Pitanja

13. svibnja 2013.

8:25

* CRM?
* Wiki?

Postoji li redovni tjedni sastanak voditelja?

* Prijedlog: pon. ujutro, matija, ana, ja, kruno, voditelji timova
* Agenda:
  + Mgmt info (Matija)
    - Novi kupci
    - Ideje
    - Ostalo
  + Product planning (Goran)
    - Status razrade novih featura (iz ideja)
  + Stanje produkcije (Davor)
    - Bugovi
    - Performance
    - Lessons learned, prijedlozi za poboljšanja
  + Tekući projekti (leadovi)
    - Stanje
  + Customer info (Kruno?)
  + Ostalo

Tjreiog neruipgrcnuerp gieqprhješjg

Erg ergwer

Kako se raspoređuju taskovi?

* + Improvements (new feature) koje netko upiše

Može li se release note generirati dirketno iz Jire?

Pitanja o Jiri (vidi page [Project monitoring](onenote:General.one#Jira&section-id={B0D9E4AB-2E26-4C69-BCCA-3B936D9DA3F2}&page-id={DA2A7322-7F1B-4AA3-B24E-445A182207BA}&base-path=https://d.docs.live.net/a76bfdf80a4c381e/Documents/Farmeron))

Ima problema s mergingom. Što je uzrok?

* Problem u proceduri?
* Problem u nepridržavanju procedure?

Project monitoring

13. svibnja 2013.

12:27

Jira:

* Treba uvesti reda u Issue:
  + Svatko proći kroz sve taskove koji su mu assignani i postaviti pravi status (Resolve issue koji su resolvani)
  + Svatko proći kroz Issue za koje je Reporter i ili zatvoriti ili reopen
  + Za bugove: uvijek pisati Affects Version
  + Uvijek pisati i Fix verziju, čim se zna (dovoljno je na do drugu decimalu)
    - Za bugove - to ide s nekim hotfixom (3. decimala) trenutne verzie, pa se može pisati druga decimala trenutno deployane verzije
    - Za taskove u novom projektu - pisati planiranu verziju projekta
    - Za feature proposale - ništa dok se ne definira u kojem porjektu ide
  + Za taskove u novom projektu - pisati due date prema proj.planu
* Time tracking:
  + koja je svrha pisanja vremena? - da možemo procijeniti koliko još posla imamo do kraja projekta
  + Zato treba za svaki projekt definirati taskove, njihova trajanja
  + Onda asignee treba redovno updateati status (možemo li dodati % complete?) i vrijeme potrošeno
  + Kada se resolve sub-task, staviti remaining time na 0, da bi račun za estimated finish taska bio točan
* Projekti
  + U Farmeron bi trebali ići maintenance i production/preview bugs taskovi
    - Međutim, tu već imaju upisane verzije - jeli se to mislilo za nove feature?
  + U Strategy - new feture requests? (ili za to novi projekt)
  + Svaki projekt ima svoj projek u Jiri
  + Test?
    - Planirati trajanje
    - TCs? - ne mogu se voditi u Jiri - previše bi bilo za svaki tc pisati task
    - Dodjeljivanje ljudi

Razgovor s Davorom:

* Project lifecycle:
  + Iz inputa od Ane, Dave, Matije napraviti Functional spec
    - Razrađeni featuri, WBS
    - Review: Matija, Ana, development
  + Na osnovu FS se radi TestCase Book, koji isto tako ide na review
    - Ako imamo popis TC-ova, kako uspješnih, tako i neuspješnih, puno ćemo efikasinje i brže testirati (neće svi ponavljati iste stvari)
  + Na osnovu FS, dev radi EE
  + Na osnovu EE i WBS se radi timeplan - raspored u sprintove i računa se kraj projekta
    - Ako dođu izmjene, mora se nešto izbaciti ili pomjeriti kraj projekta
      * Ako se netko buni, imao je review
    - Tijekom sprinta, nema izmjena u tekućim featurima
    - Unit test radi svaki developer
      * Ne treba sada raditi unit testove za sve unazad, nego kako nam nešto dolazi na impl, razviti unit test za taj dio. Na taj način ćemo nakon nekog vremena imati pokriveno većinu aplikacije
    - Odrediti nakon kojih sprintova se radi i Test
      * Pogleda se koliko TC-ova ima za skup funkcionalnosti koji pokriva srpint i vidi se jeli potrebno
      * Ako ne, ti TC-ovi se prebace na kasnije
      * Ako da, definirani tester (**ne** netko od developera) testira prema TC-ovima. Test bi trebao simulirati stvari s kojima se pravi korisnici susreću
    - Završni sprint je svakako test, migracija podataka, deploy, itd

* Za svaki novi projekt imati novi projekt u Jiri
  + WBS se pretoči u taskove i subtaskove i upše vrijeme trajanja (pojedinačni (sub)taskovi nebi trebali veći od 16h)
  + Svaki developer vodi potrošeno vrijeme
    - Bitno je pisati svo vrijeme: ako si cijeli dan radio na nečemu, nećeš pisati 5h samo zato što si imao i ručak i sastanak, itd.
    - Cilj nam nije (za sada) mjeriti produktivnost developera, nego dobiti točnu procjenu završetka radova. Zato moramo voditi vrijeme na ovaj način
    - Kasnije možemo razmisliti o uvođenu EBS ([Evidence Based Scheduling - Joel on Software](http://www.joelonsoftware.com/items/2007/10/26.html))
  + Bugovi koji se pronađu u testiranju se pišu na taj projekt u Jiri
  + Projekt **nije gotov** dok svi issuei nisu Closed - ne resolved nego closed
* Projekt Farmeron
  + Služi za vođenje bugova koji se javljaju na produkciji i za request sitnih featura (to može tražiti samo Matija, možda još netko)
  + Bug reporting
    - Upisati: verzija, steps to reproduce, expected result, actual result
    - Ispravak **ne smije** na produkciju dok issue nije Closed (tj istestiran)
      * Postavlja se pitanje regression testova kod hotfixa (x.y.**z**), no možemo poslije o tome - tu će pomoći automatizirani unit testovi
    - Kada se deploya funkcionalnost iz novog projekta, minor (x.**y**.z) verzija se uvećava
      * Svi otvoreni bugovi na prošlu verziju se moraju ispraviti na novoj verziji (jeli mogućž nekakav re-assign)