

KOMBINÁCIE AGILNÝCH METÓD VÝVOJA PRI PLÁNOVANÍ PROJEKTOV

Kanban nie je bojové umenie.

Martin Uhrin

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

Abstrakt. *Plánovanie softvérového projektu je kľúčovým aspektom v oblasti riadenia jeho vývoja. Na začiatku tvorby softvéru každého zaujíma, ako dlho bude jeho vývoj trvať. Toto sa nedá s určitosťou povedať nikdy a správny odhad je viac menej otázkou skúseností, poprípade náhody. Preto je dôležité vedieť efektívne využívať čas. Agilné metódy vývoja nám dávajú dobré rady, ako toto dosiahnuť, a preto sa čím ďalej tím viac začínajú využívať. Ľudia stále hľadajú nové a lepšie spôsoby tvorby softvéru, preto vznikajú nové metódy agilného vývoja. My sa pozrieme konkrétne na Scrum a Kanban, považujeme nad ich výhodami a nevýhodami a pokúsime sa nájsť spôsob, ktorým by sme pri vývoji softvéru využili to najlepšie z oboch.*

Kľúčové slová: *agilné metódy vývoja, Scrum, Kanban,*

Úvod

Softvér sa svojimi vlastnosťami ako sú zložitosť, neviditeľnosť, nestálosť odlišuje od projektov v iných oblastiach a tým veľmi sťažuje riadenie jeho vývoja. Softvéroví inžinieri sa musia s týmito problémami vysporiadať. Medzi problémové oblasti riadenia spadá jednoznačne aj plánovanie projektu. V súčasnej uponáhľanej dobe chcú všetci všetko hneď a v najlepšej možnej kvalite. Toto samozrejme ešte stále nie je možné dosiahnuť, no vznikajú rôzne postupy, pomocou ktorých sa tomuto darí približovať.

Agilný vývoj softvéru sa v poslednom čase čoraz viac dostáva do povedomia vývojárov, ktorí si začínajú uvedomovať jeho veľkú výhodu. Cieľom agilnej metódy

vývoja softvéru je rýchlejšie reagovať na zmeny, požiadavky zákazníka a udržiavať s ním čo možno najbližší kontakt. Týmto sa mieni predísť nedorozumeniam a dodať softvér v čo možno najlepšej kvalite v čo možno najkratšom čase.

Plánovanie však nie je len o odhade času potrebného na vývoj. Podľa môjho názoru je oveľa dôležitejšie zaoberať sa metódami vývoja, ktoré uľahčujú spoluprácu a tým skracujú čas na vývoj ako neskôr keď sa blíži dátum ukončenia hnať vývojárov a hroziť im ak to nestihnú v čas.

Prečo plánovať?

Prečo je plánovanie vlastne také dôležité? A čo to vôbec znamená? Pod pojmom plánovanie si môžeme predstaviť určitú formu stratégie. Aké postupy využijeme, koľko nám to bude trvať, nebolo by lepšie najprv spraviť niečo a potom niečo iné...

Je dôležité uvedomiť si, že všetky veci vznikajú po častiach. Každý produkt sa skladá z určitých medziproduktov, ktorých zložením dostávame výsledok. Plánovanie opisuje, ktoré časti sa kedy majú vytvoriť a pomáha odhadovať ako dlho to asi bude trvať.

Uvedomenie si, že vytvorenie výsledného produktu možno len cez menšie bližšie špecifikované časti je v tomto prípade kľúčové. Koniec koncov je to prirodzené. Každý človek, keď mu je predstavený nejaký problém, automaticky tento problém dekomponuje na menšie časti a systematicky sa ich snaží vyriešiť. Slovo systematicky je podľa mňa veľmi dôležité, pretože označuje, v akom poradí sa majú ktoré veci vykonať. Zoberme si praktický príklad, napríklad uvarenie obeda. Keď človek dostane takúto úlohu, tak vždy najprv skontroluje, či má všetky potrebné nástroje na jeho vykonanie – hrniec, sporák atď. Potom sa skontrolujú suroviny, ak chýbajú, tak ich treba najskôr nakúpiť a až potom je možné ich uvariť. V tomto prípade nám všetko uľahčuje recept, ktorý svojím spôsobom predstavuje určitú formu plánu. Hovorí, v akom poradí a kedy je potrebné dané suroviny kombinovať.

