

Flexibilita v plánovaní

LADISLAV NAGY

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

Abstrakt. Dokument skúma najvýhodnejšiu stratégiu pri plánovaní softvéru tak, aby cieľový produkt spĺňal tri hlavné kritériá: kvalitu, rozumný rozpočet a včasné dokončenie. Poskytne prehľad o tom, aké sú rôzne typy plánovania, a porovná ich klady a zápory vzhľadom na splnenie daných kritérií. Tiež sa zaoberá aktualizovaným plánovaním, kde sa menia užívateľské požiadavky počas tvorby projektu, a je potrebné použiť flexibilnejšie metódy, napríklad iteratívne plánovanie. Pri procese plánovania je taktiež dôležité vhodne zvoliť rozdelenie jednotlivých typov úloh v tíme a dohliadať na ich včasné a správne plnenie. Autor súčasne skúma aj to, prečo a do akej miery je dôležité plánovanie pri tvorbe menších študentských projektov.

Úvod

„Formálne schválený dokument používaný na kontrolu a vykonávanie projektu. Primárne použitie projektového plánu je zdokumentovať plánovanie predpokladov a rozhodnutí, uľahčiť komunikáciu vo vedení projektu, a tiež zaznamenať schválené ciele, ceny a základné termíny.“ [1]

Ako aj uvedená definícia uvádza, plánovanie je dôležitým prvkom tvorby softvéru. Pri plánovaní sa vytvára dokumentácia, kde budú zahrnuté všetky dôležité informácie ohľadom projektu. To znamená, že po dôkladnej analýze určíme, aké úlohy treba splniť, ako to budeme robiť a akým spôsobom si rozdelíme jednotlivé úlohy. Musíme taktiež zobrať do úvahy skutočnosť, že úspešný softvérový projekt je taký, ktorý bol realizovaný v stanovenom termíne, pri stanovenom rozpočte a pri danej kvalite produktu. Môžeme teda povedať, že z pohľadu zákazníka má softvérový projekt tri hlavné kritériá:

- Čas potrebný pre vyhotovenie
- Cena
- Kvalita

Manažment projektov softvérových a informačných systémov, október 2008, s. 1-7.

Ideálne pre klienta je, keď majú všetky tri kritériá najlepšiu hodnotu ale musíme si uvedomiť, že medzi sebou v značnej miere súvisia. Napríklad, klient by chcel zvýšiť kvalitu softvéru pridaním nových funkcií, keďže ale máme stále rovnako veľký vývojový tím, tak sa s určitosťou predĺži čas, ktorý potrebujeme na dokončenie. Keď sa rozhodneme, že je dôležité dokončiť projekt včas, tak musíme prijať ďalších zamestnancov, teda stúpne celková cena. Vo všeobecnosti platí, že ak chceme zlepšiť jednu z kritérií, tak sa niektorý zhorší. Cieľom projektového plánu je čo najlepšie optimalizovať tieto tri kritériá, pričom musíme dbať na to, aby zostali v rovnováhe.

Okrem hlavných kritérií sa v poslednej dekáde zameriavame aj na ďalší dôležitý faktor, a to je uspokojenie zákazníka, či už pod tým myslíme spokojnosť s konečným produktom, dobrou spoluprácou či flexibilitou. Najmä z pohľadu flexibility budeme skúmať jednotlivé typy plánovania a budeme porovnávať to, v akej miere sú vhodné pri riešení softvérových projektov. Dve hlavné typy plánovania z tohto hľadiska sú:

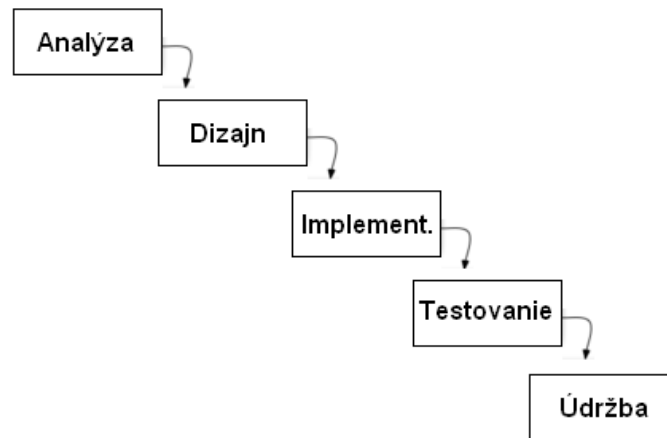
- Racionálne (tradičné) plánovanie
- Iteratívne (agilné) plánovanie

