

SLNEČNÉ ALEBO DIOPTICKÉ OKULIARE PRE SLEDOVANIE AGILNÝCH PROJEKTOV?

Jeden pohľad nemusí byť vždy ten správny.

Daniel Drahoň

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

Abstrakt. Monitorovanie softvérového projektu je dôležitou súčasťou pri jeho vývoji, má vplyv na plánovanie a riadenie projektu, zároveň umožňuje včas identifikovať problémy v tíme. Táto esej opisuje agilný prístup a samotné monitorovanie v agilných projektoch. Esej sa zameriava na opis troch základných postupov, a to burn grafy (angl. burn charts), kumulatívne diagramy (angl. cumulative flow diagrams) a metódu bilancovania výroby (angl. line of balance). Tieto metódy poskytujú informácie o aktuálnom stave práce na projekte, avšak rozličným spôsobom. Ďalšou časťou je monitorovanie prebiehajúcej práce, akou metrikou túto činnosť vyjadriť a aký má význam v agilných projektoch. Zámerom eseje je vzájomné porovnanie rôznych metód, rozdiely a výhody, ktoré poskytujú, s dôrazom na reálne využitie.

Kľúčové slová: monitorovanie, agilný, burn grafy, kumulatívne diagramy, metóda bilancovania výroby, čiastočne dokončená práca

Plávaj projekt plávaj, tým správnym smerom

Starovekí moreplavci mali dva základné problémy. Prvým z nich bolo určenie zemepisnej výšky, teda ich poloha na mape zo severu na juh. Druhým bolo určenie zemepisnej dĺžky, čiže pozície z východu na západ. Zistenie zemepisnej výšky pozorovaním hviezd Polárka, ktorá ukazovala vždy na sever, bolo pomerne jednoduché, avšak zmerať zemepisnú dĺžku bolo oveľa zložitejšie. Bohužiaľ, v tej dobe nemali k dispozícii presné prístroje ani GPS. Používali pritom iba hodinky alebo chronometre a námorník musel

svoju pozíciu vypočítať, čo znamenalo odhadnúť aktuálnu polohu pomocou dedukcie z poslednej známej pozície použitím údajov o rýchlosti, čase a kurze. Napríklad merač rýchlosti ukazoval 20 uzlov za hodinu, ak však loď čelila prudkým vlnám a vetru mohla byť skutočná rýchlosť len 13 uzlov za hodinu. Preto museli byť zarátané všetky faktory súvisiace s výpočtom, inak bola zemepisná šírka určená nesprávne.

Sledovanie vývoja projektu je veľmi podobné s navigáciou lode výpočtom. Rozdiel je však ten, že sa nesnažíme určiť jeho „polohu“ ale jeho aktuálny stav. Chceme aby sa náš projekt dostal priamo z bodu A do bodu B. Problémy v tomto prípade znamenajú zmeny v požiadavkách, rozdiel produktivity v tíme a taktiež aj chyby v tom, ako ho meriame.

Agilný svet projektov

Predpokladom pri tradičných projektoch, vytváraných napríklad vodopádovým modelom, je známy rozsah celého projektu, ktorý môže byť určený na jeho začiatku. Tieto projekty majú dobre stanovený presný plán, čím sa znižuje riziko ich neúspechu. Práca tímu má vo väčšine prípadov lineárny charakter. Nevýhodou je však neschopnosť prispôbiť sa zmenám v priebehu riešenia projektu.

Ako sa menili požiadavky na softvér, menili sa aj formy jeho vývoja, čo podnietilo vznik tzv. agilných metód. Slovo agilný pochádza z latinského slova *agilis* a v preklade znamená čulý, bystrý, horlivý. Agilné metódy majú svoje charakteristické vlastnosti. Rozsah projektu je na začiatku definovaný na vysokej úrovni abstrakcie a v priebehu riešenia sa konkretizuje. Zmeny v projekte sú očakávané a časté, pretože zákazník a používatelia sú oprávnení meniť svoje požiadavky. Veľké množstvo používateľov si totiž uvedomí čo naozaj potrebujú, až keď majú možnosť vyskúšať softvér „na vlastnej koži“. Vývoj preto prebieha iteratívne, v malých inkrementálnych vydaniach. Vo februári roku 2001 bol spísaný *Agilný manifest*, ktorý obsahoval tieto body [2]:

- Jednotlivci a interakcie majú prednosť pred procesmi a nástrojmi.
- Funkčný softvér má prednosť pred zložitou dokumentáciou.
- Spolupráca so zákazníkom má prednosť pred zmluvnými rokovaniami.
- Prispôsobenie sa zmenám má prednosť pred striktným dodržiavaním plánu.

