

Deliverable #5 Report

Team #2 SQL Injector

CSCC01H3 INTRODUCTION OF SOFTWARE ENGINEERING

2015-11-24

Nadeem Haidar

Junaid Patel

Andres Moreno

Minsoo Park

Yujian Chen

Deliverable #5 Report

Team #2 SQL Injector

Sprint / Product Backlog, Burndown Chart, and Taskboard Snapshots

- This information can be found from GitHub repository
- For Burndown Chart Snapshots, these can be found on Trello board

Code inspection

- Reports can be found in GitHub repository

Sprint Plan

- *Initial plan*

Same as the previous iteration, each team member has planned 4 hours per week to work on the project. That makes the total of 40 developer hours for this two-week iteration. Chart shows below is the detail planning of our team could work on each day. Number in the bracket indicates the task that each team member should work on.

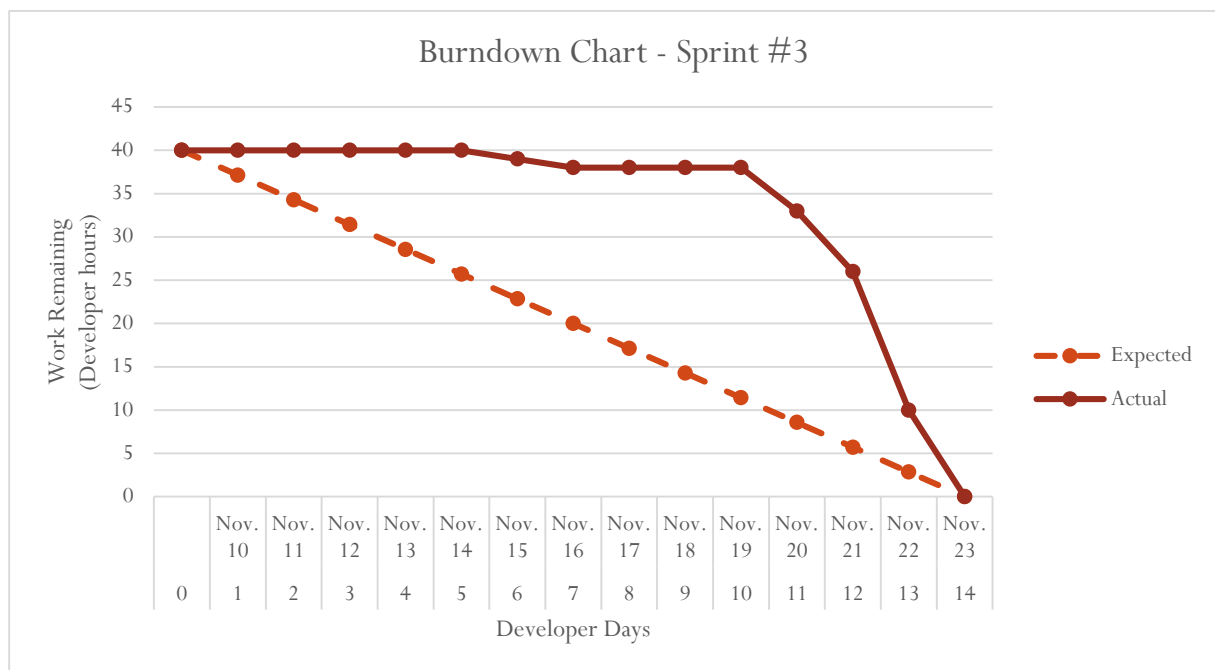
Date	Nadeem	Junaid	Andres	Minsoo	Alex	Total
	0	0	0	0	0	0
Nov. 10	0	0	0	1 (t2)	1 (t2)	2
Nov. 11	0	0	0	0	0	0
Nov. 12	0	0	0	0	2 (t4)	2
Nov. 13	0	2 (t2+t3)	0	1 (t5)	0	3
Nov. 14	2 (t2+t3)	2 (t5)	1 (t2)	1 (t8)	1 (t9)	7
Nov. 15	2 (t6)	0	1 (t4)	1 (t8)	0	4
Nov. 16	0	0	0	1 (t8)	0	1
Nov. 17	0	0	0	1 (t8)	0	1
Nov. 18	0	0	0	0	0	0
Nov. 19	0	0	0	0	0	0
Nov. 20	0	0	1 (t4)	0	0	1
Nov. 21	2 (t9)	0	1 (t11)	1 (t1)	0	4
Nov. 22	2 (t10)	2 (t7)	2 (t11)	1 (t1)	2 (t1)	9
Nov. 23	0	2 (t10)	2 (t11)	0	2 (t1)	6

● *Actual Logs of the plan*

We were unable to follow this plan. However, we are able to finish all tasks we decided to do in this iteration. This mainly due to the reason that our implementations have unexpected bugs, difficulties migrating the third party library in our plugin, and The chart of daily log will show below:

Date	Nadeem	Junaid	Andres	Minsoo	Alex	Total
	0	0	0	0	0	0
Nov. 10	0	0	0	0	0	0
Nov. 11	0	0	0	0	0	0
Nov. 12	0	0	0	0	0	0
Nov. 13	0	0	0	0	0	0
Nov. 14	0	0	0	0	0	0
Nov. 15	0	0	1	0	0	1
Nov. 16	0	0	0	1	0	1
Nov. 17	0	0	0	0	0	0
Nov. 18	0	0	0	0	0	0
Nov. 19	0	0	0	0	0	0
Nov. 20	0	0	3	1	1	5
Nov. 21	2	1	0	3	1	7
Nov. 22	6	4	1	2	3	16
Nov. 23	0	3	3	1	3	10

● *Burndown Chart*



- *Current state of the project*

The current state of our project is we finish what we planned for this iteration. We keep delivering important feature and functionality done during this iteration. At the beginning of this iteration, we decided we will finish the leftover editing tags from batch-editing feature, and start working on customize output style feature, which is our **B04** and **B05** from our sprint backlog. Taskboard snapshot in GitHub will reflect what we planned. During our researching, we found out on GitHub, there is a open source project call csl-editor is designed for customizing output styles as exactly stated in our **B04** and **B05** user story. However, there are some modification need to make in order to make this library work in our plug-in: It requires an in-app browser in order to load the library locally or from server, a server that can finish our request, an option that install the style file right into Zotero from the library without user downloading the csl file and install it manually.

The largest problem we have so far in this iteration is in the csl editor, it depended on flash which is not included in Zotero standalone application, we need to modify the code to pull out the csl file without using flash to finish that, means that it might break the functionality of the library.

B04: As John, a researcher, I want to be able to add or remove fields from a specific output style and modify the format of the style, so that I can match the requirements requested by my professor.

B05: As John, a grad student, I want to be able to create new output styles without modifying the original ones, so that I can keep the original ones intact.

- *System design*

We modified our system design in this iteration since we introduce a third party library and server in order to make our customized output style working. Batch editing remains the same as stated in last deliverable report: The batch editing component have a backend logic (**ExtBatch.js**) for removing, renaming, adding, editing, merging tags and pulling data in Zotero system and Zotero database. The UI file (**ui.xul**) contains all layout and load all user requirements and information by calling function in the plug-in core (**zoteroEXT.js**).

For customizing output styles, it builds up with several components: The plugin will call for an embedded browser for loading our plugin, this browser will contact our server where stores our source code to send and receive query. After user finish searching and editing, plugin will export a citation styles file (*.csl) and install this file automatically to Zotero.

