

IDEÁLNA METÓDA PLÁNOVANIA

*„Manažment a plánovanie projektu musia byť
dynamické a pružné, pretože dynamické je aj okolie.“*

Ján Tóth

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

Abstrakt. Veľké množstvo projektov počas realizácie mešká alebo skončí úplným neúspechom. Existuje veľké množstvo dôvodov, prečo je tomu tak. Významnou mierou na to vplýva aj proces plánovania, vytvárania a rozdeľovania úloh. Každá z týchto činností v konečnom dôsledku predstavuje náročnú a zodpovednú úlohu. Táto práca bude zameraná najmä na problém naplánovania existujúcich úloh v projekte. Existujú zaužívané metódy rozdeľovania úloh ako PERT, CPM, Ganttové diagramy a iné. V rámci výskumu existujú algoritmy a simulácie, ktoré využívajú tieto metódy k plánovaniu úloh v tíme. Poskytujú tak pomerne silný nástroj pre plánovanie projektu a odvrátenie prípadného neúspechu súvisiaceho so zlyhaním plánovania. Táto práca sa bude venovať aj analýze dosiahnutých úspechov spomínaných algoritmov a simulácii. Bude analyzovaná najmä vhodnosť ich použitia pre malé tímy.

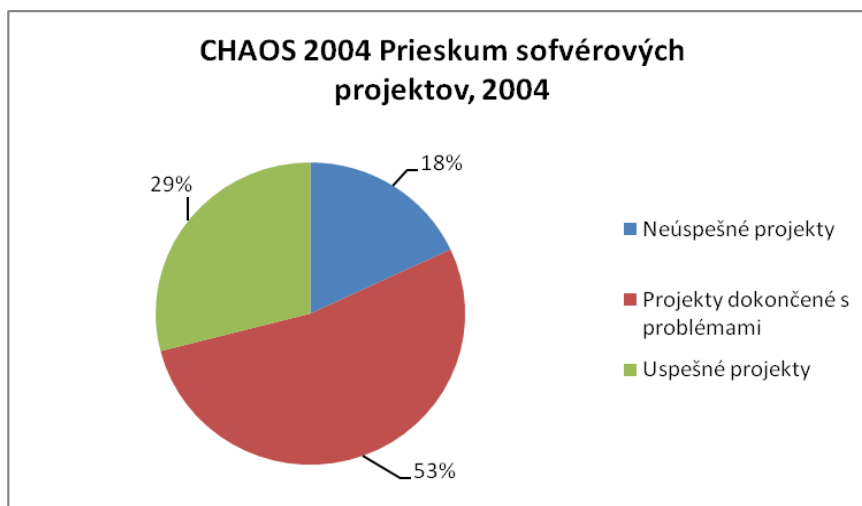
Kľúčové slová: Ganttové diagramy, CPM, PERT, simulácie plánovania

Úvod

Tvorba ľubovoľného softvérového projektu si vyžaduje nemalé úsilie. Je zrejmé, že tento proces potrebuje vhodnú analýzu danej problematiky, vypracovanie návrhu a plánu riešenia a v konečnom dôsledku stanovenie cieľov, ktoré je potrebné dosiahnuť.

Chaos 2004 uznávaný výskum v priemysle a vo vládnych organizáciách z roku 2004 ukázal, že iba 29 % projektov skončilo s úspechom (to znamená, že boli doručené načas,

neprekročili rozpočet, atď.), 53 % všetkých projektov zápasilo počas realizácie s problémami (najmä prekročili svoj rozpočet a plán) a 18 % projektov bolo úplne nedokončených. Projekty, ktoré zápasili s problémami svoj rozpočet prekročili v priemere o 53 % [5].



Obr. 1: Chaos 2004 Prieskum softvérových projektov, 2004 [5]

Na neúspešnosť vplýva veľa faktorov. Zlý manažment rizík, zle identifikované úlohy, zdroje, zlé plánovanie a mnoho iných príčin. Veľmi dôležité je nielen správne identifikovať úlohy, ktoré treba v rámci projektu riešiť, ale aj vhodne organizovať a vytvoriť vhodnú postupnosť ich riešenia. V tejto práci sa pokúsim analyzovať zmysel a rôzne metódy plánovania existujúcich úloh špecifikovaných v rámci projektu.

Plánovať, plánovať, plánovať ...

