

# EVOLUČNÉ PLÁNOVANIE S HEURISTICKÝM POHĽADOM DO BUDÚCNOSTI

*Čo dokážeme zmeniť, zmeňme; čo zmeniť  
nedokážeme, zmení nás.*

*Peter Mindek*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
mindek06 [zavináč] student.fiit.stuba[.]sk

**Abstrakt.** Úspech každého väčšieho projektu je už od začiatku odsúdený na neexistenciu, ak započatiu prác na projekte nepredchádza plánovanie. Vytvorenie, a predovšetkým udržiavanie plánu a jeho priebežné prispôbovanie okolnostiam je esenciálne pre udržanie správneho smeru vývoja projektu. Plán by mal hľadiť dopredu. Odhaliť situácie, ktoré by mohli nastať, a predložiť návod, ako ich zvládnuť. Aj keď práve plán by mal predchádzať všetkým nečakaným situáciám, pred prekvapeniami členov tímu dokonale ochrániť nedokáže. Preto je dôležité aktualizovať plán a nespoliehať sa na jeho prvú verziu, ktorá bola vytvorená bez znalostí o priebehu projektu. Pri vytváraní plánu je dôležité zohľadniť všetky dostupné informácie. Často sa však zabúda na zohľadnenie dynamiky progresu vývoja projektu. Táto esej sa zaoberá predovšetkým prispôbovaním stratégie plánovania doterajšiemu vývoju, a zohľadňovaním nadobudnutých skúseností pri odhadoch budúceho vývoja.

**Kľúčové slová:** projekt, plánovanie, tím, dynamika, vývoj, rozdeľovanie úloh, odhadovanie

## Úvod

Komplexnosť sveta, v ktorom žijeme, neustále rastie. Prejavuje sa to samozrejme aj na projektoch, na ktorých ľudia pracujú. Softvérové inžinierstvo je jednou z oblastí, v ktorých

zložitosť a rozsah projektov stúpa závažným tempom. Okrem toho, softvérové projekty sú čoraz dôležitejšie aj pre rôzne iné oblasti, a to ich robí ešte náročnejšími na úspešné uskutočnenie. Existuje veľa prípadov neúspešných projektov. Treba však povedať, že hlavnou príčinou neúspechu nie je zložitosť projektu, ale povaha a schopnosti ľudí [5]. Pre softvérové projekty sú najdôležitejšie ľudské zdroje, schopnosti tímu. A z toho sa dá ťažiť. Nedá sa zrejme spraviť nič pre zníženie komplexnosti projektov. Naproti tomu, ľudia je možné (aspoň do určitej miery) riadiť. A tak predsa len nejaká nádej na úspech existuje.

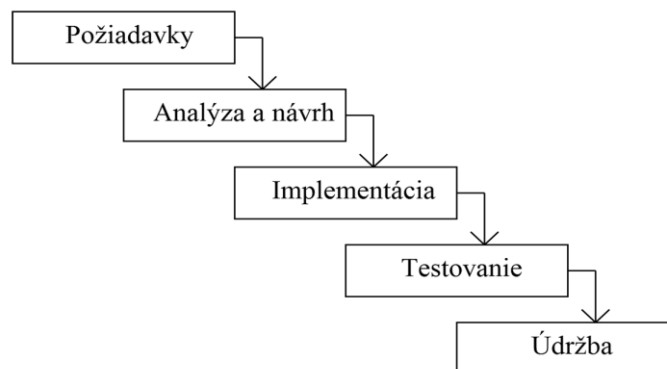
Intuícia vraví, že každý projekt sa skladá z postupnosti krokov, ktoré treba vykonať. Hovoríme o postupnosti krokov a nie o jednom veľkom „kroku“, pretože pre ľudský mozog je prirodzené každú činnosť deliť na jednoduché, atomické akcie. Pre splnenie každej netriviálnej úlohy je nutné tieto akcie, alebo kroky, dopredu naplánovať (napr. určiť poradie ich vykonávania). Vytvorenie plánu je akýmsi pohľadom do budúcnosti, pokusom o odhadnutie možností využitia dostupných zdrojov pre čo najefektívnejšie splnenie zadanej úlohy zlozenej z menších podúloh. Ide o vytýčenie cesty, ktorá vedie k úspešnému zavŕšeniu projektu.