Plánovanie je teda veľmi dôležité, pomáha nám lepšie si predstaviť budúcnosť, odhaľovať skryté chyby, ktoré by mohli nastať atď. Poďme sa teraz pozrieť na to, aké metódy plánovania existujú a aké nástroje na to máme k dispozícii.

Plánovanie v agilnom tíme

Ako už vieme, agilné riadenie projektu predstavuje sériu bežne používaných prístupov k riadeniu s cieľom lepšie zvládnuť neviditeľnosť a nepredvídateľnosť, čo sa však nie vždy darí. Napríklad, pre veľké projekty je presné plánovanie a odhadovanie pre celú životnosť projektu veľmi náročné. Kvôli neustálej zmene požiadaviek na projekt a ich neúplnosť na začiatku projektu, plánovanie musí byť pod kontrolou a musí sa neustále prepracovávať podľa potrieb. V agilnom vývoji je plánovanie založené najmä na odhadovaní úsilia, ktoré je treba vložiť do splnenia pridelenej úlohy [1].

Takto odhadnuté úsilie, ktoré sa zakladá zväčša na skúsenostiach tímu, však nemusí byť a ani nie je vždy presné, napriek tomu, že existuje stratégia, pomocou ktorej dosiahneme lepšie a presnejšie odhady. Nazýva sa plánovací poker (angl. *planning poker*).

Táto metóda spočíva vo využívaní kariet na odhadnutie potrebného úsilia, ktoré treba vynaložiť na splnenie úlohy. Karty sú označené číslami stúpajúcej postupnosti

(väčšinou Fibonacciho postupnosť). Každý člen tímu dostane takéto karty. Podstatou tejto metódy je, že všetci ukážu svoj odhad naraz. Týmto je zabezpečená určitá eliminácia ovplyvňovania jednotlivých členov. Napríklad ak si niekto myslí, že úsilie, ktoré treba vynaložiť na splnenie nejakej konkrétnej úlohy, má hodnotu 21, iný si môže myslieť napríklad 5. Keby to ten, čo si myslí 21 povedal nahlas, mohol by ovplyvniť toho, ktorý si myslí 5 a ten by povedal napríklad 13. Tomuto sa presne chce zabrániť, preto všetci ukážu svoj odhad naraz.

Veľká výhoda je podľa mňa nielen v zabránení ovplyvňovania, ale aj v podnietení konverzácie celého tímu. Každý sa môže vyjadriť, prečo si myslí, že daná úloha zaberie toľko a toľko času a tím sa dospeje k ešte lepším výsledkom. Takisto môže zmeniť názor majiteľa produktu (angl. *product owner*), ktorý po konzultácii môže pozmeniť poradie dôležitosti jednotlivých požiadaviek.

Vysoké stávky

Hoci metóda plánovacieho pokru sa javí ako ideál a ako taká aj funguje, vždy je niečo vylepšovať. Podľa mňa dobrou alternatívou je napríklad použitie pokrových žetónov pri rozhodovaní, ktorý používateľský scenár (angl. *user story*) sa zahrnie do prvého vydania [5].

Zákazníci majú v tomto rozhodovaní tú najvyššiu autoritu. Oni rozhodujú, ktorú funkciu zaradia do vydania a ktorú nie. Ako im teda my aspoň priblížime, ktorá funkcionálna si vyžaduje veľké množstvo práce a ktorá menej? Presne na tento účel slúžia pokrové žetóny. Aby zákazníci a manažéri mali akú takú predstavu o tom, koľko úsilia si implementovanie danej funkcionality vyžaduje.

Žetóny v tomto prípade predstavujú taký fyzickejší prístup k plánovaniu. Plánovanie sa stáva niečím, čo zapája zmysly. Nielenže je to väčšia zábava, ale umocňuje pocit, že manažér alebo teda zákazník musí niečo staviť, niečo obetovať na danú úlohu. Toto potom napomáha k presnejšiemu určeniu priorít jednotlivých funkcionality.

