# CE291 Team Final Product Report

Team Number: 5

Module Code: CE291

CSEEGit URL: [*https://github.com/benleong94/pmchallenge*](https://github.com/benleong94/pmchallenge)

CSEEJira Project URL: [*https://bendemeerben.atlassian.net/jira/software/projects/SS/boards/4*](https://bendemeerben.atlassian.net/jira/software/projects/SS/boards/4)

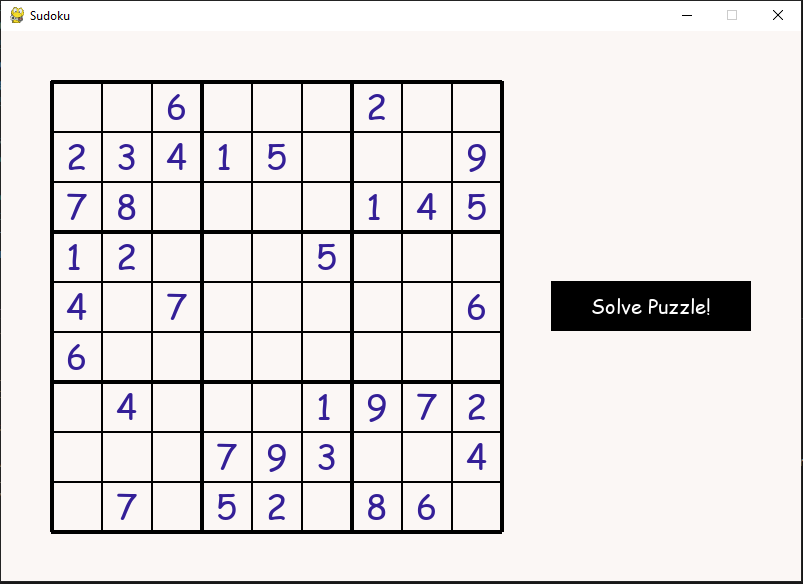
Team Members:

* *Chio Yong Kiong*
* *Kong Chee Wee*
* *Benjamin Leong Jia Juin*
* *Nelson Tan Chun Soon*

# Final Product Demonstration

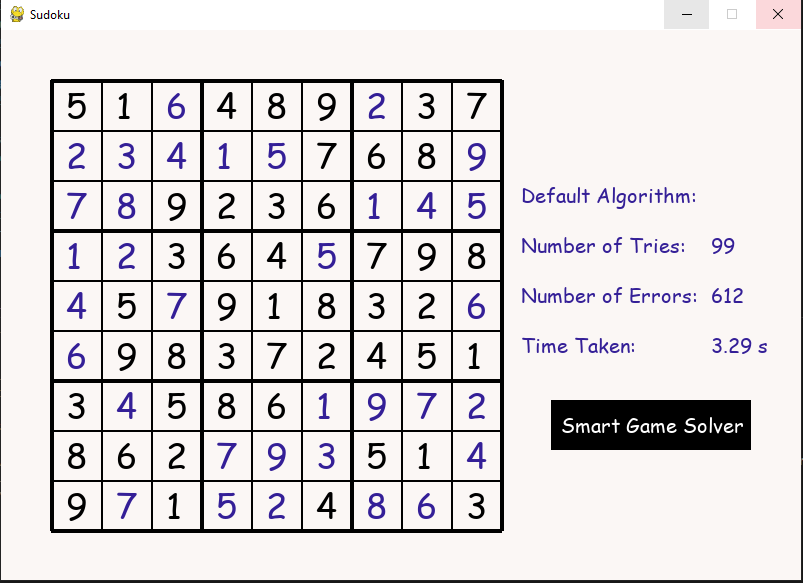
**Product Demonstration Report**

**Picture 1:**

[](https://user-images.githubusercontent.com/56427412/181763865-b6aef0c2-9c93-44d9-aeee-fbe6d1a8deb0.png)

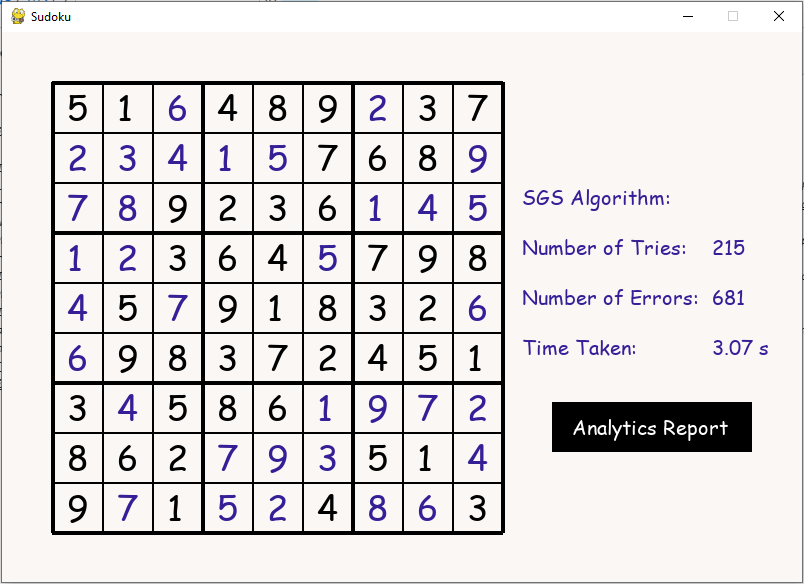
* On starting the program, the Game Level Generator is launched and it will generate a grid, which is the data file. The user can proceed to click "solve puzzle" to execute the algorithm to solve the puzzle.