Racionálne plánovanie

Pri racionálnom plánovaní sa využíva tzv. „vodopádový model“ (obr. 1.) [3], kde sa každý krok vykonáva sekvenčne v danom poradí: analýza požiadaviek, dizajn, implementácia, testovanie a údržba; pričom platí, že kým sme neskončili predchádzajúcu časť plánu, tak nezačínáme ďalšiu. Prvýkrát tento zaviedol Winston W. Royce v roku 1970. Bol to prvý model, ktorý bol širšie využívaný pri tvorbe softvérových projektov. [5]

Existujú samozrejme obmeny vodopádového modelu, kde nás model nenúti kroky vykonávať sekvenčne. Jedná sa napríklad o Model „Sashimi“, pri ktorom sa za sebou idúce fázy čiastočne splývajú.

Ďalším príkladom je „vodopádový model s redukciou rizika“, kde do určitej miery rátame s nečakanými udalosťami, a model nám dovolí v určitom rozsahu vrátiť sa na predošlú fázu.



Obr. 1. Znáozornenie vodopádového modelu.

Racionálne plánovanie prináša do tvorby projektu disciplínu, keď je každá fáza životného cyklu presne vymedzená či už časovo alebo rozsahom úloh, teda zákazník aj vývojový tím môže ľahko kontrolovať napredovanie projektu. Dôraz kladený na požiadavky a dizajn ešte pred napísaním čo i len jedného riadku kódu znižujú pravdepodobnosť toho, že projekt nebude hotový včas alebo nesplní požiadavky klienta. Pri dôslednom plánovaní je možné tiež odhaliť určité chyby, ktoré by sa pri testovacej fáze ťažko hľadali a opravovali. Racionálne plánovanie je výhodné, keď sa navrhujú väčšie systémy, pri ktorých musí spolupracovať veľký počet ľudí a pri ktorom sa vôbec alebo len v malej miere menia požiadavky zákazníka. [7]

Práve spomínané zmeny požiadaviek sú aj najväčšou slabinou, nakoľko v drivej väčšine prípadov zákazník nevie presne, čo od projektu očakáva, a niektoré svoje požiadavky vytvorí až pri tvorbe projektu. Plán projektu sa ale v tomto štádiu vývoja mení ťažko a to zvyšuje zdroje a čas potrebné na ukončenie. Ďalším záporom je, že väčšina navrhnutých projektov síce vyzerá pri návrhu ako skvelý a jednoducho implementovateľný, ale v množstve prípadov je realita úplne iná a vývojový tím tak často končí v slepej uličke a musí úplne prerobiť pôvodný plán.