Vytváranie plánov je neodlučiteľnou súčasťou života každého manažéra. Dalo by sa povedať, že úspech či neúspech projektu sa priamo odvíja od plánu, ktorý projekt uviedol do života, a ktorý projektu počas celého jeho životného cyklu udával smer. Je preto samozrejmé, že proces plánovania sa manažéri snažia čo najvýraznejšie optimalizovať. Či už pridržaním sa rôznych publikovaných metodík, alebo akumulovaním vlastných skúseností do viac-menej jednotného celku pravidiel a postupov, ktoré sa manažérovi preukázali ako vhodné pri riadení práce na projekte. Väčšinou sa jedná o odporúčania a postupy týkajúce sa jednotlivých aspektov vytvárania plánu. Príkladom môže byť odporúčanie využívať psychologické testy ako pomôcku pri pridelovaní úloh v projekte. Tieto odporúčania sú poväčšine overené praxou a rokmi skúseností manažérov z komerčného prostredia, a preto v značnej miere napomáhajú zvyšovaniu efektivity plánovania. Za ich nevýhodu by sa však dala považovať ich statickosť. Je úlohou manažéra prispôbiť metodiky a postupy plánovania požiadavkám konkrétneho projektu, avšak ich formalizmus a striktnosť s akou interpretujú empiricky overené fakty, môže často zvädzať k čo najpresnejšiemu pridržiavaniu sa daného postupu.

Keď vravíme o statickosti metód plánovania, nemáme na mysli ich tendenciu viesť k statickým plánom. Niet pochýb o tom, že dobrý plán musí byť dynamický, prispôsobujúci sa okolnostiam a dynamike vývoja projektu. Presné plnenie plánu so sebou takmer bez výnimky prináša nutnosť modifikovať ho, a väčšina manažérov si tento fakt plne uvedomuje. Bolo by však zaujímavé zamyslieť sa nad tým, že postupnú modifikáciu by si mnohokrát zaslúžil aj prístup manažéra k vytváraniu plánu, a nie len plán samotný. Niečo ako ciele poučenie sa z vlastných chýb, potopenie vlastného ega v záujme vyššej kvality výstupu.

### **Vodopádový model vývoja softvéru**

Každý väčší ( netriviálny) softvérový projekt musí byť nejakým spôsobom riadený. Jednou z úrovní riadenia, ktorá do istej miery súvisí aj s plánovaním, je pridržiavanie sa určitého modelu vývoja softvéru. Asi najznámejším, najjednoduchším a zároveň jedným z najstarších modelov je vodopádový model, ktorý je znázornený na obrázku 1.



**Obr. 1.** Vodopádový model vývoja softvéru [4]

Po špecifikovaní požiadaviek klienta je verifikovaná ich uskutočniteľnosť. Ďalším krokom je analýza a návrh, ktoré predchádzajú implementácií softvérového systému. Tieto dva kroky majú podiel na plánovaní projektu, pretože bez nich by bola implementácia s najväčšou pravdepodobnosťou tak chaotická, že by sa k úspechu vo väčšine prípadov ani nepriblížila. Po implementácii prichádza na rad testovanie, overenie funkčnosti systému a splnenia špecifikovaných požiadaviek. Posledným krokom je údržba po nasadení softvérového systému. Existuje viacero pohľadov na vodopádový model vývoja softvéru, avšak princíp je pri všetkých rovnaký – rozdeliť prácu na projekte do logických, po sebe nasledujúcich blokov činností. Po pridaní termínov k jednotlivým činnostiam a rozdelení úloh je plán na svete.

S takýmto plánom sa však dá pomerne ťažko manipulovať v prípade nečakaných situácií. Ak sa napríklad nestíha implementácia, je možné prideliť na jej plnenie viac ľudských zdrojov, ako bolo pôvodne plánované. Nič viac sa však spraviť nedá. Posúvať termíny je možné iba na úkor nasledujúcich úloh. To však nie je dobré, lebo zameškaný čas treba vynahradiť niekde inde.

Vodopádový model sa ukazuje ako veľmi riskantný – okrem iného preto, lebo neposkytuje manažérovi potrebnú flexibilitu a možnosti prispôsobovania plánu okolnostiam. Lokálne zlyhania majú v tomto modeli tendenciu stávať sa zlyhaniami globálnymi. Vytvorených bolo preto množstvo iných modelov, ktoré sú vhodnejšie na riadenie väčších softvérových projektov.

Rovnako, ako aj ďalšie modely vývoja softvéru, Vodopádový model obsahuje dôležitý prvok. Je ním dekompozícia projektu na menšie samostatné úlohy, ktorých vykonanie vedie k úspešnému završeniu projektu.

