

Metodika na manažment úloh

Verzia 11.11.2015

Tabuľka 1. Autori

Autor	Rola
Tomáš Donko	

Tabuľka 2. História zmien

Verzia	Dátum	Autor	Popis
1.0	11.11.2015	Tomáš Donko	Vytvorenie dokumentu

Obsah

1	Úvod	1
1.1.	Dedikácia.....	1
1.2.	Súvisiace dokumenty.....	1
1.3.	Slovník pojmov	1
2	Úlohy	2
2.1.	Vyhodnotenie úloh na začiatku šprintu.....	2
2.2.	Vytváranie a pridelovanie úloh.....	2
	User story.....	2
	Task	2
	Code-review.....	2
2.3.	Upravovanie úlohy	3
2.4.	Ukončenie úlohy	3
2.5.	Sledovanie úloh.....	3

1 Úvod

Na manažment úloh používame prostredie Team Foundation Server (TFS), ktoré je zosynchronizované s vývojovým prostredím Microsoft Visual Studio. Commity sa viažu na danú úlohu a takisto pri code-review sa požiada o pull-request a v TFS sa prezerá a poznámkuje kód. V TFS máme vložené názvy šprintov, epic úlohy, user stories a tasky.

1.1. Dedikácia

Táto metodika je určená pre každého člena tímu, ktorý je povinný si viesť záznamy o svojej práci, čo má spraviť a v akom stave sú jeho úlohy.

1.2. Súvisiace dokumenty

Metodika code-review,

Metodika verziovanie zdrojového kódu

1.3. Slovník pojmov

TFS (Team Foundation Server) – je robustný nástroj, ktorý v sebe zahŕňa viacero funkcií. Umožňuje spravovať verzie a slúži ako podporný nástroj pre manažment riadenia projektu, umožňuje vykonávať aj prehliadky zdrojového kódu.

Visual Studio 2015 – vývojové prostredie používané .NET vývojármi.

Commit – je uloženie priebežného stavu práce na lokálnom počítači, spolu s popisom vykonaných zmien

Code-review – revízia zdrojového kódu.

Pull-request – požiadavka na stiahnutie a kontrolu zdrojového kódu.

Epic task – úloha veľkého rozsahu, taká ktorá sa rozkladá aj na niekoľko šprintov.

Šprint – časové ohraničenie na vypracovanie zadaných úloh, používa sa pri agilnom vývoji

User story – úloha, ktorá je vykonateľná spravidla v jednom šprinte, obsahuje sa na tasky.

Task – čiastková úloha, ktorú zahŕňa user story.

2 Úlohy

2.1. Vyhodnotenie úloh na začiatku šprintu

Na začiatku každého šprintu sa rekapituluje čo všetko sa stihlo, prípadne nestihlo spraviť za uplynulý šprint. Úlohy, ktoré boli kompletne, teda spravené, zdokumentované, otestované a prebehlo code-review sa uzatvoria. Ak úloha nebola dokončená preloží sa na ďalší šprint.

2.2. Vytváranie a pridelovanie úloh

User story

Po retrospektíve šprintu sa vymýšľajú nové úlohy (user stories), ktoré dokážeme za tento šprint spraviť. V TFS sa vytvorí nové user story, ktorému vyplníme názov a krátky popis, čo to všetko zahŕňa a obnáša. Následne o tom diskutujeme, dokedy nie je každému členovi tímu úplne jasné, aká je jej zložitosť. Potom musíme tento user story ohodnotiť pomocou kariet v porovnaní s inými user stories. Keď sa všetci zhodneme na výslednom hodnotení, pridelíme mu túto hodnotu a prioritu. Potom sa dohodneme, kto spraví túto úlohu, alebo kto ju má na starosti. Tento proces opakujeme pre každú vymyslenú úlohu (user story).

Task

Po tomto procese je povinný hneď na stretnutí vytvoriť tasky, ktoré obnáša jeho user story. Pri tomto je nutné vyplniť:

- Názov tasku
- Odhadovaný čas
- Zostávajúci čas (na začiatku je rovnaký ako odhadovaný)
- Komu je tento task určený (môže sa líšiť od user story)

Zostávajúci čas je dôležitý pretože z neho sa počíta celkový zostávajúci čas na spravenie úloh a týmto je možné kontrolovať progres celého tímu a ľahšie zistiť, či stíhame spraviť úlohy na šprint alebo nie.

Ďalej je možné aj explicitne zmeniť prioritu daného tasku.

Code-review

Ku každému user story hneď pri vytvorení vytvoríme task „Code-review“ a pridelíme niekoho na túto úlohu. Ten ma na starosti spraviť code-review, keď sú všetky tasky v user story hotové (okrem code-review) a keď bol spravený pull-request.

