IFB299

A critical analysis of an application design and development sprint.

Retrospective – A Critical Review

Sprint 1

Contents

[Table of Tables: 1](#_Toc493885553)

[Table of Figures: 2](#_Toc493885554)

[1: Team Communication 2](#_Toc493885555)

[1.1: Positive Aspects: 2](#_Toc493885556)

[1.2: Aspects Decreased the Quality of Communication: 2](#_Toc493885557)

[1.3: Countermeasures to Correct Negative Aspects: 2](#_Toc493885558)

[2: Effective Team participation 2](#_Toc493885559)

[2.1: Positive Aspects: 2](#_Toc493885560)

[2.2: Negative Aspects of Team Participation: 2](#_Toc493885561)

[2.3: Countermeasures to Correct Negative Aspects: 3](#_Toc493885562)

[3: Quality control 4](#_Toc493885563)

[3.1: Client Review of Quality: 4](#_Toc493885564)

[3.2: Computer Science Development Team Member Review of Quality: 4](#_Toc493885565)

[3.3: Meeting Acceptance Criteria (Quality Check): 5](#_Toc493885566)

[3.4: Positive Aspects: 5](#_Toc493885567)

[3.5: Negative Aspects of Team’s Quality Control: 6](#_Toc493885568)

[3.6: Countermeasures to Correct Negative Aspects: 6](#_Toc493885569)

[4: Keeping Client Informed: 6](#_Toc493885570)

[4.1: Positive Aspects: 6](#_Toc493885571)

[4.2: What was Done Wrong: 6](#_Toc493885572)

[4.3: Countermeasures to Correct Negative Aspects: 6](#_Toc493885573)

[5: Issue Handling (Review): 6](#_Toc493885574)

[5.1: Positive Aspects: 6](#_Toc493885575)

[5.2: What was Done Wrong: 7](#_Toc493885576)

[5.3: Countermeasures to Correct Negative Aspects: 7](#_Toc493885577)

[6: Issues: 7](#_Toc493885578)

[6.1: Issue 1: 7](#_Toc493885579)

[6.2: Issue 2: 7](#_Toc493885580)

[6.3: Issue 3: 8](#_Toc493885581)

# Table of Tables:

[Table 2 Client Review 3](#_Toc493838869)

[Table 3 Computer Science Review 4](#_Toc493838870)

[Table 4 Acceptance Criteria Review 4](#_Toc493838871)

# Table of Figures:

[Figure 1 Burndown Chart 2](#_Toc493838884)

# 1: Team Communication

## 1.1: Positive Aspects:

* The team communicated well through the course of the first sprint and actively communicated consistently through the use of messaging applications and met as a team. With the team meeting 7 times (excluding tutorials) over the whole sprint.
* There were no massive periods of silence from any group members with all members contributing to group discussions and were reachable at all times.

## 1.2: Aspects Decreased the Quality of Communication:

* Group members did not actively discuss and document their availability (when they were free or busy) for team meetings (or did not honor team meetings).
* There was some miscommunication at times that created difficulties when planning weekly work and getting updates from team members.

## 1.3: Countermeasures to Correct Negative Aspects:

* Keep each other informed of schedules with work commitments so that we can plan meetings and reach deadlines more effectively.
* With this information plan team meetings at least a week in advance, so that team members can plan their daily activities around this.
* Create a more visual method of keeping track of weekly schedules that can be accessed easily (such as an excel spreadsheet/graph). This will make it easier to plan around each member's schedules.

# 2: Effective Team participation

## 2.1: Positive Aspects:

* We actively informed our tutor of updates when required through the use of emails and through the GIT repository where the tutor could view the work done.
* All team members contributed to the completion of most work to some capacity (minimum contribution included overviews or work and feedback or the group starting work together).
* Most of the work was delegated evenly between members of the group so all group members had roughly the same workload during the sprint.

## 2.2: Negative Aspects of Team Participation:

* There were some instances where members were unable to take the initiative to find work or stay up to date on work that needed to be completed each week.
* It was found that there was minimal work and participation in the first 2 weeks of the sprint (this is shown in the burndown chart below and the ‘major issues’ retrospective file).
* There was not a sufficient level of participation in actively keeping files updated in the Git repository, with only 2 contributors for the majority of the sprint. Many of the participants used Google Drive to keep documentation and files updated during the early stages of the sprint.

