

TEAM EVALUATION: PLAN AND OUTCOME

Project Client on Board

Date: 2023/01/28

Group: Brave Alligators

Authors:

Adrian Krantz – 524202

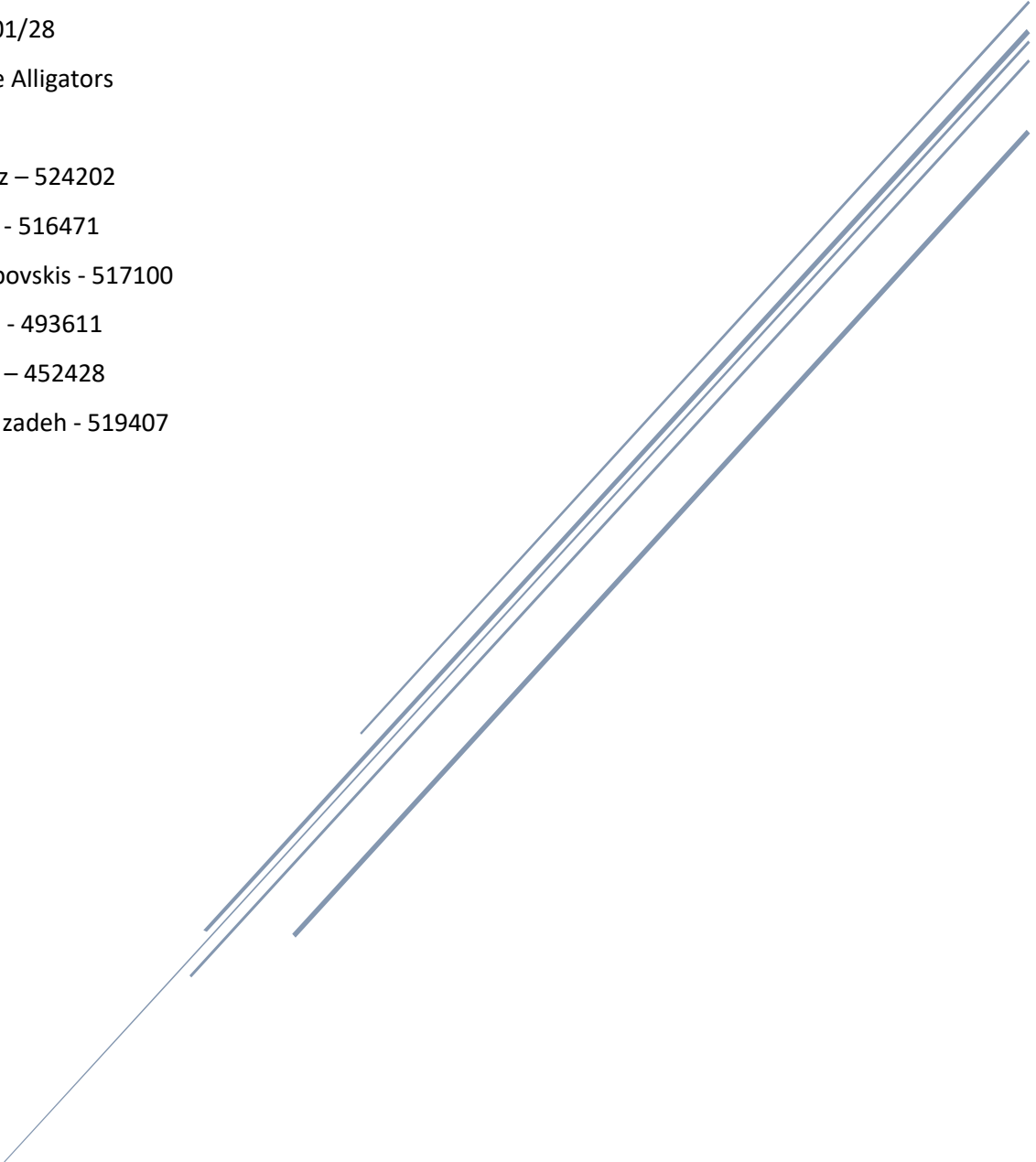
Darius Bejan - 516471

Georgs Jakubovskis - 517100

Hai Ha Pham - 493611

Tiffany Deng – 452428

Sadra Samadzadeh - 519407



Preface

This evaluation document aims to report on how well the execution of the team plan and the Sprint plans took place during this project, on both technical level and process level. Successes and failures will be analyzed and explained, resulting in lessons learned to improve in the future.

