

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Metodika manažmentu úloh

Tímový projekt
2018/2019

Tím č. 04 : Fastar

Bc. Michaela Balážová
Bc. Kamil Janeček
Bc. Tomáš Jendrejčák
Bc. Matúš Kalafut
Bc. Matej Končál
Bc. Michal Maňak
Bc. Ján Vnenčák

Vedúci projektu: Ing. Kamil Burda

Dátum poslednej úpravy: 13.12.2018

Metodika manažmentu úloh

V našom projekte používame nástroj JIRA pre agilné zaznamenávanie úloh vykonávaných na projekte. Tento nástroj priamo umožňuje vytváranie backlogu, user stories a šprintov.

Úlohy

Úlohy sú usporiadané v 3-úrovňovej hierarchii:

- epic story
- story
- sub-task

Epic story - najväčší celok, ktorý zastrešuje stories tematicky patriace pod danú epic story.

User story - jedna z vlastností, ktorú má výsledný produkt obsahovať. Definuje product owner.

Managment story - slúži pre interné záležitosti v tíme a nesúvisí priamo s produktom.

Sub-task - konkrétna úloha, ktorú je potrebné splniť pre naplnenie user alebo managment story.

Roly

Riešiteľ

- tvorí riešenie úlohy, ktorú má pridelenú.

Watcher

- osoba, ktorá je akýmkoľvek spôsobom zapojena do riešenia úlohy alebo sa úloha týka tej časti vývoja, na ktorej spolupracujú.

Vytváranie úloh

Práca, ktorá je vykonávaná na projekte, musí mať vytvorený príslušný sub-task. Takto dosiahneme prehľad o celkovej práci a práci jednotlivých členov tímu.

Pri vytváraní sub-tasku je potrebné do popisu zadať popis úlohy, zdroje, ktoré by mohli pomôcť pri riešení a postrehy jednotlivých členov tímu. Ďalej sa pre story zadefinujú akceptačné kritéria, aby riešiteľ aj recenzent presne vedeli, čo má daná úloha spĺňať. Sub-tasku sa nastaví priorita, pridajú sa prílohy, napríklad poznámky k sub-tasku zo stretnutia, na ktorom sa daný sub-task rozoberal. Sub-tasku sa priradí šprint, do ktorého patrí a prípadne sa priradí závislosť na iný sub-task.

K sub-task-om je potrebné priradiť watchers, aby boli kompetentné osoby upozorňované na všetku aktivitu týkajúcu sa sub-tasku.

Stavy

Story aj sub-task prechádza tými istými stavmi, avšak pre každý typ to znamená niečo málo odlišné.

Story má vždy jeden zo 6 možných stavov:

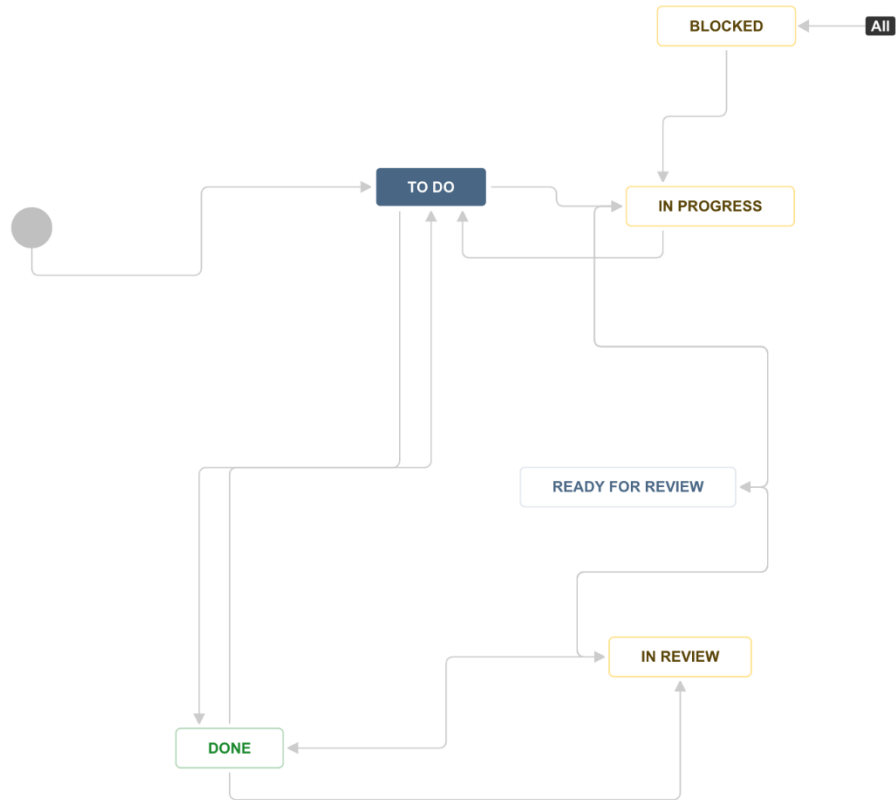
- TO DO - úloha, ktorú je potrebné vyriešiť a nezačala práca na nej. V tejto fáze sa vytvárajú sub-tasks, ktoré predstavujú menšiu granularitu práce na *story*.
- IN PROGRESS - na úlohe sa aktuálne pracuje, sú vytvorené *sub-tasks*, ktoré je potrebné realizovať v rámci danej *story*. *Sub-tasks* môžu byť priradené rôznym osobám.
- READY FOR REVIEW - bola dokončená práca na tejto úlohe, dokončili sa všetky *sub-tasks*, ktoré boli vytvorené k *story*. Úloha čaká na kontrolu.
- IN REVIEW - úloha prebieha kontrolou správnosti, väčšinou formou demo ukážky, ktoré sa prezentuje Product Ownerovi (PO).
- DONE - úloha bola skontrolovaná PO a spĺňa jeho kritériá.
- BLOCKED - do tohto stavu dáme úlohu, na ktorej nevieme pracovať, pretože nie je dokončená iná úloha, od ktorej je závislá.

Sub-task má vždy jeden zo 6 možných stavov:

- TO DO - úloha, ktorú je potrebné vyriešiť a nikto na nej práve nepracuje.
- IN PROGRESS - úlohu si prebrala konkrétna osoba a pracuje na nej.
- READY FOR REVIEW - osoba, zodpovedná za úlohu dokončila svoju prácu. Úloha čaká na kontrolu.
- IN REVIEW - úloha prebieha kontrolou správnosti a či spĺňa definition of done.
- DONE - úloha bola skontrolovaná a spĺňa definition of done.
- BLOCKED - do tohto stavu dáme úlohu, na ktorej nevieme pracovať, pretože nie je dokončená iná úloha, od ktorej je závislá.

Na obrázku č. 1 je znázornený proces, ako sa môžu jednotlivé stavy medzi sebou meniť.

Do stavu BLOCKED môžu prejsť všetky úlohy zo všetkých stavov.



Obr. č. 2 : Grafické znázornenie stavov a prechodov medzi nimi

Akceptovanie user story Product Ownerom

1. Tím vytvorí demo s požadovanou funkcionalitou, ktorá bola špecifikovaná v *story*.
2. Product owner (PO) presunie *story* zo stavu READY FOR REVIEW na stav IN REVIEW pred odskúšaním demo ukážky.
3. PO odskúša demo.
4. Rozhodnutie PO:
 1. Ak je demo v poriadku a PO nemá žiadne výhrady, presunie ho do stavu DONE a táto *story* sa považuje za vyriešenú.
 2. Ak časť funkcionality nie je v poriadku:
 1. PO presunie aktuálnu *story* do stavu DONE
 2. Tím odhadne chýbajúcu funkcionalitu *story points* a odpočíta nové *story points* od pôvodného odhadu.
 3. Chýbajúcu časť pridá s novým odhadom do ďalšieho šprintu.
 3. Ak demo nebolo vôbec vyhovujúce, PO vráti tímu *story* na prerobenie (z IN REVIEW sa potrebujeme chceme dostať do stavu TO DO):
 1. Zo stavu IN REVIEW sa pomocou "Stop QA" dostaneme naspäť do stavu READY FOR REVIEW.
 2. Zo stavu READY FOR REVIEW sa pomocou "Not Completed" dostaneme do stavu IN PROGRESS.

3. Zo stavu IN PROGRESS sa pomocou "Stop Work" dostaneme do stavu TO DO.