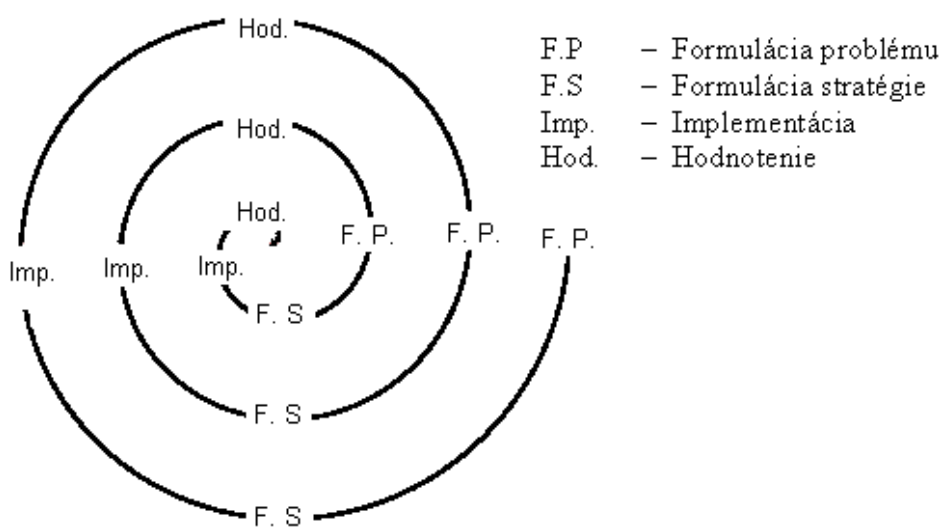
Iteratívne plánovanie

Pojem iteratívne plánovanie (obr. 2.) znamená, že v určitých intervaloch vyhodnocujeme, či projekt stále smeruje dobrým smerom. Vo všeobecnosti máme otvorené vždy dve iteratívne fázy.[4] Prvá fáza je tá práve prebiehajúca a druhá je nasledujúca, pričom v terajšej fáze už sa plánuje, ako bude prebiehať tá nasledujúca. Samozrejme to vyžaduje od projektového manažéra agilný prístup a to aj v tom prípade, keď zmena plánu dorazí v neskoršej fáze terajšieho cyklu, aby bol plán včas hotový pre ďalšiu fázu. Každá iterácia obsahuje úlohy, ktoré rozsahovo nepresahujú dvoj- alebo maximálne trojdňové intervaly. Keď je zadefinovaná takáto úloha, tak je

najvýhodnejším riešením rozloženie projektu na podúlohy. V podstate je založené na princípe priebežnej adaptácie „Muddling trough“ prezentované Lindbolnom v roku 1959:

„Použitie množstva malých krokov pri adaptácii, namiesto veľkých a viac závažných zmien“.[2]

Pri každom kroku skúmame, či sme splnili ciele daného intervalu a či je potreba implementovať určité zmeny.



Obr. 2. Znáozornenie iteratívneho plánovania.

Vzhľadom na kratšie intervaly iterácií máme vždy skorú odozvu od zákazníka a dokážeme ľahšie integrovať prípadné zmeny. Dosiahne sa tým, že zákazník sa dostane oveľa skoršie k základnej verzii a dodatočná funkcionálna sa pridáva po každej iterácii. Intervaly a častá odozva zákazníka zaručujú aj to, že sa vývojár zamerá na dôležité problémy a nezačne implementovať také časti, ktoré neboli explicitne definované zákazníkom.

Pri nevhodne krátkych intervaloch ale dosahujeme opačný efekt, kde už trávime viac času retrospektívou a preplánovaním, ako samotnou tvorbou projektu.

Pri rozložení na podúlohy sa nám môže stať, že nebudeme mať dobrý prehľad o celkovom obraze projektu.

Porovnanie

V predošlých dvoch častiach sme predstavili dva hlavné typy plánovania a uviedli sme ich kladné a záporné stránky. Nasledujúca tabuľka (Tab. 1.) obsahuje porovnáva ich hlavné črty.

Tab. 1. Iteratívne a Racionálne plánovanie porovnanie.

	Iteratívne plánovanie	Racionálne plánovanie
Veľkosť tímu	4 až 15	nad 15
Flexibilita	vysoká	nízka
Prípady použitia	priebežne sa meniace požiadavky	pevne stanovené požiadavky
Životný cyklus projektu	sekvenčne opakujúce sa fázy	pevne stanovené fázy

Pri porovnávaní sa len ťažko môžeme rozhodnúť, ktorá je správna metóda a ktorú bezpodmienečne musíme použiť, aby sme sa dopracovali k úspešne ukončenému projektu. Univerzálne riešenie neexistuje a ku plánovaniu každého projektu treba pristupovať osobitne. Pri skúmaní črt ale môžeme ľahšie odhadnúť, aký typ plánovania sa hodí na daný projekt viac a aký menej.

Aj keď má racionálne plánovanie a vodopádový model značné nedostatky, stále sa využíva pri množstve špecializovaných produktov, hlavne tam, kde už má zákazník presnú predstavu o požiadavkách a výslednom produkte. Podľa môjho názoru pri tvorbe moderných softvérových systémov má iteratívne plánovanie väčšiu škálu využitia hlavne kvôli svojej flexibilita. Kritické v tomto prípade je ale správny odhad dĺžky iterácií, nakoľko pri nesprávnom výbere môžeme stráviť viac času revíziou plánovania ako samotnou tvorbou projektu. Taktiež platí, že je oveľa viac takých projektov, kde veľkosť tímu nepresahuje hranicu 15-ich členov a je teda výhodnejšie postupné osvojovanie si vedomostí a flexibilnejšie riešenie projektu, ako detailné plánovanie.