"A document called "Plan and Outcome" should in my opinion contain an honest report on things that you planned to do, things that succeeded (and why did they succeed?). But also, things that did not succeed, a reasonable explanation of why that happened as well as lessons learned on how to avoid that in the future."

-Frederik Bonte-

Evaluation of Team Plan – Process

In this section, execution of the Team Plan outlined at the start of the project is evaluated. This includes the Code of Conduct and the Code Quality – Programming Etiquette.

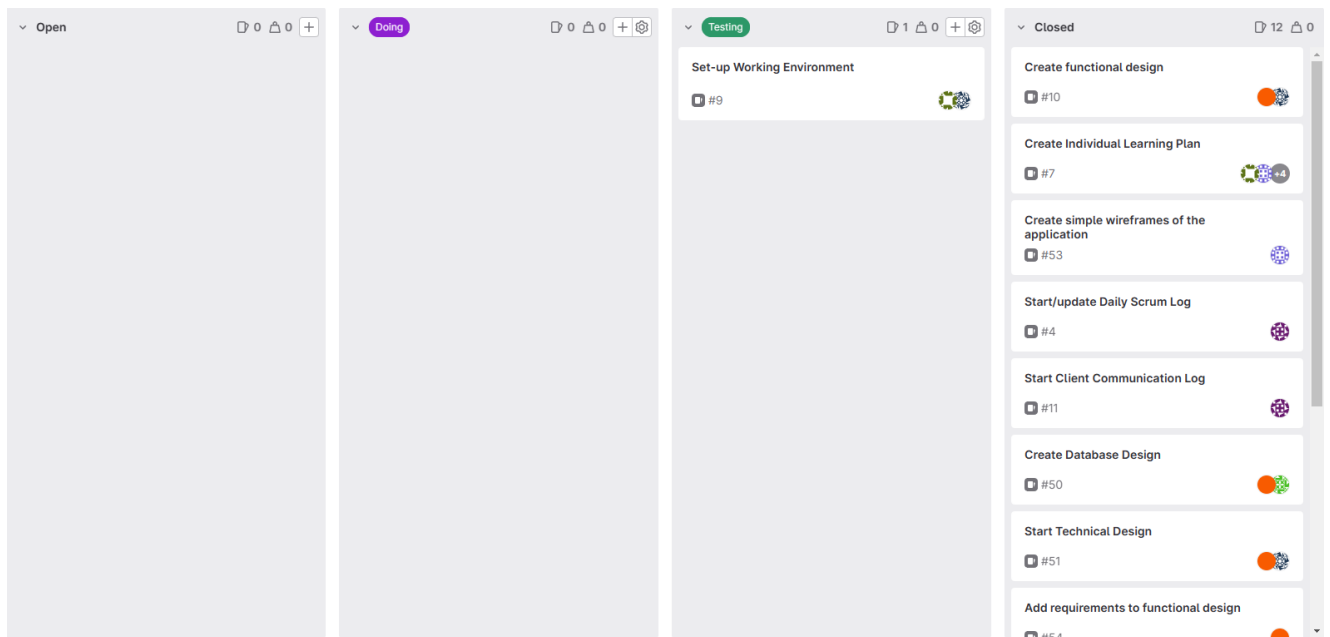
Criterion	Succeed/Failed	Explanation, Elaboration & Remarks
“Scrum standup meeting every 2 days, to keep the team on track and log these meetings as deliverables. We answer 3 standard questions and decide if someone needs to shift their task around (and to who).”	Succeed	The team maintained standup procedure regularly throughout the project. This is demonstrable in the standup logs. Every member showed reasonable professionalism that is expected of second-year students to fulfill this criterion.
“Show up at lab sessions and scheduled group meetings. Notify the group beforehand in case of absent.”	Succeed	The team did not experience a single situation where a member went absent without notice to the team, which indeed make this criterion fulfilled. Every member showed reasonable professionalism that is expected of second-year students to fulfill this criterion.
“Show reasonable progress, be reliable. If you get stuck at something, ask the team for an explanation or help. If you cannot do it at all after (seriously) trying, move on, pass the task on to another member. There is no shame in not knowing something – we are all here to learn.”	Succeed	See ‘Evaluation of Sprint Plannings’ chapter
“Follow the Git etiquette to keep the code base manageable “	Succeed	Every feature of the application was developed on a separate feature branch. The Develop branch was carefully managed, kept clean and always contained only working functionalities. Commits were written in a way that is easy to understand.
“As suggested in Project Details (Blackboard), we use the 3 strikes out system, where a strike is a formal	N/A	Since the team remained professional throughout the project and no issue that warranted a strikeout arose,