Prečo vlastne plánovať? Prečo rozdeliť projekt na skupinu menších úloh? Sú to zásadné otázky, kľúčové pre úspešné vyriešenie projektu. Zmysel a význam rozdelenia úlohy (projektu) na menšie úlohy závisí aj na veľkosti zadanej úlohy. Predstavme si jednoduchú úlohu ako napríklad umyť si zuby. Zadanie je jednoduché a úloha si preto nevyžaduje zložité plánovanie alebo rozdeľovanie na menšie úlohy. Pri zadaní komplikovanejšej úlohy ako napríklad uvariť obed, ľudia intuitívne rozmýšľajú o menších úlohách. Automaticky sa vynárajú otázky: Mám kúpené všetky suroviny? V akom poradí ich spracovať a variť? A mnohé iné. Jediným používaným „pomocníkom“ je recept. Zadaná úloha stále nie je príliš zložitá na to, aby si vyžadovala použitie zaužívanej metódy plánovania alebo nejakého sofistikovaného nástroja pre vytvorenie plánu. Táto myšlienka však naznačuje, že pri zložitejších úlohách prirodzene inklinujeme k rozdeleniu úloh na menšie celky, máme tak väčšiu kontrolu nad zložitými problémami a v konečnom dôsledku aj väčšiu kontrolu nad priebehom celého projektu. Neviem si predstaviť, ako by bolo možné riešiť veľký projekt, napríklad rozsiahly informačný systém, bez toho, aby neboli identifikované úlohy a podrobne vypracovaný plán.

Teraz sa pokúsim analyzovať rôzne metódy a techniky používané na plánovanie úloh.

Ganttove diagramy

Ide o také diagramy, ktoré zobrazujú začiatok a koniec jednotlivých úloh v časovom rozvrhu. Je potom jednoduché vyčítať, kedy daná úloha začína a končí, na koľko je splnená a je tu možné vidieť aj nadväznosť jednotlivých úloh. Nevýhoda Ganttových diagramov je tá, že slabo zachytávajú detaily jednotlivých úloh. Okrem toho sa pomocou nich len veľmi ťažko zobrazujú veľké projekty, ktoré obsahujú veľké množstvo paralelných úloh [4].

Metóda kritickej cesty

Táto metóda vychádza z toho, že jednotlivé úlohy majú určený resp. odhadnutý čas, koľko bude trvať ich vypracovanie. Úlohy sú zobrazované najčastejšie pomocou grafu, kde jednotlivé uzly predstavujú úlohy a hrany predstavujú prechody medzi jednotlivými úlohami. Veľkou výhodou tejto metódy je, že takto sa zobrazujú úlohy v poradí v akom ich treba vykonať resp. zobrazujú poradie ich vykonávania tak, ako to bolo naplánované. V angličtine sa metóda označuje skratkou CPM (Critical path method – Metóda kritickej cesty). Táto metóda hovorí o tom, že najdlhšia cesta grafom vlastne prináša informáciu, koľko bude trvať najdlhšia sekvencia úloh v danom projekte. To vlastne znamená, že táto informácia hovorí o tom, ako dlho bude trvať vypracovanie celého projektu.

Technika hodnotenia a revízie projektu

Názov pochádza z anglickej skratky PERT (Project Evaluation and Review Technique). Využíva sa pri plánovaní, monitorovaní a kontrolovaní aktivít, ktoré majú niečo spoločné [6]. Ide o podobnú metódu akou je CPM a často sa pri tvorbe plánov využíva kombinácia metód PERT a CMP. Rozdiel medzi metódou PERT a CPM je, že PERT pozerá na čas trvania úlohy ako ma náhodnú premennú [1]. PERT zavádza okrem odhadovaného času trvania úlohy aj najpesimistickejší a najoptimistickejší odhad trvania úlohy a výsledný čas, za ktorý by mala byť úloha vypracovaná je daný nasledovným vzorcom (t_o – optimistický odhad, t_p – pesimistický odhad, t_m – odhad koľko by malo trvať vypracovanie úlohy za normálnych okolností):

$$t_e = \frac{t_o + 4 * t_m + t_p}{6} \quad (1)$$

V porovnaní metód PERT a CPM je metóda CPM aplikovaná len na vytvorenie čo možno najideálnejšieho plánu. Ak je plán akceptovaný, metóda sa už viac nepoužíva. Naproti tomu, sa metóda PERT okrem toho zaoberá aj dôsledkami časovej dĺžky vykonávaných úloh počas realizácie projektu. Z určitého hľadiska je preto schopná čiastočne reagovať na zmeny počas tvorby projektu [2].

