálna čiara reprezentujúca aktuálny dátum a čas. Tým jednoducho zistíme predpokladaný stav projektu a aktuálne prebiehajúce činnosti.

Základným predpokladom pre tvorbu Ganttovho diagramu je správne vytvorený rozpis práce. Jeho tvorba by mala predchádzať samotná tvorba Ganttovho diagramu. Toto býva častou chybou, ktorej sa manažéri dopúšťajú. Dôvodom je pri súbežnej tvorbe rozpisu práce (dekomponovaní úloh na menšie) a Ganttovho diagramu porušenie tzv. pravidla 100%. Spomínané pravidlo určuje, že po dekomponovaní práce na určitú úroveň, musia všetky vytvorené elementy spoločne tvoriť 100% rodičovského elementu [3].

„Ganttová schéma uvažuje iba fixné dĺžky trvania úloh. Preto vzbudzuje dojem presnej vedomosti o danom probléme, akoby máme k dispozícii presný čas začiatku a ukončenia činnosti. Toto môže jednoducho viesť manažéra k zanedbaniu dočasných nepresností plánu aj keď stupeň nepresnosti sa môže veľmi líšiť podľa druhu projektu“ [2].

Z hľadiska obmedzení projektu sa Ganttová schéma zameriava iba na rozvrh a jeho manažovanie. Neuvažuje rozmery projektu z pohľadu množstva zdrojov a splnenia požiadaviek. Preto podľa Ganttovej schémy nie je vhodné projekty porovnávať.

Výhodou je jednoduchosť na pochopenie a z toho vyplývajúca prístupnosť širším masám.

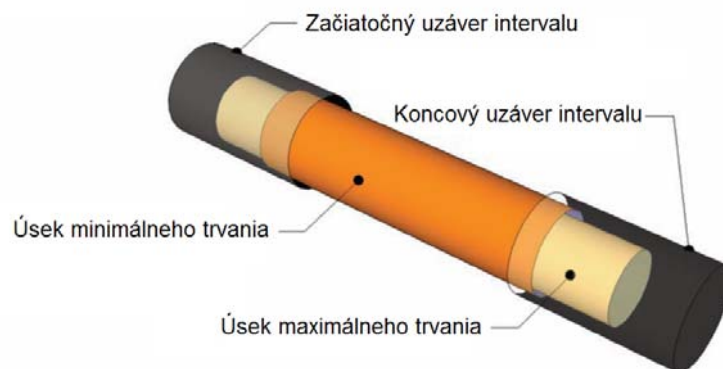
## PlanningLines

Metóda PlanningLines používa piktogramy, ktoré predstavujú komplexné neurčité úlohy. Ukážku spomínaného piktogramu môžeme vidieť na Obr. 2. Tie sa zakresľujú do hierarchickej štruktúry predstavujúcej plán projektu tak, aby bol ľahko a intuitívne

pochopiteľný. Táto štruktúra sa podľa môjho názoru až príliš veľmi podobá na Ganttov diagram. Stačí porovnať s príkladom na Obr. 3.

Piktogram sa skladá z uzavretých úsekov. Jeden reprezentuje maximálnu dĺžku trvania činnosti a druhý minimálnu. Úseky sú ukončené uzávermi, ktoré predstavujú začiatok a koniec činnosti. Takže všetky atribúty piktogramu sú: [1]

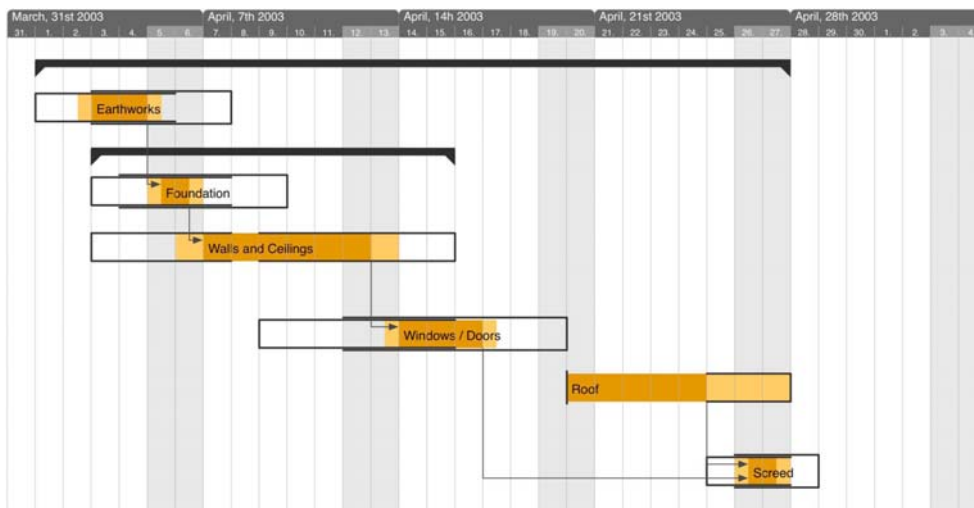
- začiatok intervalu
- najskorší čas začiatku činnosti
- najneskorší čas začiatku činnosti
- koniec intervalu
- najskorší čas ukončenia činnosti
- najneskorší čas ukončenia činnosti
- doba trvania činnosti
- minimálna doba trvania činnosti
- maximálna doba trvania činnosti