<p>warning, a second strike has grading consequences and the third means direct exit from the project. The team discuss this option together and decide with the teacher if necessary.”</p>		<p>this option was never considered and thus irrelevant to evaluate.</p>
<p>Code Quality</p>	<p>Succeed</p>	<p>In general, the members of the team followed the etiquette outlined in the Team Plan – the code base is generally clean and in sync as a whole. Some instances of code duplication were noted, but most were required for functions to work as intended.</p>

Evaluation of Sprint Planning

In this section, we take a closer look at the planning of each Sprint – mainly at the GitLab issue board. We check whether the issues were completed, were weighted correctly, and were divided reasonably between team members.

Sprint 0



The above screenshot shows a visual example of the issue board of Sprint 0. During this Sprint, tasks that needed to be done were mostly setup tasks – initial designs, scrum log, requirements, etc.

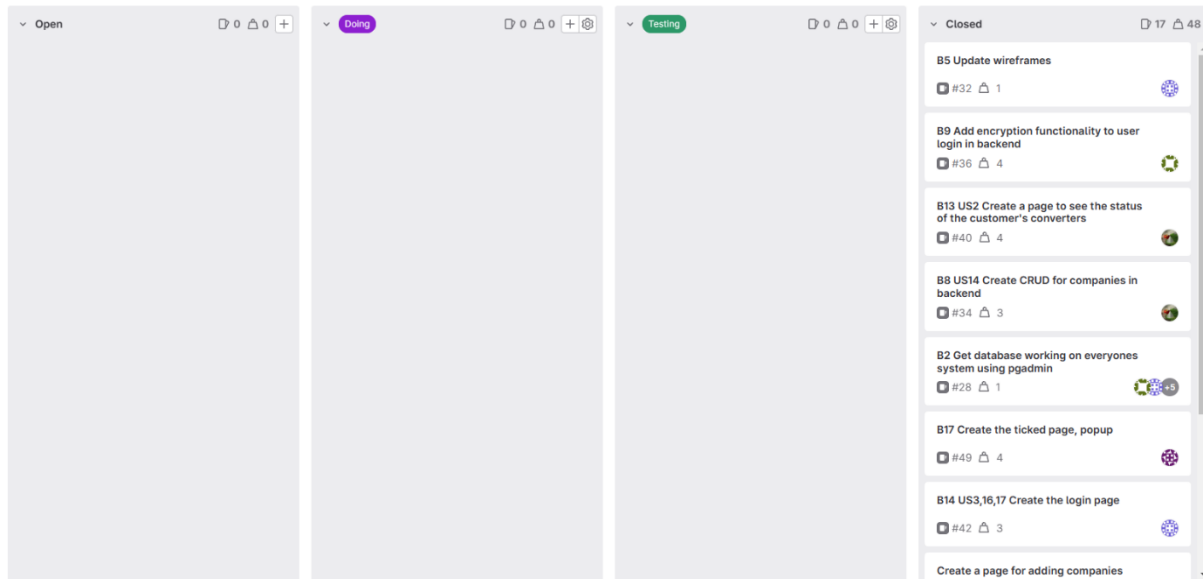
Completeness: Most of the tasks were done and submitted in time, except for the working environment that took until the beginning of Sprint 1 to be finished, due to technical issues.

Weighting: Weight of the tasks were not assigned during this Sprint. The main reason is due to the 'setup' nature of the Sprint 0 and the tasks were also relatively small in size. This did not have any major negative impact on the execution of the tasks.

Distribution: Since the tasks were not weighted, we assume all tasks are of roughly the same weight (though it might not to be the case, however it does not make sense, and also not important, to retrospectively assign any weight for these tasks).