Ktorá metóda je výhodnejšia?

Obe metódy sú veľmi podobné. Pri hlbšej analýze sa dá dospieť k poznatku, že z určitého hľadiska je CPM len špeciálnym prípadom metódy PERT, kedy sú rovnaké všetky tri odhady trvania danej úlohy. CPM používa len jeden údaj o dĺžke trvania nejakej úlohy. Je však tento údaj presný? Do akej miery je možné správne odhadnúť čas úlohy? Ide o pomerne náročný problém a jeho zvládnutie si vyžaduje veľmi dobré manažérske zručnosti. Schopnosť správne odhadnúť čas trvania úlohy je daná nielen intuíciou, ale najmä dôkladným poznaním danej oblasti, ktorej sa konkrétna úloha týka a v nemalej miere aj skúsenosťami. Myslím si, že najdôležitejšia je veľmi dobrá znalosť danej oblasti, vďaka nej je totiž možné zohľadniť všetky problémy, ktoré môžu pri riešení konkrétneho problému nastať a tým určiť relatívne presný čas trvania úlohy. Čas hrá v tomto prípade kľúčovú úlohu. Aj napriek faktu, že analýza úloh bude uskutočnená čo najdôkladnejšie, nazdávam sa, že riziko zlého odhadu je stále pomerne vysoké. Toto je najväčšia nevýhoda metódy CPM, pri jej použití totiž nevieme nakoľko sú údaje presné. V niektorých prípadoch to môže viesť k veľmi zlému naplánovaniu úloh a tým pádom, k nedodržaniu termínov, čo v konečnom dôsledku môže znamenať úplné zlyhanie riešenia projektu. Toto riziko znižuje metóda PERT tým, že zavádza aj najoptimistickejší a najpesimistickejší odhad trvania úlohy. Iste ide o presnejšiu metódu, nakoniec ale aj tak ide len o odhad. Považujem to za slabinu metódy PERT, no napriek tomu súhlasím s faktom, že riziko zlého odhadnutia času potrebného na splnenie úlohy je výrazne menšie.

Tradičná metóda PERT sa najmä zaoberá kritickými úlohami a málo sa venuje iným úlohám, ktoré si tiež zaslúžia pozornosť manažérov [2]. Samotná metóda sa pritom nevenuje mnohým faktorom, ktoré si pri plánovaní zaslúžia pozornosť, napríklad zdroje, ktoré sú potrebné pri práci na jednotlivých úlohách.

Plánovanie s využitím simulácií

James Gantt navrhol systém GITPASE použiteľný pri plánovaní úloh, kde si úlohy navzájom zdieľajú tie isté zdroje [2]. Ako je uvedené v tejto práci, ide o matematicky náročnú úlohu. Systém po vytvorení plánu umožňuje jeho simuláciu. Okrem toho dovoľuje zásah používateľa (ak je to nevyhnutné) a okamžite zobrazuje vykonané zmeny alebo umožňuje navrhovať zmeny tak, aby bol dosiahnutý požadovaný výsledok. Podľa môjho názoru ide o zaujímavý spôsob ako zabezpečiť metódam CPM a PERT istú flexibilitu najmä vďaka tomu, že možné upraviť trvanie úloh alebo prípadne určiť, ktoré zdroje daná úloha vyžaduje. Používateľ má tak možnosť vidieť efekt týchto zmien.

GITPASE podporuje hierarchizáciu a vytváranie podsietí. Vďaka tomu je možné rozdeliť akúkoľvek úlohu na menšie úlohy. Vidím v tom dva dobré dôvody, prečo je táto vlastnosť tohto nástroja dobrá. V prípade, že by išlo o veľký projekt s množstvom úloh, je možné veľké úlohy rozdeliť na menšie systémy. Rozdelenie úloh zvyšuje prehľadnosť a zrozumiteľnosť celého systému. Niekedy môže byť vhodné rozdeliť aj malú úlohu. Poskytne sa tým manažérom určitý „nadhľad“ a vďaka tomu uvidia, z čoho sa úloha skladá. Môže to viesť k zaujímavým poznatkom a objavom. Výsledkom môže byť uvedomenie si, na aké konkrétne úlohy treba rozdeliť hlavnú úlohu resp. z ktorých činností sa daná úloha skladá. Dokonca by sa mohlo dôjsť k záveru, že dané úlohy je

možné riešiť úplne samostatne a mali by byť riešené na úplne inom mieste. Výsledkom by bolo, že by sa plán úloh musel opäť vytvoriť, čo by mohlo viesť k zefektívneniu celého procesu tvorby softvérového systému. Toto je však už veľmi optimistický a extrémny scenár, nič to však nemení na fakte, že by to pomohlo lepšie pochopiť konkrétnu úlohu.