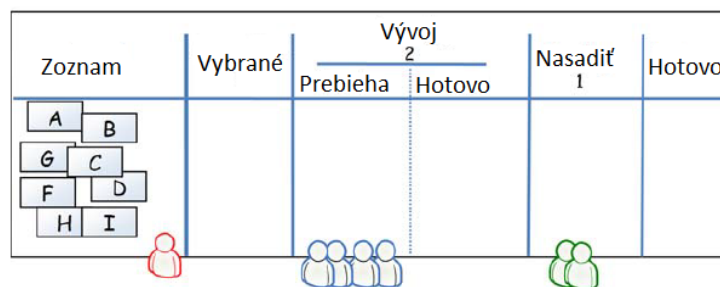
Čo je to Kanban?

Počuli ste už niekedy slovo Kanban? Prosím? Že či je to nejaké japonské bojové umenie? Vôbec nie, hoci slovo „kanban“ má pôvod v Japonsku. Ak teda netušíte čo to je, pokúsím sa vám to v pár vetách priblížiť.

Kanban slúži na vizualizáciu práce a jej limitáciu [2]. Jeho základom je tabuľa, na ktorej sú primárne tri stĺpce:

1. Čo sa má urobiť
2. Čo je vo vývoji
3. Čo je hotové

Samozrejme, stĺpcov môže byť a väčšinou aj býva viacej, pre jednoduchosť však uvedme len tieto tri. Príklad zložitejšej tabule môžeme vidieť na obrázku nižšie:



Obr.1. Kanban tabuľa [2].

Jediná vec, ktorú Kanban predpisuje, je vizualizácia práce, ktorá prebieha a jej limitácia (angl. *work in progress* - *WIP*). Cieľom je vytvoriť plynulý prechod medzi jednotlivými fázami a minimalizovať režijný čas.

Na obrázku č.1 môžeme vidieť znázornených aj ľudí, ktorí budú vykonávať tú-ktorú činnosť. Červeným je znázornený majiteľ produktu (angl. *product owner*), modrým sú vývojári a zeleným tí, ktorí softvér nasadzujú a testujú.

Teraz keď už máme akú takú predstavu o tom, čo Kanban je, poďme sa pozrieť ako a či sa vôbec Scrum od Kanbanu líši.

Scrum vs Kanban

Na Scrum môžeme nazerať aj ako na dosť prísny prístup k manažovaniu projektu – šprinty, zoznamy, plánovanie šprintov atď. Scrum je však niečo viac ako len toto – jeho úlohou je vzbudiť zmenu a zlepšiť prístup k vývoju projektu.

Cieľ Scrumu – priniesť nejakú zmenu a tým pádom zlepšiť proces vývoja, sa dosahuje pomocou záväzkov [4]. Tieto záväzky môžeme rozdeliť do dvoch úrovní. Na prvej úrovni sú to denné záväzky, ktoré sa prezentujú formou troch otázok: „čo si urobil dnes?“ „čo mieniš urobiť zajtra?“ a „čo ti stojí v ceste?“. Na druhej úrovni je to záväzok voči tímu a zákazníkovi – dodať funkčný projekt na konci jednotlivých šprintov. Neúspech vyvolá diskusiu o tom, čo bolo zle a podnieti zmenu k lepšiemu.

Keď sa nejaká organizácia rozhodne začať používať Scrum, musí prejsť veľkými zmenami. Pre manažérov to môže znamenať naberanie rôznych zodpovedností, rolí, pre vývojárov zase nový a úplne iný systém vývoja. Môžeme teda povedať, že Scrum je skôr revolúcia ako evolúcia.

Kanban na druhej strane využíva limitáciu množstva prebiehajúcej práce. Cieľom je takisto podnietiť konverzáciu a problémy riešiť hneď, ako sa objavia. Nedodržiavanie tejto limitácie a nediskutovanie o problémoch môže viesť k stagnácii a neschopnosti zlepšovať sa.

Takže Kanban používa limitáciu množstva prebiehajúcej práce na podnietenie zmeny, Scrum používa záväzky. To je ten zásadný rozdiel v prístupe.