Iteratívne plánovania v študentských projektoch

O iteratívnom plánovaní môžeme hovoriť aj pri menších študentských projektoch, napríklad pri tímovom projekte. Pri každom stretnutí si tím najprv urobí revíziu plánu z predošlého týždňa, skúma či boli splnené všetky plánované úlohy, poprípade prediskutujú nedokončené a problémové oblasti a podľa toho sa vytvorí plán a stanovia sa ciele pre dané obdobie. Podľa môjho názoru je taký typ plánovania najlepší pre projekty tohto typu.[6]

Hlavným dôvodom použitia interaktívneho plánovania je, že na začiatku daného projektu sa vo väčšine prípadov tímoví kolegovia medzi sebou nepoznajú, takže nevedia, čo možno očakávať od druhého, a aké sú klady a zápory jednotlivých členov. Pri revízií plánovania sa silné a slabé stránky jednotlivcov prejaví a podľa toho potom môže byť prispôsobené ďalšie smerovanie projektu. Taktiež platí, že sa nevytvárajú

zložité systémy, kde by implementácia jednotlivých modulov a plnenie úloh trvala niekoľko týždňov či dokonca mesiacov, takže potrebujeme posudzovať a vyhodnocovať plnenie plánov v kratších intervaloch. Presný rozsah závisí už od jednotlivých skupín ale podľa mojich skúseností by veľkosť jednotlivých iterácií nemala presahovať 2 týždne.

Záver

Cieľom eseje bolo oboznámenie sa so základnou problematikou plánovania. Spoznať dôvody, prečo vlastne plánujeme a aké sú hlavné kritériá plánovania. Skúmali sme dve dôležité metódy plánovania iteratívne a tradičné plánovanie. Po oboznámení sa s kladmi a zápormi týchto dvoch metód sme ich porovnávali a zisťovali, pri akých projektových vlastnostiach je ich použitie vhodné. Boli skúmané hlavne pri zmene užívateľských požiadaviek počas životného cyklu softvérového projektu, teda hlavne z hľadiska flexibility. Nakoniec sme zisťovali, prečo je vhodnejšia flexibilná metóda pri riešení malých študentských projektov.

Použitá literatúra

1. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, Third Edition, Project Management Institute. ISBN 1-930699-45-X.
2. Barstad Johan: *Iterative Planning Processes: Supporting and impeding factors*, February, 2002, Norway.
3. Gilb Tom: *Evolutionary Delivery versus the "waterfall model"*, ACM SIGSOFT Software, Engineering Notes, V.10 I. 3 July 1985.
4. Grisham S. P., Perry D. E.: *Customer relationships and Extreme Programming*, ACM SIGSOFT Software, Engineering Notes, v.30, n.4, July 2005.
5. Paulk C.: *The Rational Planning of (Software) Projects*, Software Engineering Institute, Pittsburgh, USA 1993.
6. Rettig M., Simons, G.: *A Project Planning and Development Process for Small Teams*, *Communications of the ACM*, Vol. 36, No. 10, (Oct 1993), 45-55.
7. Royce, Winston (1970): *Managing the Development of Large Software Systems*, Proceedings of IEEE WESCON 26 August 1970, 1-9.

Annotation

Flexibility in planning

The document examines the best strategy for software planning, so that the final product satisfies the three main criteria, quality, reasonable budget and completing due time. It grants an overview of the various types of planning, and compares the pros and cons considering the accomplishment of the given criteria. It is also dealing with actualized planning, when the user requirements change during the construction of the project, and it is necessary to use more

flexible methods, for instance iterative planning. During the planning process it is essential to properly choose the distribution of the assignments in the team, and looking after that it is fulfilled in time and with accuracy. The author concurrently examines also why, and in what scale is planning necessary in the creating of small scale student projects.