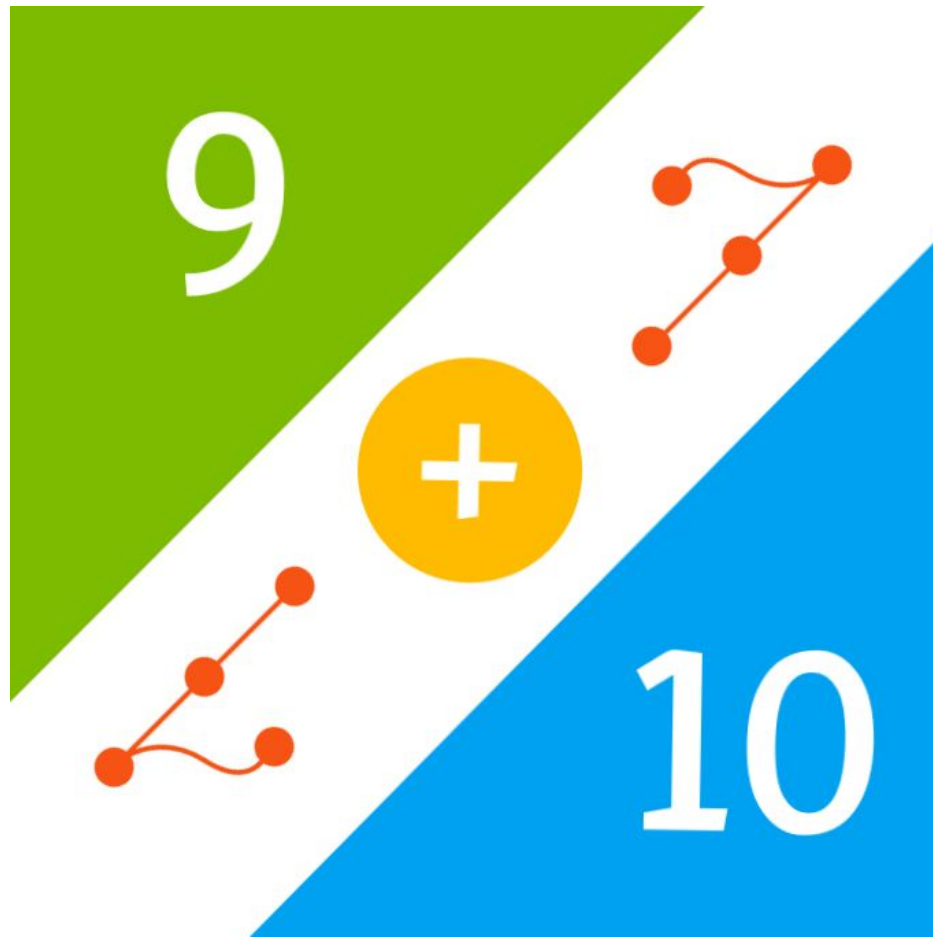


# CSCC01 Introduction to Software Engineering

Project Deliverable 4 - Verification and Code Review



*Team 9 + 10*

**Shruti Mistry**  
**Dennis Tismenko**  
**Abbas Rattansi**  
**Qi Lin**  
**Raya Farhadi**

<b>Sprint Tasks</b>	<b>3</b>
<b>Sprint 3</b>	<b>5</b>
Plan	5
Report	5
Burndown Chart	5
<b>Sprint 4</b>	<b>6</b>
Plan	6
Report	6
Burndown Chart	6
<b>Acceptance Test:</b>	<b>6</b>
<b>Code Review Strategy Checklist</b>	<b>9</b>
<b>Code Review Summary</b>	<b>10</b>

# Sprint Tasks

User Story	Task	Dependency	Story Points	Description
9	9a. Create a "data entry" tab.		4	Create a template selector (choiceBox) and present an upload button and a status bar that reflect upload process.
2	2c. Create expandable column header with submitting and cancel button on it.		3	
10	10a. View datasets in their templates		3	Search the template, once it is selected, present the current dataset of it as a table.
9	9b. View and upload new datasets	9a	4	Create a table to view uploaded data (before submitting). Create a submit button to submit data to database.
2	2e. Create existing fields in the database from the templates provided		3	Implement premade columns/fields to the database from the excel/template they provide
2	2f. Pre-upload all templates given		2	
2	2h. Populate database with created template		3	Connect the created template to the database by populating all related fields
3	3a. Templates page		2	
3	3c. Retrieve the data about the agency		2	Allow every user to view their current account information