Radšej Evolúcia

Rozdiel medzi Kanbanom a Scrumom spočíva takisto v tom, že pri nasadení Kanbanu sa proces vývoja ako taký nemení, aspoň nie hneď na začiatku. Začína sa s procesom, ktorý

sa používa a len mierne sa „kanbanizuje“ tým, že sa nejakým spôsobom zobrazuje priebeh práce, limituje sa jej množstvo. Existujúce roly, zodpovednosti a praktiky zostanú nezmenené.

Môžeme teda povedať, že Kanban je určený na rozvíjanie a zlepšovanie existujúceho procesu, bez ohľadu na to, aký je tento proces na začiatku. Tu je opäť možné vidieť rozdiel medzi Kanbanom a Scrumom. Existuje viacero dotazníkov¹, pomocou ktorých sa dá zistiť, či ten-ktorý tím naozaj dodržiava všetko presne podľa toho, ako to Scrum predpisuje. Naopak pre Kanban neexistuje žiadny potvrdzovací test. Tým, že Kanban použijeme na to, aby sme podnietili nejakú zmenu, vylepšili súčasný proces, vznikne nový, pre konkrétny tím unikátny proces, ktorý je efektívnejší ako ten starý.

Rámec vs celý systém

Ďalšie odlišnosti v používaní Kanbanu môžeme vidieť vo vlastnostiach, ktoré vznikajú ako vedľajší efekt nepoužívania Scrum prístupu. Scrum ohraničuje vývoj určitej funkcionality v rámci šprintu, pričom v ideálnom prípade nedochádza ku žiadnemu rušeniu z vonka. Želaným efektom je v tomto prípade urobiť prácu potrebnú na vykonanie úlohy čo najpredpovedateľnejšou, aby sa dodržali šprintové požiadavky.

Kanban nevyžaduje takýto rámec, naopak podporuje pohľad na celkový systém. Pomocou skupiny techník zjednodušuje koordináciu jednotlivých úloh v rámci celého systému. Podľa mňa používanie limitácie prebiehajúcej práce a jej vizualizácie pomocou Kanban tabule môže výrazne pomôcť pri komunikácii so zákazníkmi. Myslím si, že takáto forma vizualizácie pomôže hlavne v tom, že nie je potrebné, aby sa so zákazníkmi komunikovalo prostredníctvom produktového vlastníka (angl. *product owner*), ale je možné komunikovať s viacerými ľuďmi z danej organizácie naraz. Vizualizácia práce pomáha zákazníkovi lepšie a rýchlejšie pochopiť, čo sa momentálne vyvíja, čo sa bude najbližšie vyvíjať, čo už je hotové a vývojárom pomáha v lepšom pochopení vlastností žiadaného systému.

Ďalší rozdiel vidím aj v spôsobe zhodnocovania vykonanej práce, hlavne sa jedná o takzvané denné stretnutia v stoji (angl. *daily stand-up meetings*). Obsahom týchto stretnutí v tímoch, ktoré používajú Scrum je zhodnotenie vykonanej práce, zostávajúcej práce a diskusia o problémoch, ktoré môžu nastať, resp. ktoré nastali a treba ich nejakým spôsobom riešiť. Kanban je podľa mňa omnoho efektívnejší, čo sa týka takýchto stretnutí, pretože tým, že vykonávaná práca je prehľadne zobrazená na Kanban tabuli, nie je potrebné ju jednotlivo zhodnocovať. Preto sa aj toto stretnutie môže viacej sústrediť na prácu ako na ľudí, čo v konečnom dôsledku môže podľa mňa zabezpečiť efektívne stretnutia aj s vyšším počtom ľudí.

Záver

Ako teda skombinujeme tieto metódy, aby sme dosiahli čo možno najlepší výsledok? Dá sa to vôbec? Na tú druhú otázku je odpoveď veľmi jednoduchá. Dá sa to. Kombinácia týchto dvoch agilných metód je možná a dokonca aj úspešne využívaná, nazýva sa Scrumban [3].

¹ <http://antoine.vernois.net/scrumbut/>

Na prvú otázku je však odpoveď už omnoho ťažšia. Ktoré princípy adoptujeme z jednej metódy a ktoré z druhej tak, aby si navzájom neodporovali, práve naopak aby sa navzájom dopĺňali a umožňovali tak efektívnejšie, rýchlejšie a spoľahlivejšie vyvíjať softvér?