2.3. Upravovanie úlohy

Keď člen tímu začne reálne pracovať na danej úlohe (nielen programovať, ale aj študovať, alebo analyzovať), je potrebné zmeniť stav tasku z „new“ na „active“. Avšak ak si dá prestávku, prípadne ide spať, alebo robiť niečo iné, tak sa tento stav nemení. Zmení sa až v momente, kedy je táto úloha dokončená podľa požiadaviek na stav „closed“.

Pri upravovaní úlohy sa nastavuje aj aktuálna aktivita na:

- Deployment – keď sa úloha nasadzuje na server
- Design – keď sa úloha modeluje
- Development – keď sa úloha vyvíja
- Documentation – keď sa úloha dokumentuje
- Testing – keď sa úloha testuje

Avšak na tieto aktivity sa nekladie dôraz a je to použité len pre vlastnú potrebu.

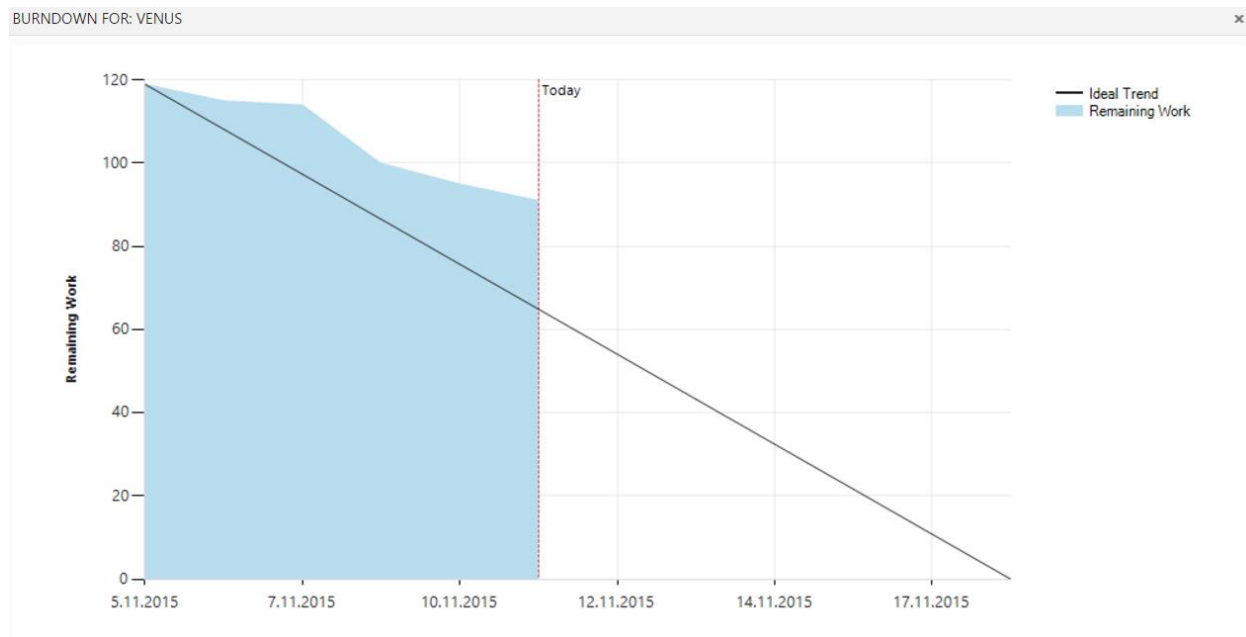
Pri práci na úlohe je možnosť meniť zostávajúci čas, ale pri krátkych úlohách (do 4 hodín) je to skôr zbytočné meniť, pretože pri skončení úlohy sa nakoniec odráta tento čas z celkového zostávajúceho času na tento šprint automaticky. Pri väčších úlohách je tento čas potrebný meniť, kvôli evidencii časového plánu.

2.4. Ukončenie úlohy

User stories, ktoré boli kompletne, teda spravené, zdokumentované, otestované a prebehlo code-review a nakoniec domergované zo svojej vetvy do „dev“ vetvy sa uzatvoria, teda stav sa zmení z „active“ na „closed“. Následne znova pri začiatku šprintu sa vyhodnotí celé user story a stav sa zmení na „resolved“

2.5. Sledovanie úloh

Na prehľadné sledovanie plnenia úloh je v TFS burndown chart pre aktuálny šprint (pozri obrázok č. 1). Na začiatku sa definovali úlohy (5.11.2015) a odtiaľ začala viesť čiara po koniec šprintu, čo je ideálny progres. Pri spaľovaní úloh počas šprintu sa modrá oblasť znižuje. Na konci šprintu by sme mali skončiť na nulovej úrovni ako končí ideálny trend. Ak sme nad ideálnou čiarou, označuje to, že nestíhame časový plán. Vtedy je potrebné zistiť, ktoré úlohy neboli dokončené a kontaktovať člena tímu, ktorý ma na starosti danú úlohu a riešiť tento problém.



Obrázok 1: Burnout chart pre šprint Venus