

PRIJATÍM ZÁVÄZKU K TÍMOVEJ SYNERGII

*Nemaj strach prijať záväzok, je to práve to, čo ti
pomáha napredovať.*

Pavol Škvarenina

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

Abstrakt. V predkladanej eseji sa venujem problematike plánovania iterácií v rámci agilných metodík vývoja. V skratke opisujem dva základné typy plánovania iterácie a to plánovanie na základe rýchlosti vývoja a plánovanie na základe prijatých záväzkov. Následne rozoberám vhodnosť jednotlivých metód, pričom pravdivosť svojich názorov sa snažím dokázať podložením relevantných skúseností a znalostí iných autorov. Z konfrontácie mi pomerne jasne vychádza, že metóda plánovanie iterácie pomocou prijatých záväzkov lepšie propaguje myšlienky agilného vývoja. Prijatím záväzku voči všetkým zainteresovaným dokáže vývojársky tím budovať vlastnú motiváciu, dôveru v samých seba a vo svoj tím, vďaka čomu vniká naprieč vývojárskym tímom efekt synergie, ktorý umožňuje vyvíjať rýchlejšie, profesionálnejšie a kvalitnejšie.

Kľúčové slová: plánovanie iterácie, agilné plánovanie, plánovanie na základe rýchlosti vývoja, plánovanie na základe prijatých záväzkov

Úvod

Plánovanie iterácie predstavuje dôležitý proces, ktorý prebieha pred začatím každej iterácie. Plánovacieho procesu sa zúčastňuje celý vývojársky tím spolu s poverenou osobou od zákazníka, ktorá zastupuje potreby klienta. Proces prebieha formou mítingu v rámci ktorého zástupca zákazníka alebo samotný tím určí cieľ nasledujúcej iterácie. Vývojársky tím následne plánuje pomocou rôznych metód rozsah a postup práce potrebnej na splnenie určeného cieľa.

Plánovanie iterácie

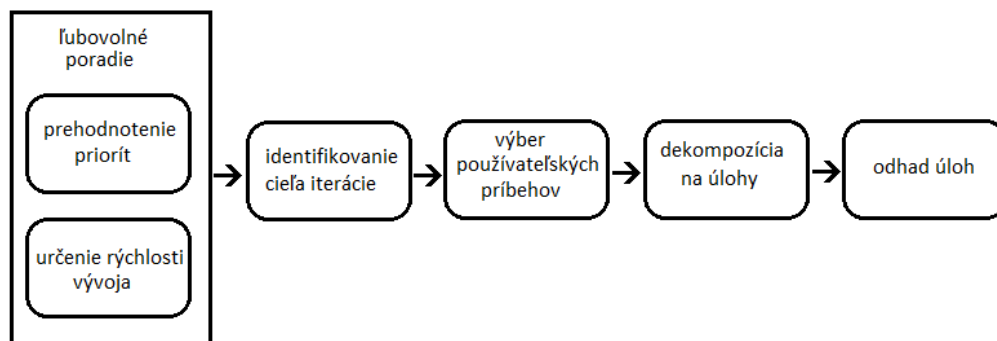
Z pohľadu agilných metodík sa na softvérový projekt pozerá ako na tok rozširujúcich možností a nových znalostí. Nové možnosti rozširujú vyvíjaný produkt, najmä jeho funkcionálnosť. Novozískané znalosti umožňujú produkt vytvárať lepšie, teda vylepšujú samotný proces vývoja. Neustály tok nových znalostí v rámci každej iterácie dáva vývojárskemu tímu možnosť lepšie posudzovať a odhadovať svoju prácu a tým dovoľuje lepšie plánovať a usmerňovať nie len samých seba ale aj celý tím k splneniu zvolených cieľov. Samo-organizácia a neustále zlepšovanie patria medzi základné princípy agilných metodík [1].

Plánovanie iterácie predstavuje v tomto ohľade proces počas ktorého sa definuje rozpätie používateľských príbehov, ktoré budú implementované počas nasledujúcej iterácie. Používateľské príbehy, teda hrubo definované požiadavky na vyvíjaný produkt sú počas plánovania iterácie dekomponované na jednotlivé vývojárske úlohy, ktoré sú základe príslušných metód plánovania buď zahrnuté alebo nezahrnuté do plánu iterácie. V podstate je možné rozlíšiť dve rôzne metódy agilného plánovania iterácie a to plánovanie na základe rýchlosti vývoja a plánovanie na základe prijatých záväzkov [1,4].

Plánovanie na základe rýchlosti vývoja

Pojem rýchlosť vývoja v kontexte agilného vývoja je definovaný ako pomer vyriešených používateľských príbehov, ohodnotených buď ideálnymi dňami alebo príbehovými bodmi, za jednotku času. Rýchlosť vývoja predstavuje takto mieru napredovania vývojárskeho tímu [1,4].