Obr. 2. Model PlanningLines

## Simulácia pri tvorbe plánov

„Najviac zrejme je použitie simulácie v manažmente projektov na simulovanie plánu, aby nám to umožnilo vidieť ako nejasnosti a nepresnosti v dĺžkach trvania úloh ovplyvnia výsledok. Nezáleží na tom ako podrobne simulujeme rôzne elementy týkajúce sa plánu, simulácia bude nereálna, ak je plán statický. Preto manažovanie projektu musí byť veľmi dynamické, keďže samotný projekt je dynamický“ [4].



**Obr. 3.** Príklad jednoduchého plánu projektu

Simulácia svoje opodstatnenie nadobúda hlavne až pri väčších projektoch. Moderným prístupom je v tejto oblasti použitie umelej (počítačovej) inteligencie, kedy vykonávame simuláciu pomocou agentov. V prípade stredných a menších softvérových projektov sa simulácia však veľmi nepoužíva. Myslím si, že pokiaľ by sme vedeli presnejšie určiť a matematicky opísať priebeh softvérového projektu, určite by si aj v tejto oblasti simulácia našla oveľa väčšie uplatnenie pri plánovaní.

## Porovnanie metód

V článku [2] autori porovnávajú metódy PERT a PlanningLines. Zisťujú, ktorá z týchto metód je lepšia z hľadiska jednoduchosti reprezentácie plánu, rýchlosti a chybovosti čítania daného plánu manažérmi a identifikácie neurčitosti v dĺžke trvania jednotlivých aktivít.

Výsledné porovnanie ukazuje prevahu metódy PlanningLines nad PERT, kde PlanningLines dosahujú rovnaké výsledky čo sa týka času, ale manažéri pri práci s nimi spravia menej chýb. Výsledok je dosť logický, keďže PERT nemá exaktný spôsob vyjadrenia nepresností plánov.

Myslím si, že aj tak je vhodné používať obidve metódy resp. ich kombináciu. Ako to býva vo väčšine prípadov, aj teraz je pravda (a aj správne riešenie) niekde v strede. Optimálne je použiť kombináciu spomínaných metód, prípadne ich použiť viacej a priamo sa presvedčiť, ako ich dokážeme efektívne využiť a na aký druh projektov sú vhodné. Napríklad Ganttová schéma sa dá veľmi jednoducho rozšíriť o prvky PlanningLines, keď následne dokáže oveľa lepšie počítať s variabilnejšou dĺžkou časových intervalov. Rovnako aj záver autorov v spomínanej práci je: „Keďže dané

techniky zdieľajú spoločné prvky na opis úloh v projekte, je možné skombinovať ich silné stránky. Metóda PERT môže byť použitá na vstup a zmenu atribútov samotných úloh, pokiaľ PlanningLines zabezpečujú pohľad na úlohy a analýzu možných rizík v pláne“ [2].

## Moje skúsenosti s plánovaním v podnikoch

Osobne som sa s tvorbou plánov a manažovaním projektu pomocou spomínaných pokročilých metód bohužiaľ nestretol. V podnikoch, kde som pracoval si moji nadriadení manažéri vystačili iba s primitívnym vykazovaním, meraním a určovaním človeko-dni bez ohľadu na špecifiká vývoja softvéru a hlavne nejasnosti daného projektu. Plánovanie bolo výrazne podcenené a neprístupovalo sa k nemu na primeranej profesionálnej úrovni. Jednoducho manažér rozdelil úlohy, čo nebolo náročné vzhľadom na zvyčajne malý počet členov tímu a potom sa pýtal na ich odhad dĺžky práce na daných úlohách. Sám výsledné čísla trochu skorigoval podľa osobných skúsenosti s jednotlivými zamestnancami a požiadaviek projektu, prípadne pridal časové zdroje potrebné pre koordináciu a komunikáciu tímu. Vytvorený plán bol odovzdaný nadriadeným. Týmto som sa mohol aj ja podieľať na pláne, keď som mal odhadovať, koľko mi daná práca bude trvať. Takéto ohodnotenie v prípade softvéru je podľa mňa však veľmi ťažké a nepresné. Rozhodnutie, že koľko času treba danému zamestnancovi pre danú úlohu má vykonávať expert v danej oblasti alebo samotný manažér. Ten ma posúdiť úlohu a jej časové trvanie upraviť podľa kvalít zamestnanca.

