

# AKO SI NENAPLÁNOVAŤ NEÚSPECH

*Čas je najvzácnejšia vec, ktorú človek má.*

*Peter Michalec*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
michalecp[zavináč]pobox[.]sk

**Abstrakt.** *Plánovanie rôznych rozvrhov je v dnešnej dobe bežné nielen v školách, ale aj v rôznych firmách, či osobných životoch. Taktiež zohráva veľkú úlohu pri softvérových projektoch ako pre jednotlivcov, tak aj pre tímy ľudí. Každý sa snaží v rámci svojich schopností dobre naplánovať daný projekt, i keď miera, do akej je projekt dobre naplánovaný je ťažko merateľná. Softvér na zákazku sa tvorí už dlhé roky a vznikli viaceré rady, ktorými by sa mali manažéri držať pri plánovaní, aby sa im celý projekt nezrútil. Metodika scrum, ako jedna z agilných metód vývoja softvéru, sa snaží poskytnúť riešenie ako zlepšiť a sprehľadniť toto plánovanie softvérových projektov. Po niekoľkých rokoch skúseností s touto metodikou existujú rady, ako o niečo vylepšiť takéto plánovanie. Dôležité je však dávať pozor a snažiť sa riadiť týmito radami. Toto je hlavne dôležité pre neskúsených študentov, ktorý sa po prvý krát stretávajú s tímovou prácou na predmete tímový projekt na našej fakulte.*

**Kľúčové slová:** *plánovanie, scrum, úlohy*

## Úvod

Všade v našich životoch môžeme uviesť isté formy plánovania. Či už si plánujeme voľný deň, dovolenku, alebo aj úlohy do školy, či práce. Aj keď si to možno neuvedomujeme, plánovanie je neoddeliteľnou súčasťou nášho života.

Toto platí aj v softvérových projektoch všetkých veľkostí. Plánovanie je základným kameňom úspechu, prípadne neúspechu, takýchto projektov. Či už sú to chyby ako neplánovanie vôbec, nedostatočné plánovanie, alebo neučenie sa z predošlých chýb [3].

Metodika scrum sa snaží odstraňovať tieto problémy. Je ale tento prístup bezchybný? Môže sa nám stať, že aj pri použití agilnej metódy vývoja softvéru nám projekt zlyhá? Tieto otázky sú zvlášť dôležité v tímovom projekte na našej fakulte. Osobitne by si ich mali položiť študenti ako členovia tímov, ktoré sa riadia touto metodikou. Na úspešné dokončenie tímového projektu je potrebného veľa úsilia, ale treba si dávať pozor na mnohé zákutia, ktoré ešte neskúsených študentov čakajú.

## Čo to má byť?

Prvým krokom pri plánovaní je vedomie cieľa, t.j. čo vlastne chce robiť. Toto vedomie je potrebné mať nie len pri softvérových projektoch, ale pri hocijakej činnosti. Ak nemám túto vidinu cieľa, ako môžem robiť danú prácu? Výsledkom našej práce v tímovom projekte má byť určitý softvérový produkt, to už vieme. Aj keby sa mohlo zdať, že máme istú výhodu oproti veľkým firmám, nie je to úplne tak. Naš tím si síce vyberá z prezentovaných tém, avšak na začiatku nemáme úplnú predstavu, čo všetko má výsledný produkt obsahovať. Vo väčších firmách je požiadavka zadávaná často zákazníkom. U nás je zástupcom zákazníka pedagogický vedúci, ktorý v rámci metodiky scrum zastáva úlohu product ownera.

Product owner má v tíme riadeným metodikou scrum zabezpečovať tlmočenie požiadaviek od zákazníka. V prípade tímového projektu je pedagogický vedúci zároveň aj samotným zákazníkom, takže všetko, čo tam chce ako zákazník mať vie on sám. Aj keď má všetky tieto požiadavky u seba „v hlave,“ môže nastať problém, že si na všetky funkcie, ktoré chce mať vo výslednom produkte, nespomenie. Táto skutočnosť mi bola potvrdená iným pedagogickým vedúcim.

Na stretnutiach k tímovému projektu sa preberá veľa vecí a je teda pochopiteľné, že sa niečo zabudne povedať. Začne sa riešiť nejaká iná dôležitá funkcionality a náš product owner nám zabudne povedať niektoré veci. Alebo aj keď myslí na niečo, čo by nám na stretnutí chcel povedať, môže sa stať, že na stretnutí si už na to kvôli rôznym okolnostiam jednoducho nespomenie. Takýmto spôsobom sa môže stať, že náš pedagogický vedúci, spolu s členmi tímu stratia víziu takzvanej „big picture,“ a budú sa sústreďovať na nepodstatné, alebo menej podstatné veci.