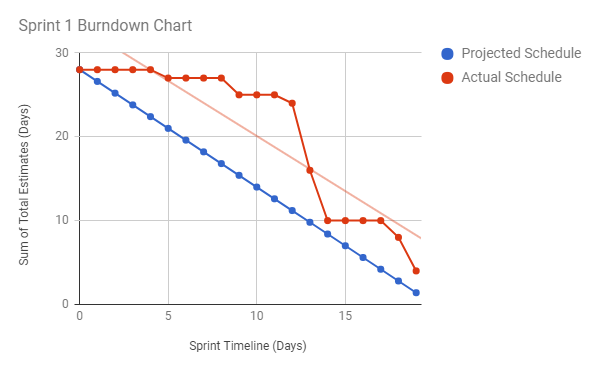


Figure Burndown Chart

## 2.3: Countermeasures to Correct Negative Aspects:

* Ensure that work to be completed is more clearly documented so that all members of the group can view and discuss deadlines.
* Work that each member has and plans to get done will also be documented in a single file, which will ensure that its shared evenly amongst the team.
* Make efforts to keep the progression of work more consistent over the 4 weeks rather than at the end of the sprint.
* This will be beneficial to keep the team informed, and force members to actively participate in tasks.
* This will be done by assigning more important tasks earlier on in the sprint to team members.
* Make sure that all members understand how to use Git to ensure all group members have the most updated version of the project and can upload their work as they complete it.
* Ensure one member creates one set of files for all members to use. The one set of files must be uploaded to Git as soon as possible in order to prevent file name conflicts, as well as making it easier to communicate with the team.
* Ensure all members use the Git terminal or upload through the Git GUI on the website to the correct folder, otherwise errors may occur to those using the terminal over the GUI.

# 3: Quality control

## 3.1: Client Review of Quality:

|  |  |  |
| --- | --- | --- |
| Quality Standard: (Client/Information Systems Perspective) | Quality Control Monitor’s Rating (1-10): Jamie | Justification: If Applicable |
| Fulfilling Acceptance criteria (of solution) | 7 | Did not fulfill Story ID 3 in sprint  Did not fulfill Story ID 30 in sprint  5/7 complete  = 71.42857142857143  =7/10 |
| Visual Quality (of solution) | 7 | Meets expectations, may look outdated as it is an early version of software |
| User Experience Quality (of solution) | 7 | Meets expectations for early development, more details on what to improve will be in the Client Review Documentation. |
| Deliverables Quality (Models, meeting notes) | 8 | Meeting notes well made.  Models checked by most members and were used in meetings, models are good quality, and the models were created by a variety of members.  Some parts of the models, there is partial errors in logics/industry standards due to the Team’s lack of knowledge in the modeling techniques. |

Table Client Review

## 3.2: Computer Science Development Team Member Review of Quality:

|  |  |  |
| --- | --- | --- |
| Quality Standard: (Computer Science Perspective) | Quality Control Monitor’s Rating (1-10): | Justification: If Applicable |
| Code Quality (of solution) | 8 | The quality of the code is a high standard. Functions, indentation and general code structure was implemented well. |
| Presence of bugs (of solution) | 6 | At this point in the project there are a few bugs related to registering accounts, along with what is outlined in the Client Review Documentation. However, from our tests we have not encountered any bugs in other areas of the website. |
| Visual Quality (of solution) | 6 | The website’s user interface looks appealing in all forms, however more styling can be applied to improve the design overall and meet client expectations. |

Table Computer Science Review

## 3.3: Meeting Acceptance Criteria (Quality Check):

|  |  |
| --- | --- |
| Sprint User Story: | Did it meet the Acceptance Criteria? |
| 4 | Completed. All criteria were met, as they are functional and in the registration page. |
| 3 | Not fully completed. All 3 criteria were ‘met’ for user story 3. However, the three user types are not fully functional. |
| 11 | Completed. All 3 criteria were met. The user can log in through the home page, there is a dedicated login page, and there is also a link in the navigation bar that redirects the user to a registration page. |
| 30 | No. This story point has been pushed back to sprint 2, due to the lack of time. |
| 15 | Completed. The map allows for all users to access city information such as parks, zoos, malls, museums and so forth. |
| 8 | Completed. Django already completes the majority of this for the developers. Admins can access the accounts and look at the hash, salt, and encryption algorithm. |
| 13 | Completed. The way the programmers coded the software was flexible. This was achieved through the use of classes, separate files that can be imported, and the use of functions. |

Table Acceptance Criteria Review

## 3.4: Positive Aspects:

* The documentation was set up well which allowed us to easily monitor the quality of work.
* Everyone actively helped with the quality control and checked other members work where required.
* The team (as shown in the release and sprint planning + user stories) put measurable (pass/not pass) Acceptance Criteria on all User Stories. Before any user story was considered ‘complete’ in the Release and Sprint Plan, a check was performed to check that the Acceptance Criteria was completed. If the Acceptance Criteria was not met, the feature was refined to assure compliance with the Criteria.