2	2a. Create a user management tab		4	Users can search for a user and display their account information.
14	14a. Generate mock data for templates		3	

## Sprint 3

### Plan

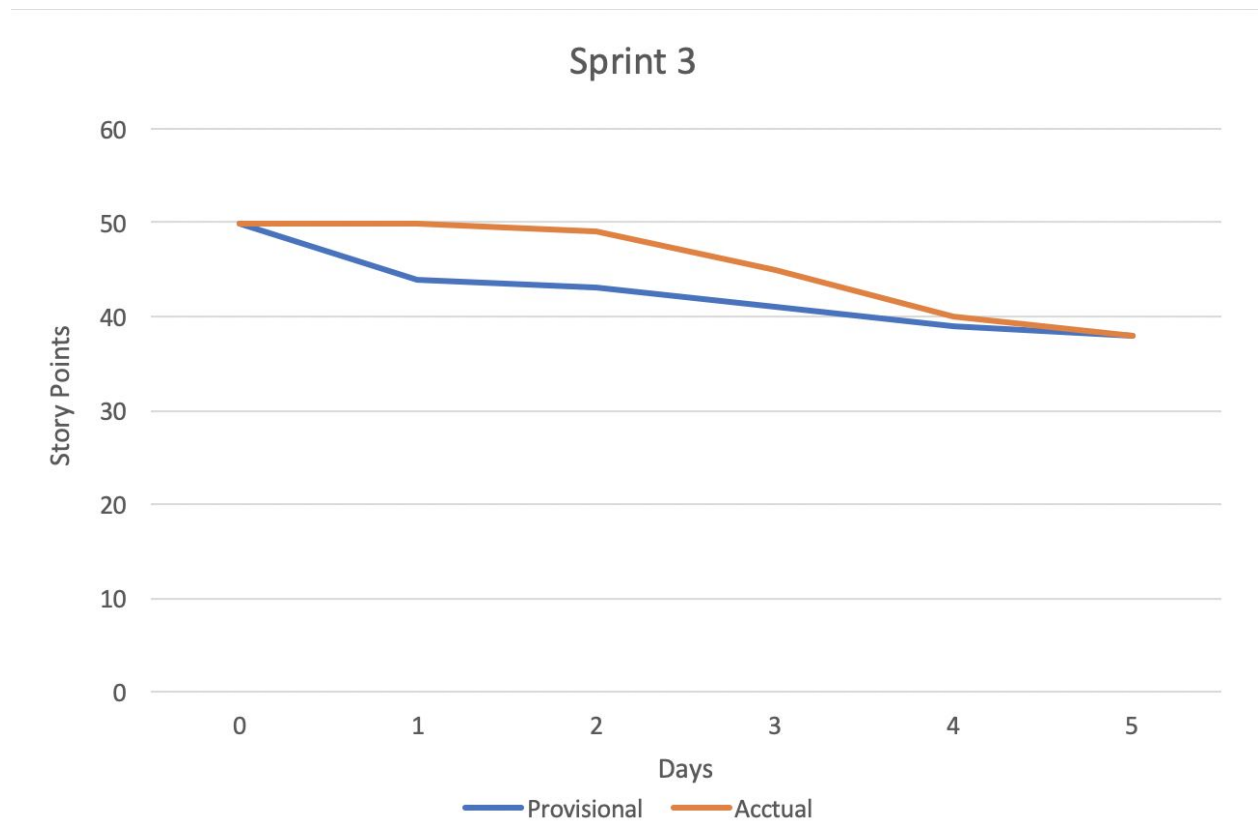
Sprint 3 Plan	2 hour per developer = 5*5*2 = 50									
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5		
2	2e		3 D:1	D:1	D:1					
2	2f	2e	2		D:1	D:1				
2	2h		3 Q:1						*2 story points completed in sprint 2	
3	3a	2f	2			D:1	D:1			
3	3c		2							