Súčasťou tohto manifestu boli aj princípy agilného projektu. Medzi základné patrí aktívna účasť používateľa na projekte, pričom tím musí mať možnosť rozhodovať sa. Zároveň sa dôraz kladie na funkčnosť softvéru, ktorý by mal byť dodaný v pravidelných intervaloch, preto je potrebné testovať skoro a často.

Agilné metódy sú veľmi vhodné pri projektoch s menším počtom členov v tíme a krátkou predpokladanou dobou vývoja. Medzi dve dnes najpoužívanejšie agilné metódy patria Scrum a XP (*angl. Extreme Programming*).

Agilné metódy poskytujú vývojárom väčšiu voľnosť v priebehu ich práce. Táto voľnosť im umožňuje využívať väčšiu kreativitu a spája ich osobný prínos spolu s projektom. Bez kontrolného procesu by táto voľnosť mohla mať na projekt katastrofálny dopad. Je preto odporúčané, ba dokonca nevyhnutné monitorovať ich činnosť a sledovať, či vytváraný produkt nemá skryté problémy.

Meter, vodováha a iné metriky na meranie projektu

Monitorovanie a meranie je dnes dôležitou súčasťou pri vývoji každého softvéru. Výnimkou nie sú ani projekty, ktoré sú vytvárané agilnými metódami. Je však potrebné odpovedať na dve základné otázky, konkrétne: *Prečo monitorovať? A čo je možné merať?*

Monitorovanie má viacero dôvodov [1]. Poskytuje kvantifikovateľný prehľad pre podporu rozhodovania. Slúži na vytváranie presnejších odhadov pre plánovanie, taktiež pri vyhodnocovaní vplyvu nových metód a podporných prostriedkov. V neposlednom rade pomáha pri identifikácii problémov. Bez monitorovania by sme nemohli poznať, či projekt napreduje správnym smerom, koľko času ešte potrvá jeho vývoj a či sú prostriedky a zdroje dostatočné na jeho tvorbu.

Odpoveď na druhú otázku je trochu komplikovanejšia. Existuje totiž viacero metrík a pohľadov na meranie projektu. V zásade tieto metriky možno rozdeliť do troch hlavných kategórií [1]:

1. Metriky pre produkt: veľkosť, rozsah, modularita, spoľahlivosť, zložitosť, chyby.
2. Metriky pre proces: úsilie, náklady a čas vynaložený na vývoj projektu.
3. Metriky pre zdroje: charakteristika personálu (produktivita, skúsenosti).

Určiť, ktorá metrika ponúka najkomplexnejší pohľad na projekt nie je vôbec jednoduché. Ideálne by bolo, ak by existovala metóda, ktorá by dokázala vyhodnotiť všetky uvedené metriky a jej výstupom by bol zoznam informácií o stave projektu. To samozrejme nie je reálne v praxi a preto sa musíme na problém pozrieť z iného uhla pohľadu.

Definujme nový pojem *rýchlosť projektu* [2]. Ako fyzikálna veličina rýchlosť označuje prejdenú vzdialenosť za určitý čas. Ak sa vrátíme naspäť k úvodu, rýchlosť lode bola vyjadrená v uzloch za hodinu. V prípade agilných projektov rýchlosť môžeme chápať ako kvantum vykonanej práce za istú dobu. Čas bude reprezentovaný ako jeden deň alebo jedna iterácia. Vykonanú prácu by bolo najjednoduchšie charakterizovať ako počet riadkov programového kódu. Tu však nastáva problém, čo ak sa vyprodukuje kopec riadkov kódu, ktorý ani nebude funkčný? Preto je potrebné vykonanú prácu zdefinovať iným spôsobom, napríklad ako počet dokončených a otestovaných funkcií. Funkcie v tomto prípade znamenajú používateľské príbehy (*angl. user stories*), prípady použitia (*angl. use cases*), alebo iné funkcionálne, či nefunkcionálne požiadavky. Používateľský príbeh je stručný popis funkcie z pohľadu používateľa alebo zákazníka.