Keď čas a zdroje nestačia

Manažment a plánovanie projektu musia byť dynamické a pružné, pretože dynamické je aj okolie, plán je preto potrebné podľa potrieb zmeniť. Myslím si, že je nedostatočné vytvoriť nejaký plán a potom očakávať, že plán bude bez zmien dodržaný. Ak nejaký manažér takto uvažuje, je to veľmi nedostatočné a pre samotný projekt veľmi zlé. Určite je potrebné počas tvorby projektu reagovať na zmeny, môže dôjsť k neočakávaným problémom pri riešení niektorej z úloh.

Asi najčastejší problém je zle odhadnutý čas trvania nejakej úlohy. Z určitého pohľadu to nemusí byť veľký problém, ak sa to dá vyriešiť počas tvorby projektu, napríklad pridelením väčšieho počtu programátorov, ktorí pracujú na danej úlohe alebo iným spôsobom. Nie vždy to môže pomôcť, niekedy to dokonca môže mať opačný účinok, pretože ak k danej úlohe pridáme nových programátorov, väčšinou sa musia s danou úlohou bližšie oboznámiť, čo opäť zaberie nejaký čas. Konečný výsledok môže byť taký, že úloha by bola dokončená skôr s pôvodným počtom programátorov, ako keby sme ich počet zvýšili.

Ak je pri riešení projektu potrebné príliš často preplánovať úlohy a riešiť podobné problémy, ide o chybu zlého odhadnutia úloh a zlého plánovania. Malým zmenám pri realizácii sa však niekedy nedá vyhnúť. Doposiaľ opísané metódy boli vhodné pri plánovaní úloh, keď sa nepredpokladajú zmeny plánu počas realizácie projektu, čiastočne sú pružné a môžu dobre fungovať aj pri potrebe preplánovania projektu. Za ich nevýhodu považujem fakt, že o jednotlivých úlohách si vedú len jednu potrebnú informáciu – odhadovaný čas potrebný na vypracovanie danej úlohy. Bolo by vhodnejšie, ak by sa uvažovalo aj o iných faktoroch. Podľa môjho názoru je žiaduce, aby sa brali do úvahy aj iné faktory. Napríklad je vhodné uvažovať o tom, ktorým programátorom bude daná úloha pridelená. Je dôležité si uvedomiť, že niektorí programátori nemusia byť s danou oblasťou detailne oboznámení, budú potrebovať určitý čas na naštudovanie danej problematiky, technológií.

Využitie umelej inteligencie pri plánovaní

Ako už bolo spomenuté, bolo by vhodnejšie uvažovať aj o ďalších faktoroch, ktoré vplývajú na určenie času trvania úlohy a taktiež aj k plánovaniu úloh. David Joslin a William Poole vo svojej práci *Agent-based simulation for software project planning* (Simulácie založené na agentoch pre plánovanie softvérových projektov) navrhli algoritmus plánovania s využitím simulácií [3]. Keďže pri plánovaní je nedostatočné uvažovať iba o dĺžke trvania jednotlivých úloh, je potrebné trochu zmeniť pohľad na celý problém plánovania. Doteraz uvedené metódy boli postavené viac-menej na veľmi podobnom princípe. Tento spôsob nielenže neumožňuje zachytiť aj vplyv iných faktorov, ale nie je schopný zachytiť komplexnosť celého procesu vytvárania projektového plánu.

Ich navrhnutý nástroj na plánovanie využíva prvky umelej inteligencie. Umelú inteligenciu si zvolili preto, lebo poskytuje lepšie možnosti ako prehľadať veľký stavový priestor. Samozrejme nie je možné vedieť dopredu budúcnosť a preskúmať úplne všetky možné scenáre, ktoré by mohli nastať. Ich simulačný algoritmus však prehľadáva pomerne veľkú množinu scenárov. Vhodnosť využitia simulácií je daná tým, že je tak dobre zachytiť zložitosť tvorby plánu na pomerne vysokej úrovni a zároveň sú simulácie vhodný nástroj na detekciu chýb [3].