Member	Nr. of tasks worked on	Total weight
Adrian	6	3.33
Tiffany	3	3
George	4	1.83
Darius	3	1.83
Hai Ha	2	1.5
Klaus	1	.5

Sprint 1



The above screenshot shows a visual example of the issue board of Sprint 1. In this Sprint, the main goal was to deliver a core working product that can be used as a demo to the product owner. Core functionalities that were created during the Sprint were basic frontend UI layout and basic CRUD functionalities of the backend.

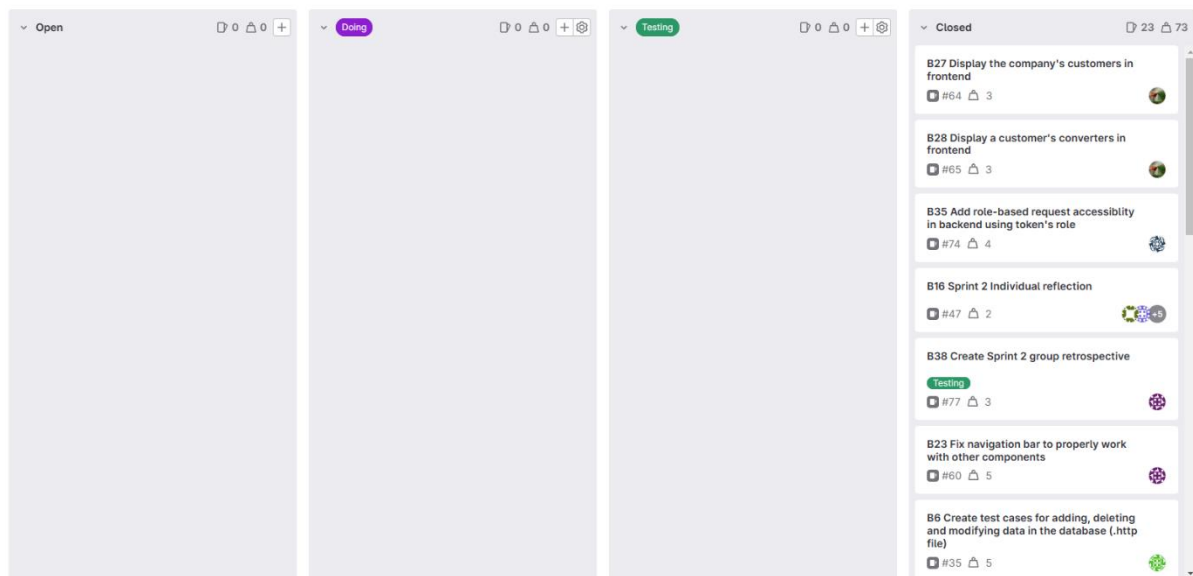
Completeness: As can be seen from the demonstration screenshot, all backlog items of this Sprint were finished successfully and closed.

Weighting: Weight of the tasks were assigned starting from this Sprint. The team sat together at the Sprint planning during lab session on the first Monday of the Sprint and decided the weight of the tasks using known scrum techniques. There were no major complaints regarding the size of the tasks.

Distribution:

Member	Nr. of tasks worked on	Total weight
Adrian	5	12
Tiffany	3	9
George	3	8
Sadra	3	8
Klaus	3	7
Darius	4	6
Hai Ha	2	4

Sprint 2



The above screenshot shows a visual example of the issue board of Sprint 2. In this Sprint, the main goal was to improve the groundwork that was set in Sprint 0 and Sprint 1: adding a fetch call to the front and fixing minor bugs so that the application can be presented to the Client as a demo, adding Bearer token authorization, adding role-based request accessibility to the backend and most importantly implementing the CRON job.