Uvažujme vo svete agilných projektov. Na začiatku je vypracovaný plán projektu, ktorý hovorí približne toto: „Najbližšie štyri mesiace, v priebehu ôsmich dvojtyždňových iterácií budeme musieť dokončiť približne 100 funkcií.“ Pre tím, ktorý urobil 15 funkcií v jednej iterácii je rýchlosť 15 funkcií za iteráciu. Za predpokladu, že rozsah projektu sa nezmenil, tím má kratšiu vzdialenosť do cieľa a teda menej práce na jeho dokončenie.

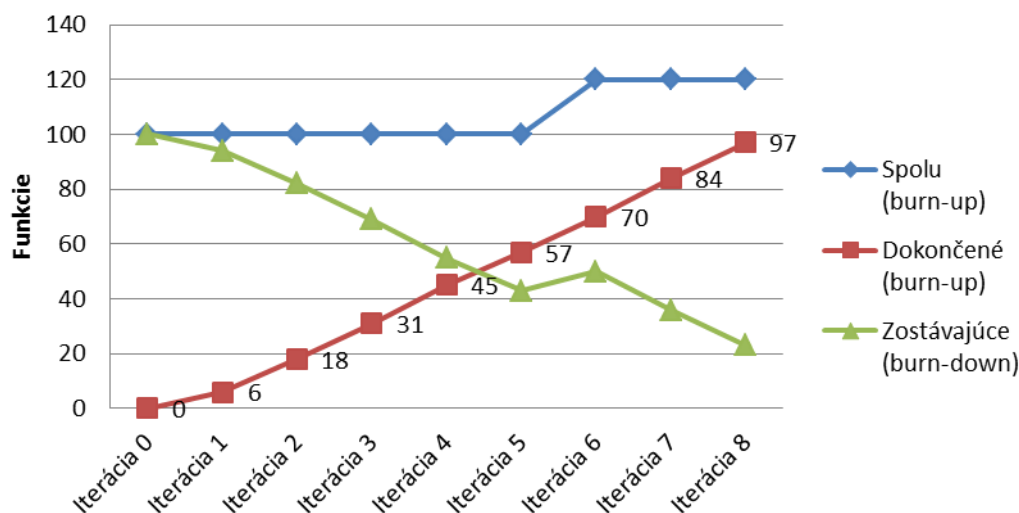
Základné techniky, ktoré sa používajú na monitorovanie agilného projektu poskytujú informácie o stave a rýchlosti vývoja daného projektu. Sú však tieto metódy dostatočne účinné a presné? Dokáže vôbec jedna metóda opísať celý projekt?

Oheň na palube, horia našťastie len grafy

Jedným zo štandardných spôsobov monitorovania agilných projektov sú tzv. *burn charts* [3]. Existujú dva typy týchto grafov, a to *burn-up graf* a *burn-down graf*. Výhodou týchto grafov je ich jednoduchá reprezentácia, preto sa dajú ľahko pochopiť, tak pre vývojárov, ako aj pre sponzorov projektu a manažérov.

Burn-up graf zobrazuje rastúce množstvo dokončenej funkcionality v projekte od závislosti v čase. V grafe sú zobrazené len tie funkcie, ktoré fungujú a boli otestované, vďaka tomu je výsledný graf viac presnejší a dôveryhodnejšie vyjadruje stav projektu.

Burn-down graf poskytuje podobné informácie spôsobom, koľko funkcií zostáva na dokončenie, zároveň tiež zobrazuje celkový postup projektu. Ak je pridaná práca do projektu, krivka vzrastie smerom nahor, ak je práca vylúčená, krivka klesne nadol. Navyše, extrapoláciou z grafu je možné určiť približný dátum ukončenia projektu. Po každej iterácii sa rozsah a odhady spresňujú.



Obr. 1 Burn-up a burn-down graf. Stav projektu monitorujú pomocou dokončených funkcií.