Dôležitým prvkom, ktorý zavádzajú autori je, že úlohy sa rozdelia na tie, ktoré sú povinné a niektoré sú voliteľné, a teda neovplyvnia celkový výsledok. Počas simulácií možných scenárov sa berú do úvahy rôzne faktory ako napríklad koľkým programátorom bude daná úloha pridelená, ako dobre sú oboznámení s danou problematikou alebo aké ďalšie zdroje úloha vyžaduje. Je dôležité, že sa zaoberajú aj tým, koľko zdrojov, ktorá úloha potrebuje. Je zrejme, že môže dôjsť k situácii, že dokončenie úloh brzdí fakt, že si navzájom blokujú potrebné zdroje. Je dobré simulovať aj scenár, kde si jednotlivé úlohy navzájom prenechávajú zdroje, tak aby bolo možné dokončiť všetky úlohy, ktorých sa to týka. Takýto scenár je obzvlášť vhodný, ak sú niektoré z úloh nepovinné. Ako je vidieť z uvedených faktov o tomto projekte, takýto systém prehľadáva veľkú množinu scenárov. Systém je zrejme mienený ako pomocný nástroj pre manažérov pri vytváraní plánu a nie ako samostatný systém, ktorý by za pomoci umelej inteligencie dokázal vytvoriť samostatný plán, ktorým by sa tvorba projektu riadila.

Joslin a Poole dosiahli v rámci výskumu a testovania svojho projektu zaujímavé výsledky. Projekt bol testovaný na množine ôsmich programátorov a boli špecifikované 3 povinné úlohy a 2 voliteľné. Pri testovaní 86 % všetkých parametrov a scenárov pomocou simulácií došli k záveru, že využitím tohto systému nezabránia zlyhaniu iba 2 % projektov. Ak sa testovali úplne všetky scenáre, ktoré systém podporuje, bolo zistené, že 90 % všetkých prípadov, by boli splnené všetky 3 povinné úlohy a 2 nepovinné [3]. Tieto výsledky boli dosiahnuté s nastavením, že úlohy sú si schopné uvoľniť zdroje, ktoré si blokujú, ak by malo prísť k situácii, že nedostatok zdrojov spomalí nejakú úlohu na toľko, že by to ohrozilo dokončenie celého projektu načas.

Plánovanie v malom tíme

Ganttové diagramy podľa môjho názoru predstavujú ten najintuitívnejší spôsob ako plánovať úlohy. Diagram je ľahko pochopiteľný a nadväznosť úloh je dobre čitateľná. Aj pri malom tíme však môže dôjsť k riešeniu relatívne zložitejších a rozsiahlejších úloh. Pri tejto metóde môže dôjsť k problémom napríklad s možným zobrazením všetkých úloh a ich previazaností. Nehovoriac o tom, že diagram nezobrazuje, aké zdroje jednotlivé úlohy vyžadujú. Nie je možné zistiť, ktorí programátori ju budú riešiť a ťažko sa pomocou tejto metódy odhaduje veľkosť projektu. Ganttové diagramy je možné použiť na vytvorenie plánu, ak ide o veľmi malý projekt. Ak je potrebné vykonať zmeny v plánoch počas procesu tvorby projektu, diagramy sú na to veľmi nevhodné. Z týchto dôvodov si myslím, že Ganttové diagramy nie sú vhodné ani v rámci malých tímov.

Ako bolo spomenuté, rozdiely medzi metódami PERT a CPM nie sú také zásadné. Výhodou oboch metód je, že dobre poukazujú na nadväznosť jednotlivých úloh a jasne poukazujú, ktoré úlohy sú kritické (trvajú najdlhšie). Nazdávam sa, že ich najväčšou

slabinou môže byť odhadovanie času. Pri metóde PERT je čiastočne eliminované. Obe metódy podľa môjho názoru nedostatočne pracujú s faktormi, ktoré ovplyvňujú rýchlosť, akou sú jednotlivé úlohy vypracovávané. Vhodným riešením sú simulácie napríklad v rámci nástroja GITPASE. Vhodná je aj možnosť hierarchizácie, ktorá umožňuje lepšie analyzovať jednotlivé úlohy. Aj napriek faktu, že metóda PERT bola vyvinutá najmä pre veľké projekty, myslím si, že je veľmi vhodné ju využiť aj pri práci v malom tíme.