## Evolučné plánovanie

Pojem *evolúcia* sa od nepamäti spája s predstavou akéhosi pokroku, modernosti, inovatívnosti. Vychádza to azda z poznania človeka, že skúsenosti nikdy nie je možné prestať nahrádzať novými. Presvedča nás o tom aj nespočetné množstvo ľudových i neľudových múdrostí o učení sa celý život, prípadne o tom že viem, že nič neviem. Na mieste je však otázka, prečo načrieť do studnice múdrostí takéhoto rangu pri úvahách o plánovaní projektu, menovite toho softvérového? Odpoveď je jednoduchá – život

človeka je totiž peknu metaforou pre plánovanie projektu. Na začiatku plánovania projektu stojí zhluk vnemov, *brainstorming*, rozličné nápady a postrehy. Z nich sa postupne, prirodzenou cestou vytvára istý smer, ktorého sa tvorba plánu bude neskôr pridržať. Plán je vtedy akoby v puberte. Možno vie, čo čakať od života (projektu), stále je však v procese formovania sa do finálnej podoby. Tu však bohužiaľ metafora veľa krát končí. Plánovanie nadobudne istý smer a úlohou manažéra je už iba dohliadať na projekt a synchronizovať jeho priebeh s plánom. Manažér tak môže zabrániť tomu, aby projekt začal po prílišnom odchýlení od plánu tápať. Bežnou praxou je prerozdelenie úloh, prípadne iné zmeny súvisiace s riadením projektu. Plán je síce dynamický, ale pomerne neprispôsobivý.

Vytváranie plánu môže byť vnímané viac ako umenie, než ako exaktná veda. Manažér musí získať čo najviac vstupov, utvoriť si predstavu o projekte a o schopnostiach a možnostiach svojho tímu. Do procesu plánovania musí taktiež zahrnúť obmedzenia, a premenné (konštanty), s ktorými sa nedá manipulovať (typickým príkladom je termín, v ktorom musí byť projekt dokončený). Na základe týchto informácií dokáže manažér vytvoriť rozdelenie úloh v tíme, ako aj časový rozvrh, ktorého by sa tím mal pridržovať. Ak sa niečo nestíha, na úlohu je pridelených viac ľudí, a podobne.

Problém pridelenia ľudí k jednotlivým úlohám je problém lineárneho programovania, známy ako „Priradovací problém“. Jeho cieľom je priradiť ľudí k úlohám minimalizujúcu funkciu ceny priradenia. Cena priradenia môže byť chápaná ako vhodnosť človeka na riešenie danej úlohy [2]. Priradovací problém sa rieši takzvanou maďarskou metódou. Treba si však uvedomiť, že priradovací problém je iba abstraktným opisom situácie a je potrebné aplikovať jeho riešenie v praxi. V kontexte plánovania softvérového projektu, kde ide predovšetkým o manažment ľudských zdrojov, treba hľadať konkrétne (aj keď možno neoptimálne) metódy, ako problém vyriešiť. Vstupy sú často nekonzistentné, pretože o každom členovi tímu môže mať manažment iné informácie. Napriek optimalizačnej metafore, spravovať tím ľudí je iné ako spravovať súbor dát.

Častým problémom ľudí je to, že sa im zle prispôsobuje plánu, ktorý niekto vytvoril za nich. Na mieste je preto úvaha, do akej miery je možné plán prispôbiť ľuďom. V súvislosti s priradovacím problémom a jeho riešením sa naskytuje myšlienka poveriť samotných členov tímu, aby rozhodovali o svojich úlohách. Ich čiastočné zapojenie do procesu plánovania môže priniesť pozitívne výsledky. Získať dáta o človeku a vyhodnotiť ich môže byť totiž v niektorých prípadoch menej efektívne, než ho jednoducho požiadať o sebahodnotenie. Jedná sa o praktický príklad známeho úslovja „nechať problém, nech sa vyrieši sám“.

Príkladom je vytvorenie zoznamu atomických úloh s pridelenými termínmi, ktoré vyplývajú z prvotného plánu. Každý člen tímu si potom zvolí úlohu, na ktorej bude pracovať, a ktorú stihne dokončiť v stanovenom termíne (aspoň podľa svojho najlepšieho vedomia a svedomia). Ostáva potom určiť nejaké základné pravidlá pre pridelovanie úloh (každý člen tímu pracuje na minimálne jednej úlohe, na žiadnej úlohe nepracuje viac ľudí ako je potrebné, a pod.), a plán sa bude vytvárať sám, priebežne počas práce na projekte.