Na **Obr. 1** sú vykreslené dva grafy. Jeden z nich je burn-up graf, kde červená krivka zobrazuje počet úspešne dokončených a implementovaných funkcií. Modrá krivka značí počet všetkých naplánovaných funkcií v projekte. Je jasne vidieť, že po piatej iterácii bolo do projektu pridaných ďalších 20 funkcií. Druhý burn-down graf obsahuje zelenú krivku, ktorá udáva počet zostávajúcich funkcií, potrebných na dokončenie.

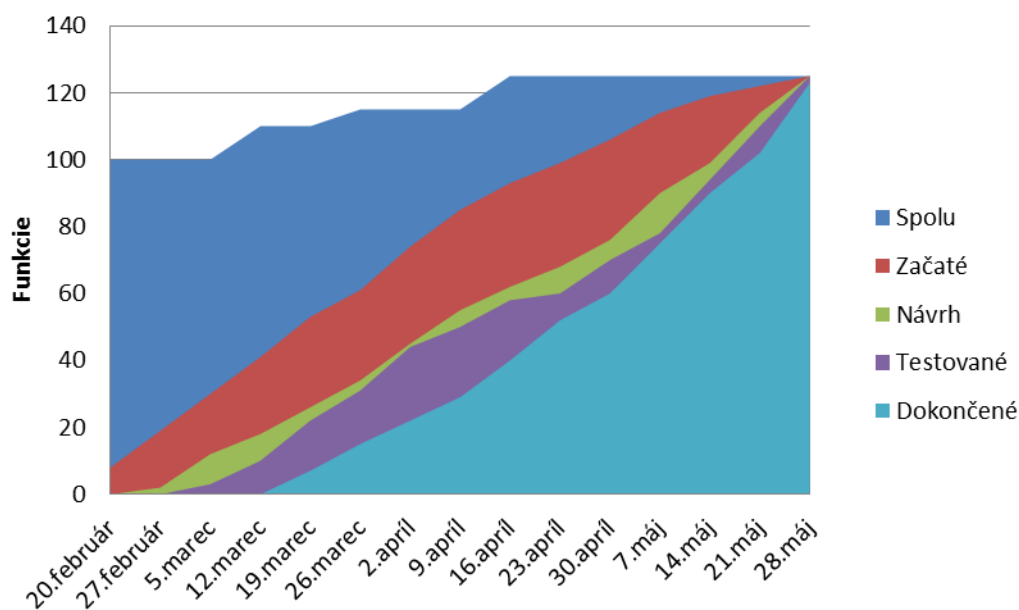
Ktorý z diagramov je výhodnejšie používať? Odpoveď je zrejme u každého individuálna. Hlavnou príčinou je psychologický faktor, ako človek dokáže vnímať pokrok. Niektorí ľudia dávajú prednosť pohybu smerom nahor, tak ako horolezci, vystupujúci na vrchol. Iní uprednostňujú pohyb nadol, kedy sa povinnosti a práca redukovujú, až pokým nezostane žiadna úloha. Oba grafy jasne ukazujú stav a rýchlosť vývoja projektu oproti plánu a poskytujú viditeľný údaj o tom, či tím stihne dokončiť projekt načas alebo nie.

Výhoda jednoduchého zobrazovania je zároveň najväčšou nevýhodou týchto grafov. Keďže monitorujú vždy len kompletne dokončenú prácu, je nesmierne náročné a prakticky nemožné určiť prebiehajúcu prácu (*angl. work in process, skr. WIP*). Ako však reprezentovať množstvo čiastočne dokončenej práce? Prináša meranie WIP iba výhody alebo sú pritom ukryté aj určité riziká?

Keď došiel benzín, použijeme kumulatívne diagramy

Kumulatívne diagramy (angl. cumulative flow diagrams, skr. CFD) predstavujú ďalšiu metódu monitorovania agilného projektu [3]. CFD sú odvodené zo základu burn-up grafov, rovnako dokážu vykresľovať počet dokončených funkcií projektu v priebehu času. Navyše však poskytujú zobrazenie prebiehajúcej práce, a tak zabezpečujú vyššiu transparentnosť celého projektu. Ich tvorba prebieha tým spôsobom, že sa spočítajú všetky funkcie, ktoré vo vývoji dosiahli určitý stav v danom čase a vykreslia sa v grafe.