Na zodpovedanie tejto otázky je potrebné sa najprv zamyslieť nad princípmi jednotlivých metód, či si v niečom odporujú, resp. či niečo riešia inak. Scrum predpisuje šprinty, ich plánovanie a vykonávanie, Kanban o príprave toho, čo sa ide robiť mlčí. Preto je podľa mňa vhodné prebranie resp. zachovanie konceptu šprintov zo strany Scrumu.

Po ich naplánovaní by sa stanovil maximálny počet úloh, ktoré môžu byť vykonávané. Toto číslo závisí od veľkosti tímu, jeho vyspelosti atď., postupne sa však môže meniť, kým sa nenájde vhodný kompromis. Týmto sa zabezpečí to, že ak nastane problém s nejakou úlohou a počet rozpracovaných úloh bude rovný maximálnemu stanovenému počtu rozpracovaných úloh, nebude možné začať pracovať na ďalšej úlohe, ale bude nutné najprv vyriešiť problém s danou úlohou.

Pri stanovovaní požiadaviek na systém a následnom plánovaní šprintov by podľa mňa veľmi pomohlo použitie pokrových žetónov. Tie by pomohli zákazníkovi lepšie pochopiť cenu realizovania jednotlivých požiadaviek, lepšie si uvedomiť, koľko sú ochotní „zaplatiť“ za tú ktorú funkciu.

Scrum takisto prikazuje denné stretnutia v stoji (angl. *daily stand-up meetings*), Kanban na druhej strane uprednostňuje vizualizáciu vykonávanej práce, čím sa potreba hovoriť o práve vykonávanej úlohe stráca. Preto si myslím, že tieto stretnutia by sa mali zachovať, avšak predmet ich stretnutia by sa mal týkať iba diskusie o takých úlohách, s ktorými majú členovia vývojového tímu problémy.

Myslím si, že takýto spôsob kombinácie týchto dvoch agilných metód je jednoducho uskutočniteľný a pomôže rýchlejšie vyvíjať softvér, ktorý bude spoľahlivejší. Samozrejme, je to len začiatok a určite sa nájde ešte viacero rôznych foriem kombinácie nielen týchto dvoch metód ale aj ďalších, pomocou ktorých by sme dosiahli ešte lepšie výsledky. Tento spôsob je však podľa mňa veľmi jednoducho aplikovateľný pre tímy, ktoré už používajú Scrum a chceli by túto metódu ešte o niečo vylepšiť.

Použitá literatúra

1. Desharnais, J.M., Buglione, L., Kocaturk, B.: Using the COSMIC method to estimate agile user stories. In: *Proc. Of the the 12th International Conference on Product Focused Software Development and Process Improvement*. New York (2011), 68-73.
2. Kniberg, H., Skarin, M.: Kanban and Scrum: How to make most of both. *InfoQ*, (2009).
3. Nikitina, N., Kajko-Mattsson, M., Strale, M.: From scrum to scrumban: A case study of a process transition. In: *Conference on Software and System Process (ICSSP), 2012 International*, (2012), 140-149.
4. Sjoberg, D.I.K., Johnsen, A., Solberg, J.: Quantifying the Effect of Using Kanban versus Scrum: A Case Study. *Software, IEEE*, Vol. 29, No. 5, (2012), 47-53.
5. Yip, J.: Hands-on release planning with poker chips. In: *Proc. Of the 14th Conference on Pattern Languages of Programs*. New York (2007).

Annotation

Combinations of agile methods of development in project planning

Planning software project is a key aspect in the field of managing development of a project. In the beginning everybody wants to know a release date. This can not be said with certainty and good estimate is just a matter of experience, coincidence. That's why it is important to know how to effectively use the time. Agile methods give us good pieces of advice how to accomplish this and that's why people start using them. People keep searching for better ways to develop a software, therefore new methods arise. We are going to have a look at Scrum and Kanban, think about their advantages and disadvantages and try to find a way to make the most of both when it comes to developing a software.