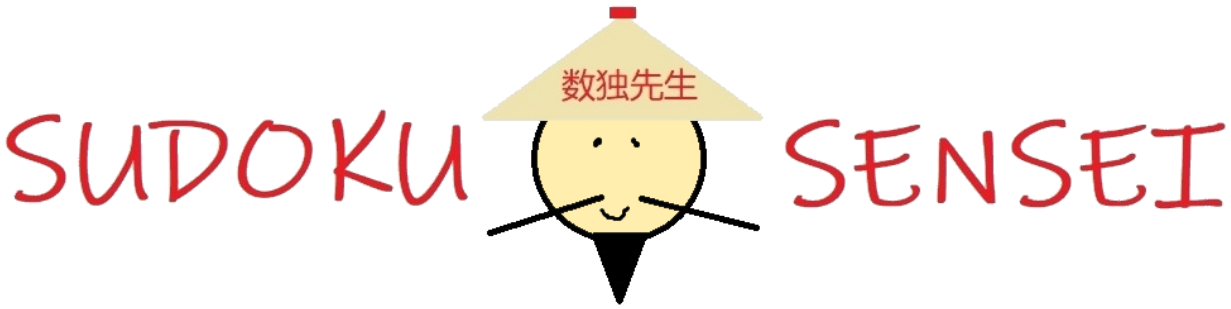
Final Project Report



Team Alpha

Vincent Badenhorst

Homayoun Banazadeh

Howard Chang

Damoon Tahmasbi

Introduction

This is a final report for a three-month-project of our team of four. The goal of the project was to build an app from scratch based on a customer's order, and to improve it per iteration based on customer's feedback. The goal, also, was to experience and practice the process of programming with a team of programmers. The process involved assigning roles to each team member, collaborating and deciding on various aspects of the software design as a team, and programming the app in a modular fashion and putting pieces of the code together afterwards. During the process, we were faced with some challenges involved in software engineering, and working as a team, which we are going to describe a few of them in the report.

Project Description

The main objective of the project was to develop a sudoku app used for language learning purposes. Moreover the customer would request more specific features as the development progresses, such as tablet support and listening comprehension.

Overall, the project was split into four iterations. In iteration 1, after receiving a brief description of the overall functionality of the app. We assigned roles to each group member; A Product Owner, Repo Manager, Scrum Master and an Observer. To be fair to everyone, roles were rotating at each iteration.

The Product Owner would ask questions from the customer, i.e. the course instructor, after every group meeting if any clarification was needed.

The Scrum Master created a workspace, i.e. group chat room, in Slack, which is an online messaging service. This specially was helpful, since all members had a very busy schedule and it was difficult to find meeting times during weekdays.

To start, the Repo Manager created a new project in gitlab and connected everyone to the project. The repository was further connected to the slack, so that any change in the master branch would be reported to everyone.

During the first meeting for each iteration, a general workflow to use gitlab was decided upon. Every change or addition to the code should be first stated as an issue in gitlab, then a branch should be created for that issue, and when the code is final, a merge request should be issued. The Repo manager would, then, review and approve the merge if there was no conflict. No direct commit to the master branch was allowed. If the issue was completely resolved, the branch could be deleted and the issue would be closed.

Software Engineering issues discussion

Software engineering issues can be categorized in coding and teamwork.
In coding, there were a few things that everyone was learning for the first time, such as using git in a team, or using android libraries. In teamwork, since each member's work was dependant on other

members' work, a full understanding of each other's skills, abilities, and personalities was required. The followings are a few of them.

Learning Git severely delayed our progress during the first few weeks due to lack of a comprehensive understanding of Git's structure. Even in later iterations, when we were familiar with the Git process, we have lost some of our useful codes as we were getting close to the deadline. For example, when there were too many people working in the same area of the code, something that was working initially would break after new merges. This usually happened when one of the commits was too large or being pushed after a long time. To solve the issue, we started creating issues for every minor change and more frequently, then, we tried to close the new branches as soon as possible if the issue was resolved.

Of course, one of the main challenges was understanding the customer's request or feedback. But even after that, there were miscommunication within the team members when we were designing the code. For example, one person was responsible to write code for the Sudoku class and its utility functions, and another person was implementing the class in the game. The utility function isAllowed() would check if the requested entry is allowed before storing the input, but the implementation was using the function after the input was stored. Miscommunications like this, would require spending a lot of time in reading and understanding each other codes for debugging, when the whole issue could have been avoided if there was a proper and comprehensive communication in the beginning.

Another challenge was debugging someone else's code. It turns out that the most efficient way in debugging a code is to have each person responsible for his/her own code. Since this was not always practical, we would try to leave high level comments in the code as much as possible. This would make the code more readable. If comments were not enough, the person in charge of the code would spend some time on a Skype call or Slack to explain the logic of his code to the other member who was assigned the task of debugging.

Concluding remarks and future

Currently, there are still elements in our app that can be improved upon. In addition, instead of adding new features, improving/fixing these specific elements will be prioritized should development continue:

- One major thing we would improve upon is the overall appearance and aesthetic for our sudoku app, as it clearly have rooms for improvement, especially the game board itself, as the page's color, as well as theme choices clearly contradicts the appearances of other pages such as the menu page.
- Moreover, there are still several issues that need to be fix in consideration of some key features. For instance, the scrolling for the sudoku page, there exists a glitch where scrolling will malfunction if pitching from the cells themselves.
- Unfortunately, due to time restrictions, some planned features were never implemented, including: the lack of support for other languages when in listening comprehension mode; the lack of a difficulty setting; as well as the lack of ability to create a game board dynamically.

Again, in the events of development continuing, the following issues needs to be resolved.

For the appearance and aesthetic improvements, maybe the main reason was gender imbalance of our group. We, surely, would have had a better design and choice of color, had there been a female member in our team. We will be seeking professional design help from our female colleagues to resolve the issue in later iterations.

Some of the other issues were related to our choice of libraries. We are going to research and try out different open source libraries, that could be implemented more efficiently or have better performance.

In addition, after spending hours in debugging, we realized that many of the code needs to be completely thrown away and programmed again from scratch. We need to rewrite the code, specially, for those parts of the code that are so tangled that we could not find any way to apply abstraction and proper inheritance to them. On a related note, we had difficulty understanding each other's code, so we will need to improve documenting and commenting on our code to ensure that the code is understandable to all members or even outsiders who may want to use or extend our code.

Finally, despite the development facing some issues, throughout our collaboration we learned that being able to code as an individual is not enough by itself. However, having the team spirit and putting the software engineering practices into reality are, in general, more important for the team's success.