Na **Obr. 2** sú všetky funkcie rozdelené do 5 častí, tri z nich predstavujú WIP (začaté, návrh, testované). Údaje označené menovkou „Spolu“ určujú koľko funkcií malo byť v projekte vytvorených. Pohľadom na obrázok môžeme zistiť, že testovanie zaberalo viac času ako návrh, z čoho by sa dalo usúdiť, že tím kládol väčší dôraz na testovanie, prípadne, že pri testovaní nastali isté problémy. Ďalej je možné vidieť, ako sa WIP postupne hromadila, približne do polovice vývoja projektu a následne začala ubúdať.



Obr. 2 Kumulatívny diagram (CFD). Dokáže zobrazit' celkový postup vývoja projektu spolu s WIP [5].

Vďaka sledovaniu WIP, ktoré táto metóda umožňuje, je možné dôkladnejšie analyzovať prácu na projekte a problémy s ním súvisiace. Na druhej strane čitateľnosť CFD grafov vzhľadom na odhad skončenia projektu je oproti burn grafom horšia.

Blížiac sa búrka

Existujú tri hlavné problémy, ktoré súvisia s meraním WIP [2]. Prvým je samotné určenie tejto veličiny a ohodnotenie jej rozsahu. Napríklad programátor implementoval funkciu, ktorá doposiaľ nebola otestovaná. Na druhej strane môže byť rovnaká funkcia, ktorá je čiastočne naprogramovaná a čiastočne otestovaná. Ktorá z týchto dvoch funkcií obsahuje viac nedokončenej práce? Na túto otázku zrejme neexistuje konkrétna odpoveď. Vieme jednoznačne určiť, kedy bola práca spustená a kedy bola dokončená. Priestor, ktorý sa nachádza medzi týmito dvoma stavmi, je pomerne ťažké definovať.

Druhý problém predstavuje základná filozofia agilných metód. Zákazník požaduje funkčný softvér, ktorý je dobre otestovaný. Neúplné funkcie v projekte akoby poukazovali na neschopnosť vývojárov, keďže neboli dokončené v stanovenom čase, a tým narušujú dôveru medzi zákazníkom a tímom.

Tretím, nemenej dôležitým problémom, je hromadenie WIP v procese vývoja agilného projektu. Čím väčšie množstvo WIP sa vytvorí, oproti dokončenej funkcionalite, tým ťažšie je získať spätnú väzbu o tom, čo sa práve vyvíja. V konečnom dôsledku môže mať tím menšiu produktivitu, keďže programátori nie sú nútení okamžite transformovať svoje myšlienky do fungujúceho softvéru [2].

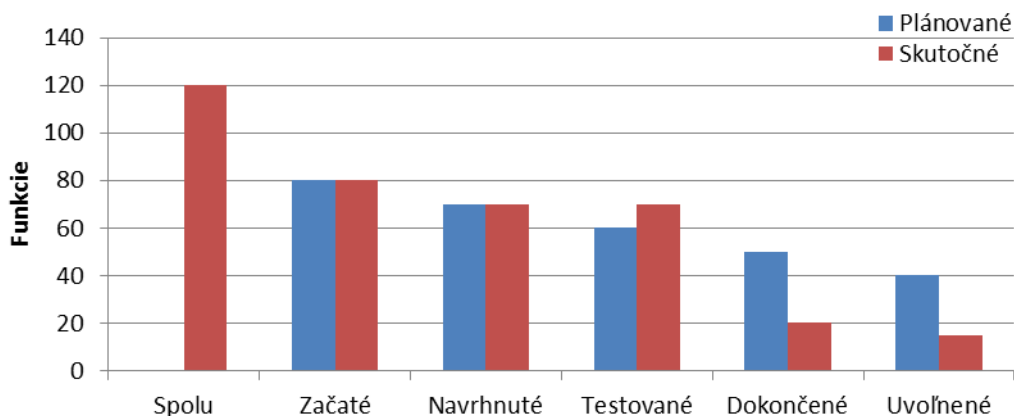
Ak je teda obtiažne a zároveň sa zdá byť aj kontraproduktívne vyjadriť získanú hodnotu WIP, mali by sme ju úplne ignorovať? Je pravda, že pri plánovaní agilných projektov sa kladie dôraz na funkčnosť a po každej iterácii by WIP mala byť úplne odstránená. Faktom však zostáva, že WIP sa nachádza v každej funkcionalite, ktorú tím začal vytvárať a ktorá ešte nebola dokončená.