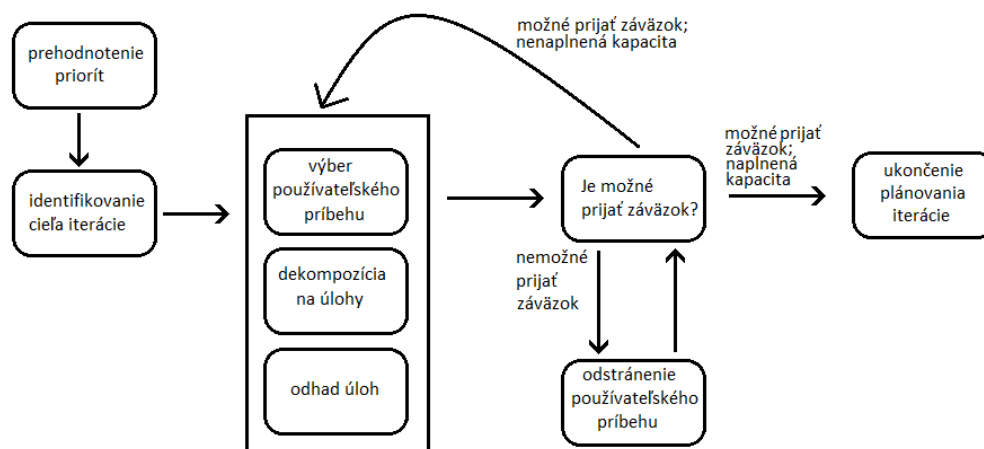
Plánovanie novej iterácie pomocou rýchlosti vývoja začína prehodnotením priorit z dôvodu získania nových znalostí ohľadom produktu. Následne sa odhadne alebo prehodnotí rýchlosť vývoja a určí sa cieľ iterácie. Na základe cieľa sa vyberú tie používateľské príbehy, ktoré spĺňajú daný cieľ. Do rozsahu plánovanej iterácie sa pridáva toľko používateľských príbehov, koľko je možné dokončiť na základe určenej rýchlosti vývoja.



Obr. 1. Plánovanie na základe rýchlosti vývoja [1].

Plánovanie na základe prijatých záväzkov

Plánovanie na základe prijatých záväzkov, na rozdiel od plánovania na základe rýchlosti začína prehodnotením priorit a identifikovaním cieľa. Plánovanie následne prebieha tak, že vývojársky tím po tom, čo jednotlivé používateľské príbehy dekomponuje na úlohy, samostatne rozhodne, či dokáže v rámci iterácie implementovať daný používateľský príbeh. Formálne sa tak vývojársky tím zaviazá k dokončeniu daného používateľského príbehu [1]. Rozsah sa naplní, keď sa vývojársky tím už nedokáže zaviazat k splneniu žiadneho používateľského príbehu.



Obr. 2. Plánovanie na základe prijatých záväzkov [1].

Ktorú metódu použiť?

Plánovanie na základe rýchlosti vývoja, predstavuje z môjho pohľadu veľmi jednoduchú metódu plánovania iterácie. Je to logické, keďže odhadnúť rýchlosť vývoja v rámci iterácie je pre každého vedúceho tímu a jeho tím pomerne jednoduchá záležitosť. Aj keď vývojársky tím je pre vedúceho nový a jednotliví členovia sa navzájom dobre nepoznajú, dokáže vedúci spolu s tímom na základe predchádzajúcich skúseností rýchlosť vývoja celkom presne odhadnúť [3]. A aj keď je prvý odhad nie úplne presný, ďalšími iteráciami sa tím dopracuje k pomerne presnému odhadu rýchlosti svojho vývoja [1].

Oplatí sa potom vôbec plánovať iteráciu inak, keď existuje pomerne jednoduchá a presná metóda plánovania iterácie? Keď sa pozriem na alternatívu, ktorú predstavuje metóda plánovania na základe prijatých záväzkov tak to určite stojí viac než na zváženie. Plánovanie na základe prijatých záväzkov je o niečo náročnejšie, lebo vyžaduje väčšiu mieru angažovanosti jednotlivých členov tímu do plánovacieho procesu. Ale nie je toto práve to, čo vedúci a od svojho tímu práve chce? Jednotliví členovia tímu takto berú za seba a tím záväzok implementácie daného používateľského príbehu, čím sa samozrejme zväčšuje miera angažovanosti jednotlivých členov tímu na úspešnom dokončení iterácie. Každý člen tímu prijatím záväzku takto pociťuje vyššiu mieru zodpovednosti a samozrejme aj značnú motiváciu. Štúdia dokazuje, že ak vývojárske tímy dostanú väčšiu

mieru voľnosti a možnosť samo-organizácie, zvyšuje sa súhra celého tímu, keďže jednotliví členovia si sami pridelujú úlohy, prijímajú záväzky, manažujú samých seba a samostatne sa hodnotia a zlepšujú svoje schopnosti [2]. Je preto z môjho pohľadu logické použiť na dosiahnutie tohto synergického efektu aj metódu plánovania na základe prijatých záväzkov ako časť samo-organizácie tímu. V spomenutej štúdií je dokonca uvedené, že vývojárske tímy si priamo pochvalovali možnosť slobodnej voľby výberu cieľa iterácie a určenia zodpovedností, ktoré vedú k splneniu daného cieľa spoločným úsilím [2].