Táto predstava je však do istej miery utopistická, pretože nezohľadňuje možnosť uskutočňovať zmeny v prvotnom pláne. Ako bolo načrtnuté vyššie, takéto zmeny sú (predovšetkým pri väčších projektoch) prakticky isté, lebo dokonalí ľudia neexistujú (či už sa jedná o členov tímu, alebo o klientov zadávajúcich požiadavky). Cieľom je preto už pri

vytváraní prvotného plánu myslieť na jeho budúci vývoj. Nechať mu priestor na zmeny, ktoré keď niekedy v budúcnosti nastanú, budú mať iba lokálny vplyv. Tento priestor bude stále menší s blížiacim sa vyvrcholením prác na projekte.

Ak chce manažér vytvárajúci plán dosiahnuť úspech, mal by podrobne sledovať prácu svojho tímu. Pri softvérových projektoch je to obzvlášť dôležité, keďže ľudia v ňom zohrávajú najdôležitejšiu úlohu. Manažér by mal vedieť, ktorí člen tímu sa hodí na aký typ úloh, a táto vedomosť by sa mala postupne spresňovať. Je to tá povestná „zohranosť“ tímu, vďaka ktorej je po niekoľkých spoločných projektoch práca efektívnejšia.

V kontexte jedného projektu však zohranosť, a teda aj možná efektivita, stúpa s časom stráveným spoločnou prácou na projekte. To že je táto efektivita možná však neznamená, že sa dostaví sama od seba. Zohranosť tímu sa v takomto prípade neprejavuje tak nonšalantne ako pri dlhodobej spoločnej činnosti, a treba ju nejakým spôsobom podnecovať. A to je práve úloha pre manažéra.

Človek z pravidiel vyhľadáva najlepšie možnosti preňho samého. Ak by si členovia tímu vybrali úlohy tak, ako to bolo opísané v predchádzajúcom príklade, celková efektivita tímu by s najväčšou pravdepodobnosťou nekonvergovala k optimu. Manažér zodpovedný za plánovanie by však mohol do istej miery riadiť delbu práce na základe skúseností, ktoré pri spolupráci s tímom nadobudol, respektíve vysledoval (napríklad určovaním množín úloh pre jednotlivých členov tímu; tieto množiny by sa prekrývali, aby členom tímu zároveň ostala určitá voľnosť výberu). Ak sa ukáže, že niektorý člen tímu má tendenciu písať neefektívny programový kód, prečo mu namiesto implementácie hardvérového ovládača neprideliť radšej úlohu súvisiacu s používateľským rozhraním, kde optimálnosť kódu nie je až taká podstatná? Manažér by tu mal nájsť kompromis medzi altruizmom a všeobecným záujmom o úspech projektu, v ktorom musí byť niektorým členom tímu odopretá časť voľnosti. Aká veľká časť, to záleží len od ľudí. Ako koniec koncov všetko, čo sa plánovania týka.

Treba ešte spomenúť, že veľa záleží od prvotného plánu, ktorý je vytvorený na začiatku projektu. Mohlo by sa zdať, že dôležitosť tejto verzie plánu nie je taká veľká, keďže tento bude počas projektu aj tak menený pod vplyvom novozistených informácií. Opak je však pravdou. Aj keď prvotný plán neobsahuje úplne jasnú predstavu o všetkých aspektoch projektu, odhady niektorých z nich musia byť v prvotnom pláne zahrnuté. Jedná sa predovšetkým o časový manažment – termín odovzdania musí byť jasný už na začiatku a cieľom je tento termín dodržať (prípadne minimalizovať potenciálne meškanie). Keďže budúcnosť nepoznáme, je dobré byť pri plánovaní pesimistom. Možné prekvapenia majú potom väčšiu šancu byť pozitívnymi. Pri rozdelení projektov na menšie úlohy je dobré určiť takú postupnosť úloh ktorá vedie k dokončeniu projektu v najdlhšom možnom čase [1]. V prípade paralelných úloh a zložitejších závislostí medzi nimi môže byť takýchto postupností viac. Údaj o trvaní takejto postupnosti môže zohrať rolu pri určovaní pesimistického odhadu trvania projektu.

Evolučné plánovanie by sa teda dalo opísať ako plánovanie, ktorého záujmom je formovať priebeh projektu takpovediac za behu, avšak nie chaoticky, ale organizovane a predovšetkým zohľadňujúc všetky informácie vyplývajúce z doterajšej práce. Manažment plánovania by sa mal učiť, vyvíjať a vylepšovať s každou novou informáciou. A novou informáciou je všetko, čo sa pri práci na projekte udeje.