Treba si ďalej uvedomiť význam informácií, ktoré prináša monitorovanie projektu spolu s WIP. Po prvé, dokážeme presnejšie určiť stav projektu a odhaliť ťažkosti v ňom ukryté. To umožňuje vývojárom a manažérom podstatne skôr reagovať na blížiace sa problémy. Po druhé, čiastočne dokončená funkcia má väčšiu hodnotu ako funkcia, ktorá ešte nebola začatá. Klasický agilný prístup k monitorovaniu, považuje obidve funkcie za rovnocenné, čo podľa môjho názoru nie je celkom pravda. V neposlednom rade je treba povedať, že monitorovanie tejto veličiny nemusí nutne podnecovať tím k jej vytváraniu.

Na scénu prichádza bilancovanie

Doteraz opísané metódy mali spoločnú vlastnosť, plánovať počet dokončených funkcií pre celý projekt. V prípade, ak projekt rozdelíme na menšie časti a pre každú z nich vytvoríme plán na dokončenie, budeme schopní pozorovať stav práce vzhľadom k danému plánu.

Line of balance (skr. *LOB*) je metóda bilancovania, ktorá sa používa v sériovej výrobe na riadenie priebehu výrobného procesu [4]. Princíp tejto metódy je ten, že umožňuje porovnať naplánovaný a dokončený /resp. skutočný/ stav výrobkov v určitom bode. Aplikovaním na agilný projekt možno týmto spôsobom monitorovať priebeh jeho vývoja. Najprv sa určia kontrolné body projektu. Tieto body musia mať dobre stanovené výstupné kritérium, na ktorom je možné merať WIP a dokončenú prácu. Ak napríklad monitorujeme projekt, v ktorom sú funkcie definované ako používateľské príbehy, potom jeden kontrolný bod môže zodpovedať jednému zo stavov životného cyklu používateľského príbehu (začatie, návrh, testovanie, dokončenie, uvoľnenie).



Obr. 3 LOB diagram ukazuje koľko funkcií malo prejsť kontrolným bodom a koľko skutočne prešlo [4].

Graf na **Obr. 3** zobrazuje stav vývoja projektu v určitom období. Cieľom je vytvoriť 120 funkčných používateľských príbehov (funkcií), ktoré označuje stĺpec s názvom „Spolu“. Pre ďalšie kontrolné body graf zobrazuje 2 stĺpce: „Plánované“ a „Skutočné“. Tieto stĺpce určujú koľko funkcií malo byť pôvodne v každom kontrolnom bode a koľko z nich ním v skutočnosti prešlo. Pozorovaním údajov možno zistiť, že začaté a navrhnuté funkcie sú v súlade s plánom. Testovanie je dokonca v predstihu. Problém však nastáva pri dokončení a uvoľnení, pretože v týchto kontrolných bodoch zaostávajú vytvorené funkcie za harmonogramom. Graf neukazuje príčinu problému, avšak je zrejme, že pomalé tempo dokončovania ohrozuje ďalšie vydanie a tým pádom celý projekt.

Tento spôsob monitorovania má ale jednu menšiu nevýhodu, a tou je vyjadrenie stavu projektu v jednom časovom bode. Vieme síce presne povedať kde sa projekt nachádza práve teraz, no je ťažké určiť ako sa bude ďalej vyvíjať, prípadne aký má doterajší progres. Preto, ak chceme vyjadriť časovú súvislosť, je potrebné doplniť do grafu tretí rozmer, v ktorom bude časová os a na nej zachytený plán projektu. Tak získame lepší prehľad o celkovom stave. Bude tiež jednoduchšie predpovedať kam projekt smeruje.