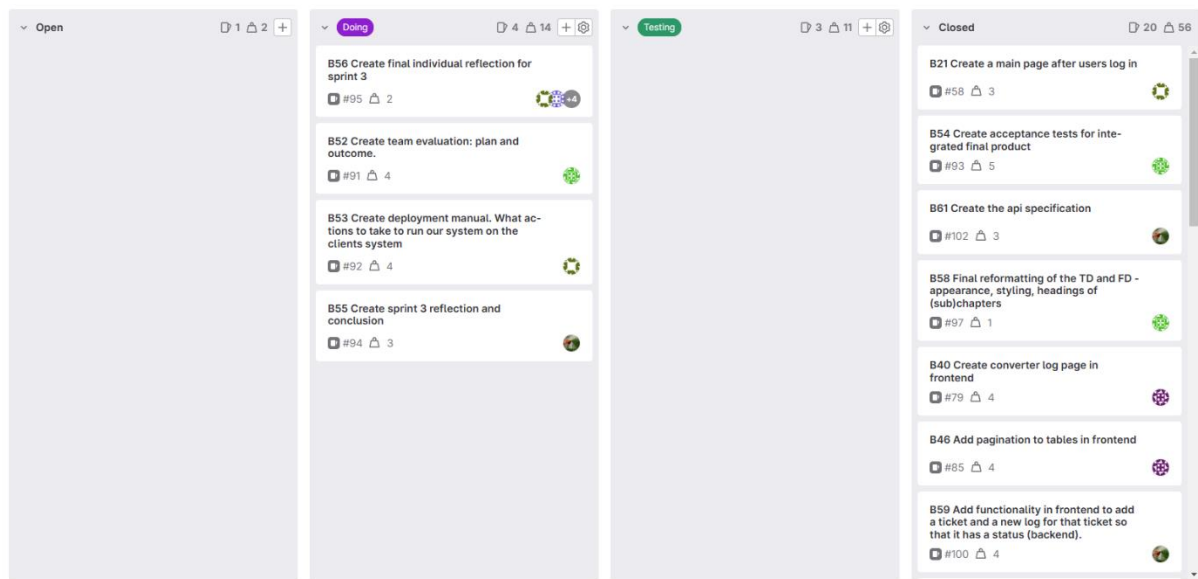
Completeness: As can be seen from the demonstration screenshot, all backlog items of this Sprint were finished successfully and closed.

Weighting: Weight of the tasks were assigned in this Sprint. The team sat together at the Sprint planning during lab session on the first Monday of the Sprint and decided the weight of the tasks using known scrum techniques. There were no major complaints regarding the size of the tasks.

Distribution:

Member	Nr. of tasks worked on	Total weight
Adrian	8	23
George	5	15
Darius	5	14
Hai Ha	3	11
Tiffany	3	10
Sadra	5	8
Klaus	2	4

Sprint 3



The above screenshot shows a visual example of the issue board of Sprint 3. In this Sprint, the main goal was to add a dummy API and set up a CRON job to fetch converter data. The CRON job fetch was considered a critical requirement for the project, so ensuring its proper functioning was of high priority. The sprint also aimed to improve the overall data restriction and security of the system. This was achieved through a combination of code cleanup and the addition of authorization and data validation in both the front-end and back-end. The codebase was reviewed and cleaned up to improve its readability, maintainability, and remove redundant code, while the addition of authorization and data validation restricted access to sensitive data and improved the overall security of the system.

Completeness: As can be seen from the demonstration screenshot, almost all backlog items of this Sprint were finished successfully and closed (at the time of taking). The remaining tasks were set to be finished and submitted with the final submission package.

Weighting: Weight of the tasks were assigned in this Sprint. The team sat together at the Sprint planning during lab session on the first Monday of the Sprint and decided the weight of the tasks using known scrum techniques. There were no major complaints regarding the size of the tasks.

Distribution:

Member	Nr. of tasks worked on	Total weight
Adrian	7	21.5
George	6	15.5
Hai Ha	5	15
Klaus	7	14
Sadra`	4	12
Tiffany	3	10
Darius	3	7

Imbalances In Distribution of Tasks

As can be noticed in the distribution table, the distributions of tasks were not always balanced. This is due to one of the following reasons:

- The tasks were moved since the original responsible member was not able to sufficiently complete them, either due to a lack of knowledge or other personal circumstances. This is not the fault of the member – the purpose of this project, is also to learn and practice new things in addition to applying the already known knowledge. What the team managed to successfully achieve in this aspect is to make sure no one member is always at the bottom of the distribution list and get a free ride – everyone contributed to the outcome of this project.
- The size of the task(s) was not close enough to the weight measurement. While weighting and sprint planning is a team decision, some tasks can take either more or less effort to be completed. If a task is considerably larger than anticipated, the time allocation of this task would eat up into that of other tasks, leading to the task needing to be moved. In contrast, if a task was too easy, the member who completed it will be idle and be able to help other members by picking up available tasks. The lesson of correctly weighting and distributing tasks can only be learned by experience: the more team members participate in projects, the better idea of the size and required effort for a task will they have, and planning will be more accurate as a result.