## Predpovedanie budúcnosti

Plánovanie, a predovšetkým dobré plánovanie, vyžaduje jednu dôležitú zručnosť – schopnosť predpovedať budúcnosť. I keď na prvý pohľad sa môže tento termín zdať nepresný a ako vhodnejšia alternatíva sa núka slovné spojenie „odhadovanie budúcnosti“, domnievam sa, že pri plánovaní ide predovšetkým o predpovedanie – určenie čo najväčšieho počtu možných budúcich scenárov. Odhadnutie, ktorý z týchto scenárov nastane, nie je pri plánovaní až také podstatné, pretože je takmer nemožné spraviť takýto odhad spoľahlivo. Istá dôležitosť však prináleží všeobecnejšej forme odhadovania – určenie pravdepodobností, s akými nastanú jednotlivé scenáre. Tento odhad nemusí (a ani nemôže) byť vo všetkých prípadoch veľmi presný, ale prináša manažérovi užitočné informácie a stavia jeho „predpoveď budúcnosti“ na vyššiu úroveň.

Je logické, že čím viac informácií má manažér k dispozícii, tým zreteľnejšie „vidí“ do budúcnosti. Veľké množstvo nových informácií sa vynára počas priebehu prác na projekte. Tu vidíme, že priebežne upravovať plán je nie len nevyhnutné, ale dokonca aj pomerne výhodné. Plán by mal byť vždy čo najpresnejší a možná presnosť postupne stúpa s každou novou informáciou (až na výnimky).

Vytvárať plán by bolo samoučelné, keby neslúžil k efektívnemu dosiahnutiu cieľa, ale bol by neustále prispôsobovaný chaotickému vývoju projektu. Takáto situácia by mohla nastať, keby plán nekorešpondoval s uskutočniteľnou realitou. Pri evolučnom plánovaní je preto dôležité vytvoriť prvotný plán a súbor podmienok vývoja plánu tak, aby skutočne tvorili plán, a nie akýsi opis prebiehajúcich prác na projekte.

## Záver

Evolučné plánovanie nie je formálnou metódou, ale súborom myšlienok. Tieto myšlienky sa v praxi pri plánovaní projektov často využívajú, avšak mnohokrát nevedomky, prípadne necielene. V tejto práci je zahrnutých niekoľko nápadov, ktoré by eventuálne mohli pomôcť pri vytváraní stratégie plánovania väčšieho softvérového projektu. Práca taktiež pojednáva o odhadoch budúcnosti, ako o významnom prostriedku plánovania. Myšlienku evolučného plánovania s heuristickým pohľadom do budúcnosti je nutné aplikovať až po prispôbení pre konkrétny projekt, pretože v opačnom prípade môže viac uškodiť, ako pomôcť.

## Použitá literatúra

1. BIUNDO, S. – HOLZER, R. SCHATTENBERT, B.: Project planning under temporal uncertainty. Planning, Scheduling, and Constraint Satisfaction: From Theory to Practice. Volume 117 of Frontiers in Artificial Intelligence and Applications. IOS (2005)
2. COLUCCI, S. – DI NOIA, T. – DI SCIASCIO, E. – DONINI, F. – MONGIELLO, M. – PISCITELLI, G.: Semantic-based approach to task assignment of individual profiles. Journal of Universal Computer Science (2004)
3. DRABBLE, B.: Knowledge Based Project Planning (1995)

4. LEWALLEN, R.: Software Development Life Cycle Models. Dostupné na internete: <http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx> [cit. 2009-october]
5. PAULK, M.: The Rational Planning of (Software) Projects. Proceedings of the First World Congress for Software Quality (1995)

## **Annotation**

### *Evolutionary planning with heuristic outlook*

*Success of every project relies on planning. Plan creating and updating is very important for project to proceed correctly. The plan should look ahead, reveal situations, which are likely to happen, and propound guidelines to handle those situations. Even though the plan itself should prevent unexpected situations from happening, surprises are inseparable part of every project. It is therefore important to actualize plan and do not fully bank on its first version, the one created at the very beginning of the project, without any relevant information about project progress. The paper presents some concepts of evolutionary planning as a technique for effective management of a software project. It is focused on continuous plan modification usefulness. Risk factors of different aspects of planning are concisely described, along with some ideas to minimize those risks.*