Metódy PERT a CPM sa však stále málo zaoberajú všetkými reálnymi problémami, ktoré môžu pri realizácii projektu nastať. V rámci simulácií síce uvažujú o možnostiach vytvorenia nového plánu z dôvodov prekročenia časov, ktoré boli odhadnuté, z rozličných príčin. Myslím si, že je podstatné sa pri plánovaní zaoberať aj ďalšími faktormi. Dôležité je uvedomiť si, že jedna úloha môže byť pridelená jednému alebo viacerým programátorom. Každý z nich môže pri riešení konkrétnej úlohy naraziť na rôzne problémy a rýchlosť, s akou sa s nimi popasujú, je individuálna. Taktiež je potrebné si uvedomiť, že riešenie úlohy si väčšinou vyžaduje aj iné zdroje, napríklad hardvérové prostriedky. V prípade pomalého riešenia nejakej úlohy je veľmi pravdepodobné, že dôjde k zdržaniu aj ďalších úloh, pretože tie využívajú rovnaké zdroje alebo potrebujú výsledky niektorej z predchádzajúcich úloh. Simulovať podobné scenáre je možné s využitím umelej inteligencie. V prospech nástrojov tohto druhu hovoria aj výsledky výskumu a experimentov. Joslin a Poole testovali svoj nástroj na množine osem členného tímu s malým počtom úloh. Tým sa podobný nástroj stáva veľmi dobrým pomocníkom pri plánovaní v malom tíme.

Záver

Porovnaním Ganttových diagramov, metód CPM a PERT som dospel k záveru, že pri plánovaní je dôležité sa zamyslieť aj nad inými faktormi ako len odhadovaním času, ktorý je potrebný na vypracovanie úloh v projekte. Dôležité je si uvedomiť, že okolie sa dynamicky mení aj počas riešenia projektu. Nie všetky chyby a nástrahy sa dajú odhaliť pri plánovaní. Z tohto dôvodu si myslím, že je vhodné používať nástroje využívajúce simulácie s prvkami umelej inteligencie, ktoré dokážu odhaliť pomerne veľkú množinu chýb, ktoré vzniknú počas tvorby plánu.

Použitá literatúra

1. Douglas D.E.: Pert and simulation. In: *Proceedings of the 10th conference on Winter simulation*, Vol. 1, IEEE Press, New Jersey, USA, 1978, 89 – 99
2. Gantt J.: Interactive PERT simulation modeling for resource-constrained project scheduling. In: *Proceedings of the 18th annual symposium on Simulation*, IEEE Computer Society Press, Los Alamitos, USA, 1985, 128 – 135
3. Joslin D., Poole W.: AGENT-BASED SIMULATION FOR SOFTWARE PROJECT PLANNING. In: *Proceedings of the 37th conference on Winter simulation*, Winter Simulation Conference, Orlando, Florida, USA, 2005, 1059 - 1066

4. Luz S., Masoodian M., McKenzie D., Broeck W. V.: *Chronos: A Tool for Interactive Scheduling and Visualisation of Task Hierarchies*. The University of Waikato, Hamilton, New Zealand, 2009
5. Stephen P. S.: A Simple Estimate of the Cost of Software Project Failures and the Breakeven Effectiveness of Project Risk Management. In: *Proc. of the First International Workshop on The Economics of Software and Computation*, IEEE Computer Society, USA, 2007
6. Swanson H. S., Woolsey R.E.D.: A PERT-CPM Tutorial. In: *ACM SIGMAP Bulletin*, ACM, New York, 1974, 54-62

Annotation

Ideal planning method

Many of software projects are late during their development or their development ends with failure. There are many reasons for this. Planning, specifying and organizing tasks have quite big influence these processes. All these activities represent difficult and responsible task. This paper will be focused on problem of planning specified tasks in software projects. There are many known method to do this, e.g. PERT, CPM, Gantt charts etc. Algorithms and simulations using these methods are part of many studies. They provide strong tool for project planning and they can divert from project failure caused by wrong planning. This paper will discuss success of such studies. I will also focus on suitability of these algorithms and simulations for small teams.