Metodika scrum sa tento rizikový faktor snaží odstrániť pomocou product a release backlogov. Kým product backlog slúži na zozbieranie všetkých nápadov do projektu, release backlog slúži na identifikovanie konkrétnej funkcionality, ktorá má byť vo výslednom produkte. Potrebnú funkcionality môžu navrhovať rôzne zainteresované osoby, čo môže pomôcť product ownerovi spomenúť si na tie veci, ktoré chcel mať vo výslednom produkte. Ďalšou výhodou je, že sa tieto backlogy spisujú. Aj staré príslovie vraví: „Lepšie raz vidieť, ako stokrát počuť.“ Týmto sa podľa mňa zvyšuje šanca, že product owner dá naozaj všetko, čo chce vo výslednom produkte mať. Ďalšou dôležitou metódou na zvýšenie pravdepodobnosti identifikovanie všetkej funkcionality, ktorú chce mať product owner v produkte je častá komunikácia s ním. Dôležité je klásť mu otázky, na ktoré sa spoločne snažíte nájsť odpovede. Netreba sa báť dávať zdanlivo hlúpe, alebo zbytočné otázky. Aj takéto otázky môže priniesť nové a zaujímavé nápady.

Pri tímovom projekte našich rozmerov je taktiež veľmi vhodné, aby sa tieto požiadavky komunikovali v rámci celého tímu. Na začiatku náš tím prichádzal len

s hmlistou predstavou o výslednom produkte, ktorý máme používať. Náš pedagogický vedúci, podobne ako my, nemal poriadnu predstavu, čo vlastne všetko má byť vo výslednom produkte. Naše hmlisté predstavy sa síce viac-menej zhodovali, avšak nemali sme nič konkrétne. Až po istom stupni analýzy a diskutovanie sa nám podarilo dostať do štádia, kde sme mali spravený product backlog s rozumnými návrhmi, z čoho sme potom vytvorili release backlog s konečnou funkcionalitou systému.

## Úlohy, úlohy, úlohy

Samotná práca, ktorej výsledkom majú byť dokumenty, prototypy, prípadne iné výstupy je v metodike scrum zachytená v úlohách. Úlohy predstavujú konkrétne činnosti, ktoré členovia tímu vykonávajú a cez ne smerujú k výslednému produktu. Ich riadenie je preto dôležité, nakoľko nesprávnym identifikovaním úloh, alebo ich opakovaním vznikne veľká šanca nedodať požadovaný výsledok.

Scott Ambler vraví, že pri agilnom vývoji softvéru je dobré neplánovať ďaleko do budúcnosti [1]. Toto je podľa mňa v dôsledku neviditeľnosťou softvéru. Keďže nevieme presne určiť, ako bude softvér vyzeráť a čo všetko bude v konečnom dôsledku obsahovať, nevieme presne určiť čas, ktorý je potrebný na jeho zhotovenie. Metodika scrum zabezpečuje toto krátkodobé plánovanie cez jednotlivé šprinty a úlohy v nich vykonávané. Je vhodné, keď stanovíme menším, ľahšie merateľnejším úlohám presný čas, do kedy sa majú ukončiť. Avšak netreba zanevrieť na naplánovanie hlavných častí celého softvérového projektu ako celku. Myslím si, že je nutné mať aj „vidinu konca,“ a že je potrebné vedieť aspoň približný čas odovzdania funkčnej finálnej verzie.

Pri metodike scrum sa s úlohami pracuje na začiatku nového šprintu, kde sa vyberie nedokončená položka z product backlogu a rozdelí sa na konkrétne úlohy. Tu ale nastáva problém: ako identifikovať úlohy? Máme úlohu nazvať podľa položky backlogu? Koľkým členom tímu ju pridelieme? Charles Bradley, ktorý pracuje ako školiteľ metodiky scrum, nám dáva dve rady pri tejto činnosti [2]: robiť úlohy čo najviac atomické (t.j. nedeliteľné) a zabezpečiť, aby im každý člen tímu porozumel. Tieto dve rady dosť spolu súvisia. Nedeliteľnosť úlohy zabezpečí jej transparentnosť, čím jej členovia tímu lepšie porozumejú. V našom tíme sa často stávalo, že keď sme si rozdelili úlohy a zapísali ich do zápisnice, pridelený členovia sa pýtali, čo má byť tá úloha vlastne zač? Treba preto vyberať dostatočne (ale nie príliš malé) úlohy a dať im dobrý opis, v ktorom bude napísaný názov, trvanie, a čo všetko sa vo vypracovaní požaduje.