### Report

Sprint 3 Execution	2 hour per developer = 5*5*2 = 50									
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5		
2	2e		3	D:1	D:2					
2	2f	2e	2		D:2					
2	2h		3			Q:1			*2 story points completed in sprint 2	
3	3a	2f	2			D:1	D:1			
3	3c		2			A:1	A:1			

### Burndown Chart

Burndown	0	1	2	3	4	5
Provisional	50	44	43	41	39	38
Acctual	50	50	49	45	40	38



## Sprint 4

### Plan

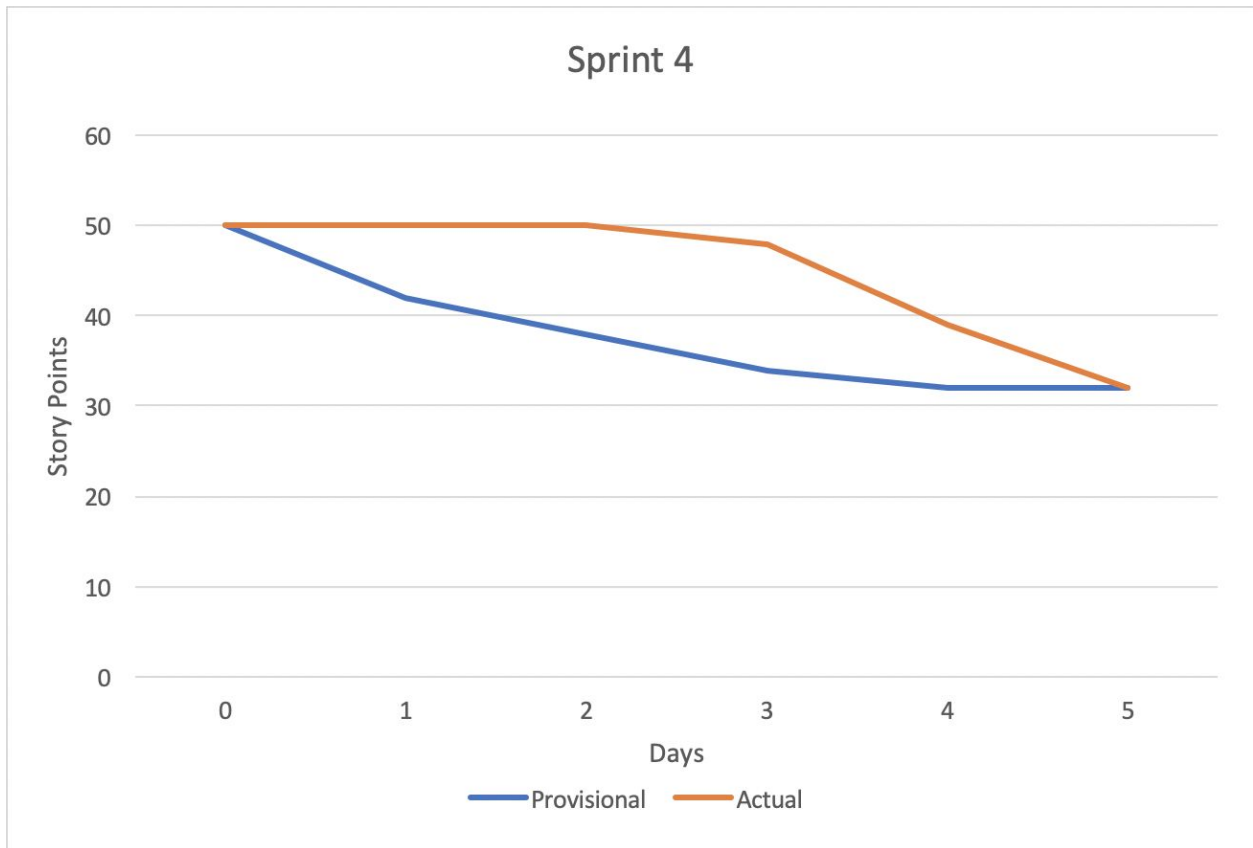
Sprint 4 Plan	2 hour per developer = 5*5*2 = 50									
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5		
2	2a		4 R:1	R:1					*2 story points completed in sprint 2	
2	2c		3	Q:1	Q:2					
2	9a		4 S:2	S:2						
3	9b	9a	4		D:2	D:2				
3	10a		3 D:2	D:1						

Report

Sprint 4 Execution		2 hour per developer = 5*5*2 = 50												
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5						
2	2a		4				R:2		*2 story points completed in sprint 2					
2	2c		3				Q:2	Q:1						
2	9a		4			S:2	S:2							
3	9b	9a	4				D:1	D:3						
3	10a		3					D:3						

Burndown Chart

Burndown	0	1	2	3	4	5
Provisional	50	42	38	34	32	32
Acctual	50	50	50	48	39	32



# Acceptance Test:

**Note:** If the User wants to close this application, they can always click the close on the top left corner of the page.