V neprospech plánovania na základe rýchlosti hovoria aj určité skúsenosti s odhadom rýchlosti a odhadom veľkosti jednotlivých používateľských príbehov. Veľkosť používateľských príbehov sa odhaduje na pomerne vysokej úrovni. Je preto možné nájsť také používateľské príbehy, ktoré aj keď majú rovnaký odhad veľkosti, ich skutočná veľkosť je o niečo vyššia, avšak nie dosť vysoká aby dostali vyššie ohodnotenie. Výberom takýchto používateľských príbehov je možné presiahnuť odhadnutú rýchlosť vývoja [1]. Pri plánovaní pomocou prijatých záväzkov sa tomuto riziku vývojársky tím vystaví oveľa menej.

Z pohľadu členov vývojárskeho tímu má tento prístup tiež svoje výhody. Je takto podľa mňa zamedzené určité vnútenie plánu iterácie. V každom projekte existujú termíny, ktoré treba dodržať a pri použití plánovania na základe rýchlosti vývoja je pre vedenie jednoduchšie vnútiť taký plán, ktorý aj keď formálne vyhovuje metóde plánovania na základe rýchlosti vývoja, avšak pre vývojársky tím je z určitých dôvodov nespĺniteľný.

Na druhú stranu prijatie záväzku je pomerne odvážny krok. Znamená, že jednotliví členovia tímu dávajú sľub svojmu tímu a zainteresovaným osobám, že plán iterácie je uskutočniteľný a že preberajú časť zodpovednosti na splnení iterácie [3]. Avšak nie každý člen tímu musí mať tento pocit. Preto sa odporúča aby každý člen tímu verbálne potvrdil pred ukončením plánovania iterácie prijatie tohto záväzku. Ak toto člen tímu nedokáže, je potrebné diskutovať dôvody a upraviť plán podľa potreby. Prijatie záväzku takto dáva členom tímu možnosť vyjadriť svoje obavy skôr ako je neskoro. Prijatie záväzku má však aj svetlé stránky, keďže pomáha zvyšovať motiváciu celého tímu, že tím dokáže vyriešiť všetky problémy, ktoré sa počas iterácie vyskytnú. Vzniká tak väčšia dôvera v seba samého ako člena tímu a dôvera v schopnosti celého tímu [3]. A práve týmto vzniká podľa mňa v rámci tímu synergia, ktorá umožňuje projekty riešiť rýchlejšie, profesionálnejšie a kvalitnejšie.

Záver

Plánovanie iterácie je dôležitý proces, ktorý udáva smer napredovania vývojárskeho tímu na pomerne dlhú dobu jednej iterácie. Je preto samozrejme nutné vykonať plánovanie iterácie čo najdôslednejšie. Z dvoch známych metód plánovania by som preto každému vývojárskemu tímu odporučil použiť plánovania na základe prijatých záväzkov, a to z dôvodu, že použitím tejto metódy vývojársky tím oveľa viac propaguje myšlienky agilného vývoja ako samo-organizácia a neustále zlepšovanie samých seba. Prijatím záväzku splnenia navrhnutého plánu vzniká v členoch tímu dôvera v samých seba a vo svoj tím, čo vedie k lepšej spolupráci a dosiahnutiu synergického efektu naprieč celým vývojárskym tímom.

Použitá literatúra

1. Cohn, M.: Agile Estimating and Planning. 1. vyd. New Jersey: Prentice Hall, 2005. ISBN 0-13-147941-5.
2. Honda, J., Noble, J., Marshall, S.: Balancing Acts: Walking the Agile Tightrope. In: *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '10)*. ACM, New York, 2005, s. 5-12.
3. Shore, J.: The Art of Agile Development. 1. vyd. Sebastopol: O'Reilly Media, 2007. ISBN 978-0596527679.
4. Zhong, S., Liping, Ch., Tian-En, Ch.: Agile Planning and Development Methods. In: *Computer Research and Development (ICCRD)*, 2011 3rd International Conference, 2011, s. 488-491.

Annotation

Creating Team Synergy with Commitments

In this essay I address the problem of creating an iteration plan within the agile methodologies. I briefly explain the two base iteration planning methods, which are velocity-driven iteration planning and commitment-driven iteration planning. I discuss the suitability of both methods with confrontation with relevant experience and knowledge of other authors. From the confrontation I suggest to use the commitment-driven iteration planning method as this method is more suitable in the context of working with agile methodologies. Commitment is a bald statement and with it the development team gives a promise to stakeholders that they will finish the iteration with completing the planned goal. With this the development team creates a great feeling of motivation, self-trust and trust with the whole team, which creates, by my opinion, a synergy effect throughout the whole team.