Ďalšou veľmi dôležitou súčasťou riadenia úloh v metodike scrum je ich pridelovanie konkrétnym členom tímu na vypracovanie. Scott Ambler navrhuje, aby si každý člen tímu sám vybral prácu, na ktorej chce robiť [1]. Motivácia za tým je, že členovia si budú vyberať úlohy, pri ktorých vedia, že majú dostatočné schopnosti na ich úspešné dokončenie. Vo firemnom svete by som videl dobré uplatnenie tohto princípu, avšak v našom tímovom projekte by to nemalo úplne takto fungovať. Podľa mňa by sme mali nájsť rovnováhu medzi prístupom, kde si členovia sami vyberajú úlohy, a kde sa im úlohy len prideliujú. Treba dbať aj na záťaž jednotlivých členov tímu a snažiť sa, aby bolo celkové úsilie za dlhšie časové obdobie (semester, alebo pol semestra) približne rovnomerne rozdelené medzi všetkých členov tímu. Takto nebude nikto znevýhodňovaný, ani zvýhodňovaný a každý priloží rovnaké úsilie k celkovému dielu. Na druhej strane je lepšie, keď sa

niektoré činnosti ako je správa webovej stránky tímu, alebo správa nástroja pre manažment verzií nechajú na jedného člena tímu počas celého trvania projektu.

Pri projekte ako celku je dobré aj vidieť jednotlivé úlohy zoradené časovo za sebou. Na to slúžia rôzne diagramy, z ktorých jeden je Ganttov diagram. V ňom sú zobrazené všetky úlohy s časovým vymedzením, ako aj nadväznosťami na iné úlohy. Súhlasím s tým, že toto zobrazenie poskytuje dobrý prehľad o úlohách a ich nadväznosti, ale nakoľko sa tam zobrazuje každá jedna (aj malá) úloha, už v menších projektoch (ako je napríklad tímový projekt na našej škole) tam môže vzniknúť zmes čiar a obdĺžnikov, čím sa stratí výpovedná hodnota diagramu. Vizualizácia úloh je podľa mňa dobrý zlepšením prehľadu o plánovaní, avšak treba nájsť správny spôsob zobrazenia. Podobný názor má aj Scott Ambler, i keď on tvrdí, že existujú dôkazy, že Ganttov diagram má malú hodnotu v agilných projektoch [1]. Ako výnimku spomína Ganttov diagram vyššej úrovne, ktorý podľa neho dáva prehľad o celkových závislostiach medzi úlohami.

## Obzriet sa späť

Pri plánovaní myslíme často iba na budúcnosť: ako a kedy sa niečo stane, ako by to malo vyzeráť a podobne. Tu je však veľmi dôležité obzrieť sa dozadu, na minulosť, a zistiť, aké chyby sa v minulosti urobili. Takto sa dokážeme z minulosti poučiť a upraviť riadenie k lepšiemu [3].

Na tímovom projekte na našej fakulte sa každý rok združia skupiny študentov a vytvoria jednotlivé tímy. Drvivá väčšina z nich nemá praktické skúsenosti s prácou v tíme, nie to ešte s metodikou scrum. Pre neskúsený tím, ako je aj ten náš, sa môže stať, že pri zlej práci na projekte nenastane zmena z lepšiemu. Podľa Jeffa Sutherlanda je veľmi dôležité sa o obzrieť späť a zhodnotiť šprint [4]. Retrospektíva šprintu by mala nasledovať po každom šprinte a má sa zamerať na samotný proces riadenia a nie na produkt. Sutherland však poznamenáva, že táto činnosť sa často vynecháva, čo považuje za neakceptovateľné. Z doterajšej skúsenosti s prácou na tímovom projekte musím žiaľ potvrdiť jeho tvrdenie. Aj náš tím sa viac sústreďuje na produkt a jednotlivé úlohy, ako na proces riadenia a jeho zlepšenie.

Toto nie je ale príčinou jednej konkrétnej chyby, ale z môjho pohľadu je to niekoľko dôvodov. Ako som spomínal, náš tím veľmi málo pozerá na proces riadenia. Vidina dokončenia produktu je omnoho silnejšia ako cesta k nemu. Doteraz sme boli často zvyknutý, že sme vypracovali zadanie, odovzdali ho a bol koniec. Teraz sa však musíme sústrediť aj na riadenie procesu vypracovávania nášho projektu. Nové prostredie tímu dáva neskúseným spolupracovníkom veľa možností ako stroskotať.