1. Loading page:
  - a. It should show the name of this software in the middle of the page
  - b. It should show the user a loading icon and tell the user that it is currently fetching data from database.
  - c. After it finishes the fetching process, it goes to the login page.
2. Login page:
  - a. It should show the user two text fields (username & password), and a login button.
  - b. Users can enter their username and password for the corresponding fields. After that, Users must click the login button to login to their account and access the features.
  - c. If the User enters the wrong username or password, the software should show user a message: Authentication Failure. User are allowed to try it numerous times.
3. Landing Tab
  - a. Depending on the user's role, each user will see a different set of tabs(Feature).
    - i. As admin/teq, you will see three tabs.  
Given Account: UserName: teq Password: teq
      1. Account creation Tab
      2. Template Editor Tab
      3. Data Browser Tab
    - ii. As Utsc  
Given Account: UserName: utsc Password: utsc
      1. Template Browser Tab
      2. Data Browser Tab
    - iii. As Agency  
Given Account: UserName: agent Password: agent
      1. My Data Tab
  - b. User can click the header of each tab to switch tabs.
  - c. When User switch tabs, the entered field will not be clear, User can feel free and switch back to original tab to continue his/her work.
  - d. Once the user finishes their work, they can click the logout button(shown on the top left of the page) to logout. It should leads the user back to the login page.
4. Account creation Tab

- a. It should shows user three text fields for entering new user's UserName, Password, Email.
- b. It should shows user a dropdown box that contains 6 role: TEQ High-level, TEQ Mid-level, TEQ low-level, Agency, UTSC Management, Administrator.
- c. When select Agency, a new text field should appear below for user to enter the new user's corresponding agency name. This Text field should only be showing when user select the Agency. If User select any other role, this text field should be hidden.
- d. In addition, the value entered in the Agency Name text field will be set to empty when user select other roles(Not Agency).
- e. It should also show user two button: Submit, clear
- f. User can click submit button to create a new user
- g. User can click clear button to clear all the text field and selection field

#### 5. Template Editor Tab

- a. It should shows User an empty table and a dropdown box selection field, four buttons.
- b. Currently, there are total 8 template available for the selection.
- c. User can select the one of template in the dropdown box. So that the table can load the selected template from database.
- d. User allows to select on column at time.
- e. User can double click the header of each column to edit its name
- f. User can click the "..." Button at the left side of table, it will add a new column to current table. Note: It will set column name to be default value: "New Column".
- g. User can click the "." Button at the left side of table, it will delete the selected column form the table.
- h. After User make all his/her changes. They can click the save icon button on the bottom right of the table. It will save the changes on the template to the database.
- i. User can also click the Cross icon button beside the Save button. It will clear the current changes and set the table back to its previous state.
- j. When the template has too many columns, the table should present a horizontal scroll bar for user to drag. So that User can drag and view all the columns in the current template.

#### 6. Data Browser Tab

- a. It should show user a dropdown box for agency selection.
- b. The drop down box should contain all the agency name in the current database
- c. Once the user select the agency name, it will shows another drop down box for template selection. It should contain 8 type of template
- d. Once the user select the template. It will shows table below the dropdown box.
- e. After the above selection, the table will load the selected agency's selected template data.
- f. The data shown in the table are not editable, it is only for the viewing.
- g. If the selection for template changes, the table below will load the new selected template data.