Tiež sa mi veľmi nepáčilo, keď v takýchto plánoch môj nadriadený manažér nezohľadňoval pri použití nových technológií dobu potrebnú na ich naučenie sa. Myslím si, že v praxi málokedy pri projekte sa vychádza a použijú sa počas celého trvania projektu iba technológie a postupy, ktorým rozumie každý člen tímu. Preto je podľa môjho názoru dôležité zohľadňovať aj tento aspekt a počítať s ním aj v pláne projektu. V malých podnikoch toto väčšina manažérov podceňuje. Zdá sa mi, že pomaly dnes platí, že čím väčšia firma, tým viac dbá na vzdelávanie sa jej zamestnancov. Tie menšie akoby zabúdali, že nestačí iba okamžite produkovať a mať vysokú efektívnosť. Časom prídu novšie postupy a technológie, ktoré môžu byť rádovo lepšie a konkurencia ich nebude váhať použiť.

## Plánovanie – pre mnohých nič nové

V dnešnej dobe prichádzajú mnohé zahraničné spoločnosti hlavne zo západnej Európy a USA a prinášajú svoje know-how, nové postupy a metodiky v daných oblastiach. Rovnako sa to týka aj metód plánovania. Myslia si, že prišli do zaostalej, Bohom zabudnutej krajiny, kde prinášajú pokrokové prevratné technológie a postupy, o ktorých my nemáme ani poňatia. Zabúdajú však, že aj napriek predchádzajúcemu dlhému obdobiu nedemokratického politického systému na Slovensku a v celom východnom bloku, máme iste skúsenosti s plánovaním. Mnohí sa nad týmto faktom iba pousmejú spochybňujúc päť a viac ročné plány v hospodárstve v tých časoch. Avšak,

keď sa bližšie pozrú na moderný spôsob plánovania, zistia, že veľa sa toho nezmenilo. Ešte viac sa prejavia skúsenosti príslušnej generácie. Rovnako je tomu aj v oblasti plánovania softvérových projektov a aj v iných odvetviach. Týmto nechcem porovnávať metódy plánovania dnes a kedysi, ani spochybňovať prínos súčasných obchodných spoločností z vyspelých krajín, len chcem poukázať na ich aroganciu a ignoranciu v tomto smere. Takýto môj názor je dost' ovplyvnený mojím predchádzajúcim zamestnaním v jednej zahraničnej firme so sídlom vo Francúzsku.

## Záver

V tejto eseji sme sa pozreli na možnosti, ktoré nám niektoré z najpoužívanejších metód tvorby plánov poskytujú. Dodržanie overenej metodiky v kombinácii s pokročilým softvérovým nástrojom na tvorbu plánov nám však ešte nezaručí úspech. Pri plánovaní sú najdôležitejší ľudia. Jednak samotný manažér, ktorý musí plán vytvárať realisticky a mať splniteľné požiadavky na svojich podriadených a aj samotní zamestnanci. Tí musia k projektu pristupovať zodpovedne a plniť daný plán.

V súčasnosti, keď manažér skombinuje všetky možnosti ako overené technológie, svoje skúsenosti, softvérové nástroje na tvorbu plánov, tak je veľmi pravdepodobné, že vytvorí dobrý plán. Dobrý v tom zmysle, že bude znamenať pre projekt prínos, zníži náklady, zvýši efektivitu, vhodne využije dostupné zdroje. Verím, že časom sa možnosti a postupy plánovania ešte zlepšia aj v oblasti softvérových projektov a bude možné presnejšie odhadovať trvanie projektu.

## Použitá literatúra

1. Aigner, W., Miksch, S., Thurnher, B., Biffl, S.: *PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and Their Evaluation*. Los Alamitos, CA, USA, IEEE Computer Society, 2005. Vol. 0. 1550-6037.
2. Biffl, S., Thurnher, B., Goluch, G., Winkler, D., Aigner, W., Miksch, S.: *An empirical investigation on the visualization of temporal uncertainties in software engineering project planning*. Los Alamitos, CA, USA, IEEE Computer Society, 2005.
3. Haugan, G. T., *Effective Work Breakdown Structures*, str. 17
4. Joslin, D., Poole, W.: *Agent-based simulation for software project planning*. Orlando, Florida, Winter Simulation Conference, 2005. 0-7803-9519-0.

## Annotation

### *Methods and practices for planning in software engineering*

Plans have been used for a very long time in all spheres of human activities and also in software projects. When talking about software we speak about plan-controlled development. However, is it appropriate to use such approach in this case? It is difficult to create an exact plan,



considering potential changes. Especially in a software which is to a large extent nonvisualizable. Various methods are used to solve problems with correct estimation of the time period necessary for the project and to create a plan consisting of the individual suitably decomposed tasks. In this essay we will get acquainted with them and we will evaluate their strong points and weaknesses. We will compare the selected methods and we will try to find the right one for creating plans in software engineering.