

BibTeXBible

Loppuraportti

Atte Juvonen
Teemu Koivisto
Eetu Mattila
Jesper Ruuth
Jim Saartia

Johdanto

Projektin tarkoituksena oli toteuttaa LaTeX -dokumentin lähdeviitteitä hallinnoiva sovellus, jonka avulla viitteitä olisi mahdollista luoda helposti ymmärrettävän käyttöliittymän avulla. Lisäksi, sovelluksen avulla olisi mahdollista exportata BibTeX standardin mukainen tiedosto, joka olisi myöhemmin liitettävissä erilliseen LaTeX-tiedostoon. Sovellus toteutettiin Java:lla, jossa käyttöliittymänä toimi konsoli-ikkuna.

Projektin eteneminen

Projekti toteutettiin viiden hengen tiimillä yhteensä kolmessa sprintissä, joista kaksi olivat viikon mittaisia ja yksi kahden. Suurin osa työskentelystä tapahtui etänä ja yhteistä työskentelyä tapahtui viikottain n. tunti jokaisen sprint planning meetingin jälkeen.

Definition of Done määriteltiin ryhmän toimesta seuraavasti:

- 90% rivikattavuus yksikkötesteissä
- Integroitu GitHubin master haaraan
- Viimeisimmän buildin "CI status" hyväksyvä

Sprintti 1

Ensimmäiseen sprinttiin valittiin Product Backlogista seuraavat User Storyt:

- *As a user I want to add reference to the document I'm writing*
- *As a user I want to be able to delete references so that they are permanently deleted*
- *As a user I want to be able to see all my current references, so that they are human readable*

Valitettavasti, yhtäkään user storya emme saaneet ensimmäisen sprintin aikana valmiiksi. Ongelmana oli lähinnä puuttellinen rivikattavuus (n. 50-70%) tiettyjen komponenttien osalta. Tähän vaikutti myös ensimmäisen sprintin tuoma alkukankeus, sekä puutteellinen yhteisymmärrys sovelluksen arkkitehtuurista. Ongelmista huolimatta, saimme tehtyä monta taskia “lähes” valmiiksi ja koimme, että seuraavassa sprintissä pystyisimme parempaan.

Sprintti 2

Toiseen sprinttiin valittiin Product Backlogista Sprint 1:n user storyt sekä uusina storyinä valittiin:

- *As a user I want to be able to export references in a BibTeX format, so that I can download it as a file*
- *As a user I want to copy my reference database to another machine*

Toisen sprintin aikana tiimin työskentely alkoi selvästi parantua, kun konsensus sovelluksen arkkitehtuurista alkoi olemaan paremmalla tasolla. Työtehtävien jakamisessa auttoi onnistunut tiimipalaveri, joka pidettiin sprintin aloitustapaamisen jälkeen. Valitettavasti emme vielä saaneet yhtään user storya valmiiksi, johtuen puuttuvista EasyB -testeistä, joten kolmanteen sprinttiin jäi vielä kiristettävää.

Sprintti 3

Kolmanteen sprinttiin valittiin seuraavat:

- *Aikasempien sprinttien keskeneräiset user storyt valmiiksi (puuttuvat EasyB -testit ja testikattavuus yli 90%)*
- *Sovelluksen julkaisu (Ladattava .jar-tiedosto)*

Kolmannen sprintin keskeinen päätös oli luopua relaatiotietokannasta ja toteuttaa viitteiden säilöntä “tiedosto import” ominaisuuden avulla. Haastavuutta aikataulutukseen toi myös koeviikko, sekä epätietoisuus sprintin kestosta. Saimme kuitenkin huimasti parannettua testikattavuutta ja koodin sisäistä laatua refaktoroinnin avulla. Sprintin lopussa valta-osa taskeista täyttikin definition of donen vaatimukset.

Loppuarvio

Mikä sujui hyvin

Ei sankarisessioita. Tiimin jäsenet pysyivät varsin hyvin neljän tunnin tavoitteessa. Itseorganisoituminen onnistui yllättävän hyvin, vaikka työskentelimme enimmäkseen etänä. Tiimin sisällä säilyi myös hyvä henki läpi projektin ja päädyimme lopulta aina yhteisymmärrykseen kehitystyötä koskevissa päätöksissä.

Mitä opimme?

Opimme projektin aikana käyttämään paljon hyödyllisiä työkaluja ja menetelmiä kuten:

- Git ja Github
- Flowdock
- Trello
- Burndown charts
- Scrumbut

Näiden lisäksi opimme myös uutta testauksesta, mockkaamisesta, tiimin sisäisestä kommunikoinnista sekä yleisestä ryhmätyöskentelystä, toisten koodin lukemisesta, sovelluksen arkkitehtuurin suunnittelusta sekä työmäärän arvioimisesta. Yleisesti opimme myös, että GitHubiin tehdyt commitit olisivat voineet vastata paremmin Sprint Backlogin tehtäväkortteja. Näin, commitit olisivat olleet loogisesti paremmin organisoitu laadunvalvontaa varten.

Mitä asioita olisitte halunneet oppia?

Tiimin yleinen mielipide oli, että vähemmän työkaluja syvällisemmin olisi ollut parempi kuin paljon työkaluja pintapuolisesti. Olsimme myös toivoneet koodikatselmointia ja tarkempaa palautetta sovelluksen rakenteesta tiimin ulkopuolelta jokaisen sprintin jälkeen.

Mikä tuntui turhalta?

Koska kyseessä oli varsin pieni sovellus, niin monessa kohtaan yksinkertainen asia jouduttiin tekemään turhan monimutkaisesti, jotta välttyttiin huonoilta oliosuunnittelu käytänteiltä. Osa tiimin jäsenistä koki myös, että suuri rivikattavuus ei tuntunut kovin mielekkäältä tavoitteelta testauksen tai sovelluksen laadun kannalta. Jotkin

teknologiat (esim. easyB) olivat myös hieman vaikeita debugata Lisäksi, aikaa tuhlautui turhan paljon ääkkös ominaisuuteen, johtuen osin huonosta standardoinnista, ilman varsinaista oppimishyötyä. Myös neljän tunnin työmäärä viikossa koettiin osin hieman rajoittavaksi tekijäksi.

Liitteet

JAR

<https://github.com/ohtu/bibtexbible/releases/download/viikko3/bibtexbible-1.0-SNAPSHOT.jar>

Trellon Task Boardit

Sprintti 1: <https://trello.com/b/VGq3K1EW/sprintin-1-backlog>

Sprintti 2: <https://trello.com/b/apijcep2/sprintin-2-backlog>

Sprintti 3: <https://trello.com/b/6fFskbpW/sprintin-3-backlog>

EasyB-esimerkki:

Story: add_reference

User can add reference and see it listed

```
scenario user can't list references when there is none
  given application is initiliazed correctly
  when correct commands are entered
  then user can see see no references listed
```

```
scenario user can add reference
  given application is initiliazed correctly
  when correct commands are entered
  then user can see reference listed
```

```
scenario user can't add duplicate reference
  given application is initialized correctly
  when correct commands are entered
  then user can see see only ONE reference listed
```