- h. If the agency selection changes, then the template selection drop down box will be set to empty and the table will be not longer be visible until User select the template.
- 7. Template Browser Tab
  - a. It should shows User an empty table and a dropdown box selection field
  - b. Currently, there are total 8 template available for the selection.
  - c. User can select the one of template in the dropdown box. So that the table can load the selected template from database.
  - d. Table on this tab should not be editable.
  - e. When the template has too many columns, the table should present a horizontal scroll bar for user to drag. So that User can drag and view all the columns in the current template.
- 8. My Data Tab
  - a. It should show user a dropdown box for template selection.
  - b. Currently, there are total 8 template available for the selection.
  - c. After User select a template, it will shows a table that contains the selected template data for the the user's agency.

## Code Review Strategy Checklist

### Functionality

- Can I understand what the code does?
- Does the code do what it is supposed to do?
- Is there some dead code or unnecessary debug statements?
- Do we handle the exceptions and errors correctly?

### Testing

- Can I understand what the tests do?
- Do I understand the tests results?
- Do we have a good test coverage?

### Design

- Does the code corresponds to our original design?
- Does the code follows the SOLID principles?
- Can we better structure the code using design patterns?

Readability and Maintainability:

- Is the documentation good enough?
- How is the coding style?
- Do we follow appropriate naming conventions?
- Do we have bad copy and paste of code?
- Do we use constants whenever it is possible?

## Code Review Video

<https://drive.google.com/file/d/1hNOFlovmFxstxMK7KVSGUodHm6SgwnXv/view?usp=sharing>

## Code Review Summary

**Code: Abbas**

**Reviewer: Shruti**

**Topic: Account Creation**

### **Summary:**

The code is functional and is easy to understand and follow despite having one unnecessary debug statement. In terms of Design, the feature implemented uses the Single Responsibility Principle where each class only performs one task. The classes created follow our original design and are placed in appropriate packages. Going forward, the Testing and Documentation can be improved. Test cases aren't written for the feature, which can make it difficult to discover bugs. If the implementation were to be changed, the code would have to be re-tested which is a huge downfall. Although the code is easy to understand, brief documentation can also be included. Naming Conventions are followed for the most part, where variable names are chosen appropriately.

**Code: Dennis**

**Reviewer: Raya**

**Topic: Template Creation, Template Viewer, Login, Data Viewer**

### **Summary:**

Overall, the code is well written, readable, and easy to understand. The code follows MVC design principles making it easy to work with and expand on. However, there are a couple

of areas that I think require better documentation. For example, the use of a lambda function in the TemplateViewController and DataViewController are difficult to understand without more detailed comments. For the most part, method and variable names are descriptive, however, there are a few cases where there are single letter variable names that should be renamed to help others understand that portion of the code better.

**Code: Qi**

**Reviewer: Abbas**

**Topic: Account Creation, Template view**

**Summary:**

The code review was completed on November 9th. The code written for the various frontend portions of the system were concise, well written and were easily readable to connect the backend to frontend of the application. For the future, better documentation would be better, for someone who has not had previous experience with the backend or the frontend of the application, the code may seem confusing and some stronger documentation would be suffice.

**Code: Shruti**

**Reviewer: Dennis**

**Topic: Data Entry upload**

**Summary:**

Code is functional and easy to understand and performs what it needs to do despite having some unnecessary debug statements. In a particular case worth noting, the code was not checking for null values before performing operations on a object. Likewise, array/list bounds were not being checked properly, which could potentially lead to "OutOfBounds" exceptions -- use of iterator pattern would have been preferred. Code follows our MVC design principle and optimally utilizes a service to perform tasks in separate thread which notifies the UI via observable pattern. While documentation is lacking, code is fairly straightforward so heavy documentation is not required. Naming conventions were mostly followed, with a single exception of "camelCase" not being followed and some cases of variables not accurately reflecting its purpose. Constants should have been used to greater effectiveness, and single method could have been broken down into smaller tasks for easier debugging and testing.

**Code: Raya**

**Reviewer: Qi**

**Topic: User Management Tab**

**Summary:**

The Code review was completed on Nov 9th, 2018. In term of overall design of the feature, it is pretty well done. Except the code doesn't follows the open and close principle. If we want to add an extra column to it, we have to modify the object holder and all the class related to it. It may be hard for us to add some more feature to it. In term of readability, all the variable/method's name are explicit enough. It improves the overall readability, but it is poorly documented. So maybe for someone who doesn't have any technical background, it will be very painful for them to read through the code. In term of functionally, the code is efficient enough, I did not see it takes any unnecessary step.