## 3.5: Negative Aspects of Team’s Quality Control:

* Throughout the sprint, we did not always officially follow and/or complete documentation which made it hard to track work that was performed.
* There was insufficient knowledge in some areas to actively provide good feedback on completed work which means that some deliverables were not up to a high level of quality.
* The acceptance criteria were not referred to when the team was creating the solution, only after during review process – and ultimately some acceptance criteria were not met (Story 3 and Story 30).

## 3.6: Countermeasures to Correct Negative Aspects:

* Actively require people to follow the documentation and ‘sign off’ when they check the quality of work which will allow the team to track work better.
* Put more focus on gaining a deeper understanding of the framework when unsure about a specific piece of work to ensure an error free/high standard of work.

# 4: Keeping Client Informed:

## 4.1: Positive Aspects:

* The client was present in most or all meetings of the meetings.
* Had weekly checklist to assure that the Client was always included in meetings and being informed week by week.

## 4.2: What was Done Wrong:

* The client was not formally informed (through the use of documentation) throughout the project of the work done in each week, rather they were informally informed through face-to-face meetings and messaging applications.
* Client was present at meetings but there wasn’t a high level of specific, documented Client - Development Team feedback being discussed. Rather the meetings were centered around discussion between all parties.

## 4.3: Countermeasures to Correct Negative Aspects:

* Customer will be formally updated on the team's progress through weekly emails to tell the client about the proceedings for that week.
* Both the team and the client will do a better job during meetings to provide constructive feedback (client will provide criticism on the project which will be considered by the development team).

# 5: Issue Handling (Review):

## 5.1: Positive Aspects:

* When issues occurred, the team openly admitted that there were issues with workload being unbalanced (made efforts to rectify this) in a formal and professional manner.
* The team had set up appropriate documentation to monitor quality.
* All issues that arose were resolved quickly through discussion as a team with little professional conflict and no personal conflict.

## 5.2: What was Done Wrong:

* The team did not put countermeasures in place to stop issues before they arose.

## 5.3: Countermeasures to Correct Negative Aspects:

* Record things that are at risk of becoming issues and monitor them (take appropriate measures to counter them if necessary).
* Assure that there are more frequent checks on current issues and potential future issues by all members of the team.

# 6: Issues:

## 6.1: Issue 1:

**Issue Title:** behind on sprint completion

**Date:** 30/07/2017

**Issue Reporter:** Jamie Kostaschuk, Caleb Spencer

**Issue Description:** there was minimal work completion down on the sprint workload during the first 2 weeks of the sprint. This was due to a lack of initiative by all members and poor planning around learning Django.

**Steps Taken to Resolve Issue:**

Communicated with the team, and the team believes that the workload can be made up for in subsequent weeks

The sprint completion will be monitored, and if necessary, the sprint workload will be shifted at a later date

## 6.2: Issue 2:

**Issue Title: Unbalanced workloads**

**Date:** 9/09/2017

**Issue Description:** It has been found that there is a dip in the contributions between the team members, though there is a positive and healthy atmosphere one member has not been assigned, or has not been able to contribute as much as other members.

**Issue Reporter:**

Caleb

Jason

Jamie

**Involved Members:**

Michele

**Steps Taken to Resolve Issue:**

The team has admitted this, and has made an active effort to assign tasks to this member more, to assure compliance with the team agreement.

## 6.3: Issue 3:

**Issue Title:** not enough use of GitHub

**Date:** 9/09/2017

**Issue Description:** It has been found that there is not a sufficient level of the GitHub repository use by the team, with only a few members actively using the repository. When the team members did you Git, they did not upload to the correct folder, resulting in undesirable outcomes for those using the terminal

**Issue Reporter:**

Caleb

Jason

**Steps Taken to Resolve Issue:**

The team has been informed of this, and the benefits that GitHub provide has been accentuated to the entire team

All members have agreed to start using the GitHub properly

Ensure all members upload to the right branch and/or folder if they are uploading via the Git GUI. Alternatively, they can upload through the use of the Git terminal.