Nespornou prednosťou je skutočne veľmi detailné sledovanie stavu projektu a mohli by sa zdať, že metóda bilancovania nemá žiadne ďalšie záporné vlastnosti. Nezabúdajme však, že sa nachádzame v rovine agilných metód a musíme preto rozlíšiť dva hlavné pojmy: plánovanie a monitorovanie prebiehajúcej práce.

Plánovanie WIP prináša so sebou viaceré riziká, pretože plánovaním pripúšťame a dokonca určujeme, aké množstvo práce bude hotové a koľko z nej zostane nedokončené. Ako už bolo spomenuté, nedokončená práca je demotivujúcim faktorom pre tím a zároveň ohrozuje dôveru so zákazníkom, ktorý očakáva fungujúci softvér. Druhý pojem je monitorovanie WIP, ktoré naozaj ponúka podrobnejší pohľad na projekt a umožňuje včasnejšiu identifikáciu problémov.

Môže teda metóda LOB fungovať bez plánovania? Odpoveď je zrejme záporná, pretože LOB podmieňuje monitorovanie práve potrebou plánovať WIP. Je to začarovaný kruh, z ktorého sa pomerne ťažko hľadá východisko. Možným riešením je vôbec neplánovať WIP a tým pádom dostaneme podobný výstup ako poskytuje CFD. Konkrétne riešenie by mohla priniesť odborná práca venovaná výskumu tejto problematiky.

Záver

Pokúsím sa odpovedať na otázku z názvu tejto eseji, teda ako detailne je vhodné monitorovať agilné projekty. Očakávanie manažérov a zákazníkov je získať rýchle, presné a hlavne jednoduché informácie o stave projektu. Ich požiadavkami sú vedieť kde sa projekt nachádza a kedy bude dokončený. Tomu presne zodpovedajú burn grafy, ktoré poskytujú ľahko pochopiteľný a čitateľný výstup, pritom je nenáročné ich vytvoriť.

Napriek tomu, že množstvo odborníkov v oblasti agilného vývoja odporúča zohľadňovať vo výstupe merania iba dokončenú prácu, predstavuje monitorovanie prebiehajúcej práce oveľa vážnejší problém. Kumulatívne diagramy sú schopné zobrazíť práve prebiehajúcu prácu, avšak nedokážu jasne vyjadriť koľko práce je potrebné na splnenie celého projektu. Oproti tomu, LOB metóda umožňuje veľmi podrobne sledovať dokončenú aj prebiehajúcu prácu vzhľadom k vytvorenému plánu. Tieto metódy sú preto vhodným kandidátom pre vodcov tímov a taktiež pre rozsiahlejšie projekty, kde je potrebné mať dôkladný prehľad nielen o stave ale hlavne o problémoch v projekte.

Vybrať však jeden konkrétny prístup k monitorovaniu a označiť ho za ideál nie je možné, pretože rozličné situácie vyžadujú rozličný uhol pohľadu.

Použitá literatúra

1. Bieliková, M.: *Softvérové inžinierstvo - Princípy a manažment*. Slovenská technická univerzita, Bratislava, 2000.
2. Cohn, M., Highsmith, J., Benefield, G.: *Agile estimating and planning*. Prentice Hall Professional Technical Reference. Upper Saddle River, 2006
3. Griffiths, M., Cabri, A.: *Earned Value and Agile Reporting*. Agile Conference 2006. Minneapolis, 2006, s. 17-22
4. Miranda, E., Bourque, P.: *Agile monitoring using the line of balance*. In: *Journal of Systems and Software*, vol. 83, 2010, s. 1205-1215.
5. Schragenheim, E., Anderson, D.J.: *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Prentice Hall, PTR, New York, 2003.

Annotation

Sunglasses or spectacles for monitoring agile projects?

Monitoring of a software project is a very important part of its development. It has influence on the planning and management of the project and at the same time enables early identification of team problems. This essay describes agile approaches and also monitoring of agile projects. The essay explains three primary methods: burn charts, cumulative flow diagrams and line of balance. These methods deliver information about actual project work status by using different approaches. In the next part is described monitoring of work in process, how this metrics express and why they are important in agile projects. The goal of the essay is the comparison of methods mentioned above, their differences and advantages and their practical application.