Ďalším dôvodom nepozerania na proces riadenia je v našom tíme čas. Na stretnutiach s pedagogickým vedúcim sa preberie mnoho záležitostí, niekedy menej a inokedy viac potrebných. Často sa nám stáva, že kým sa všetko prediskutuje, vlastne nám už ani nezostane čas na revíziu procesu riadenia. Takisto je iným dôvodom nezájem členov tímu. V tomto si myslím, že nie sme iný od ostatných tímov. Každý sa určite stretá s istou nechuťou voči metodike scrum a kladie si otázku: „Na čo nám to vlastne je?“

Aby sme sa teda vyhli nedokončenia projektu, alebo slabého výsledku, je potrebné vykonávať retrospektívu šprintu a identifikovať dobré a zlé praktiky pri riadení. Identifikovaním chýb ich môžem buď úplne odstrániť, prípadne aspoň zmierniť, a celkovo

vylepšiť riadenie projektu. Vyzdvihnutím dobrých praktík ich zas môžeme opakovane použiť a prípadne ich ešte o niečo vylepšiť. Zlepšenie riadenia je podstatná časť pre metodiku scrum a jeho uplatnenie môže viesť k vylepšeniu komunikácie v rámci tímu, ako aj súdržnosti a spolupráce, čo v konečnom dôsledku znamená vytvorenie kvalitnejšieho výsledného produktu.

Dosiahnuť to však nie je ľahké. Je na to potrebný dostatok času a úsilia. Je tu potrebné si najskôr vyskúšať praktiky v riadení, nakoľko pokiaľ nemáme v našom tíme odskúšané, či niečo ide dobre, alebo zle, nevieme sa z toho poučiť, presne tak, ako vraví príslovie: „Človek sa najlepšie učí z vlastných chýb.“ Konkrétne u tímového projektu na našej fakulte to platí ešte viac, nakoľko v každom tíme sú jedineční ľudia, z ktorých spojením vzniknú jedinečné tímy. Pre jeden tím môže byť jedna praktika dobrá, ale na pre druhý tím to môže byť katastrofa. Treba teda empiricky odskúšať, čo daný tím dokáže a ako dokáže správne fungovať. Pri tomto je veľmi dôležitá podpora od pedagogického vedúceho. Aj keď samotná zodpovednosť za riadenie je na členoch tímu, pedagogický vedúci by mal celý tím usmerňovať. Iniciatívu by mali vyjadriť aj členovia tímu, a to tak, že sa budú pýtať otázky, ako majú riešiť konkrétne situácie.

Pri pozeraní na predošlé nedostatky sa však máme snažiť zmeniť tím ako taký, a nie scrum ako proces riadenia. Sutherland to považuje za závažnú chybu [4]. Treba si uvedomiť, že aj keď metodika scrum možno nie je dokonalá, nesmieme sa snažiť ju prispôbovať, ale máme zmeniť samých seba, náš tím.

## Záver

Ako je možné vidieť, metodika scrum má stále dosť úskalí, aj napriek tomu, že sa snaží byť veľmi dobrým rámcom pre tvorbu softvérových systémov. Asi najväčšími úskaliaми sú tvorba požiadaviek s product owner-om, riadenie konkrétnych úloh, a učenie sa z vlastných chýb a nedostatkov.

Plánovanie je veľmi dôležitou súčasťou softvérového projektu, a preto je potrebné venovať mu dostatočne veľa času, aby sa mohol projekt úspešne dokončiť. Toto platí ešte viac na našej fakulte, kde sa v inžinierskom štúdiu študenti prvýkrát stretávajú s tímovou prácou a metodikou scrum.

## Použitá literatúra

1. Ambler, S. W.: Agile Project Planning Tips. Dostupné na internete: <http://www.ambysoft.com/essays/agileProjectPlanning.html>, 25.11.2012
2. Bradley, Ch.: Sprint Tasking Tips. Dostupné na internete: <http://www.scrumcrazy.com/Sprint+Tasking+Tips>, 25.11.2012
3. McConnel, S.: Construx Software: The Nine Deadly Sins of Project Planning. *IEEE Software* (2001) 5-7.
4. Sutherland, J.: The Scrum Handbook. Scrum Training Institute Press, 2010.

## **Annotation**

### *How to not plan for failure*

*Different methods for planning schedules are nowadays used not only in schools, but also in various companies or personal lives. It also plays a large role in software projects for both individuals and teams of people. Everyone tries to do their best to plan the project well, although the extent to which the project is well planned is hardly measurable. Custom software is created for many years and created several advices which managers should keep in mind during planning, so that their project will not fail. The scrum methodology as one of the agile software development methods, strives to provide solutions to improve and clarify the planning of software projects. After a few years of experience with this methodology, there are tips on how to improve such a little planning. It is important, however, to pay attention and try to follow these tips. This is especially important for inexperienced students, who for the first time face teamwork in the team project subject to our faculty.*