**Picture 2:**

[](https://user-images.githubusercontent.com/56427412/181764099-4db548cd-29b7-4242-9759-9b2ac1fa0086.png)

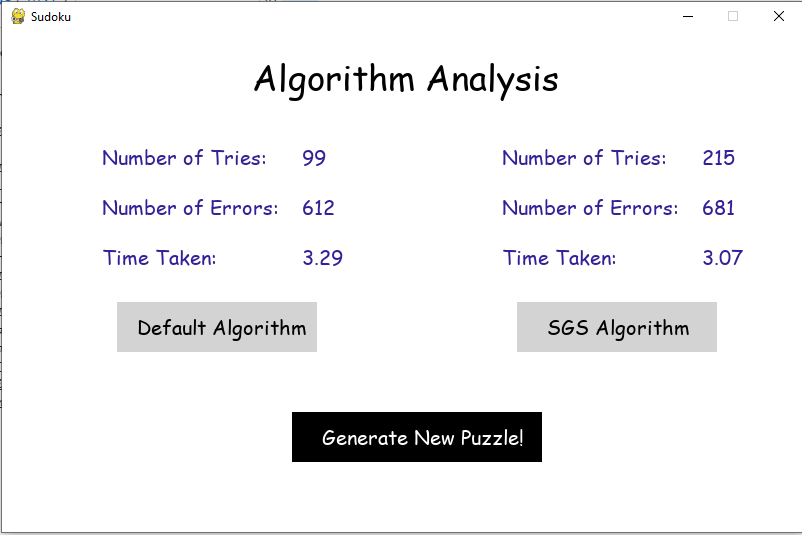
* The default algorithm, which is the recommended solution, will proceed to solve the sudoku puzzle. The algorithm is analyzed using three indicators - number of tries, errors and time taken are shown. The user can proceed to click on "Smart Game Solver" to launch the Smart Game Solver (SGS), which will provide a more efficient algorithm to solve the puzzle.

**Picture 3:**

[](https://user-images.githubusercontent.com/56427412/181764166-a848478a-3ada-4633-93a0-9cede7e4f5b2.png)

* The SGS reads the data file (Sudoku grid) and solves it with a more efficient method. In the same way, it analyzes the algorithm with the same indicators as shown. The user can proceed to click on "Analytics Report".

**Picture 4:**

[](https://user-images.githubusercontent.com/56427412/181764208-8fe8af75-b8f1-46dc-ae35-bad106e130ef.png)

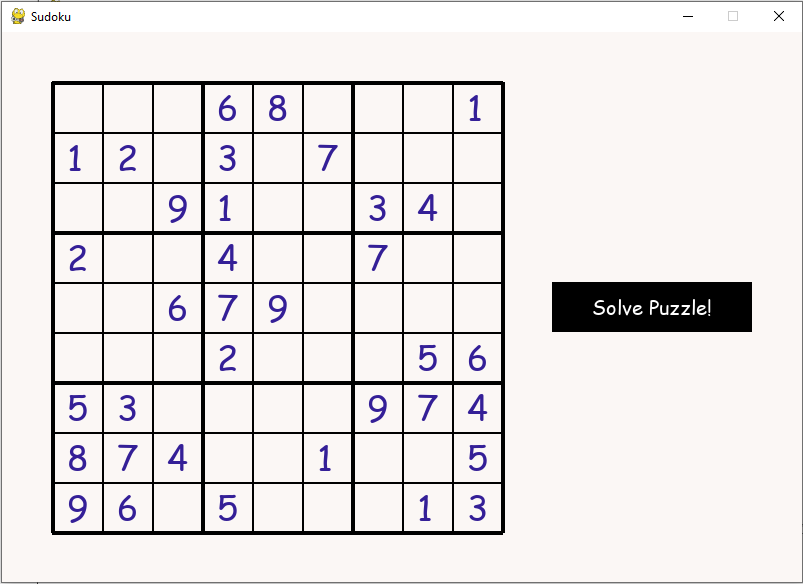
* Once clicked, the SGS generates analytics report that compares the two algorithms that were used to solve the puzzle. The user can proceed to generate a new grid.

**Picture 5:**

[](https://user-images.githubusercontent.com/56427412/181764281-bfdf6211-1607-45cc-ac51-24f9082e5fc7.png)

* There is an option to generate an "easier" or "harder" grid. When either one is selected, the SGS will use the indicators to generate a new puzzle. It will do that by taking the indicators of faster algorithm, run it through a calculation process that will produce a certain values. These values will be used to select an "easier" or "harder" grid from a list of grids.

**Picture 6:**

[](https://user-images.githubusercontent.com/56427412/181764523-b215ab26-331a-41ec-bcaa-257d4e18cf11.png)

* The values are delivered back to the GLG, which will generate a new puzzle based on the user's selection.

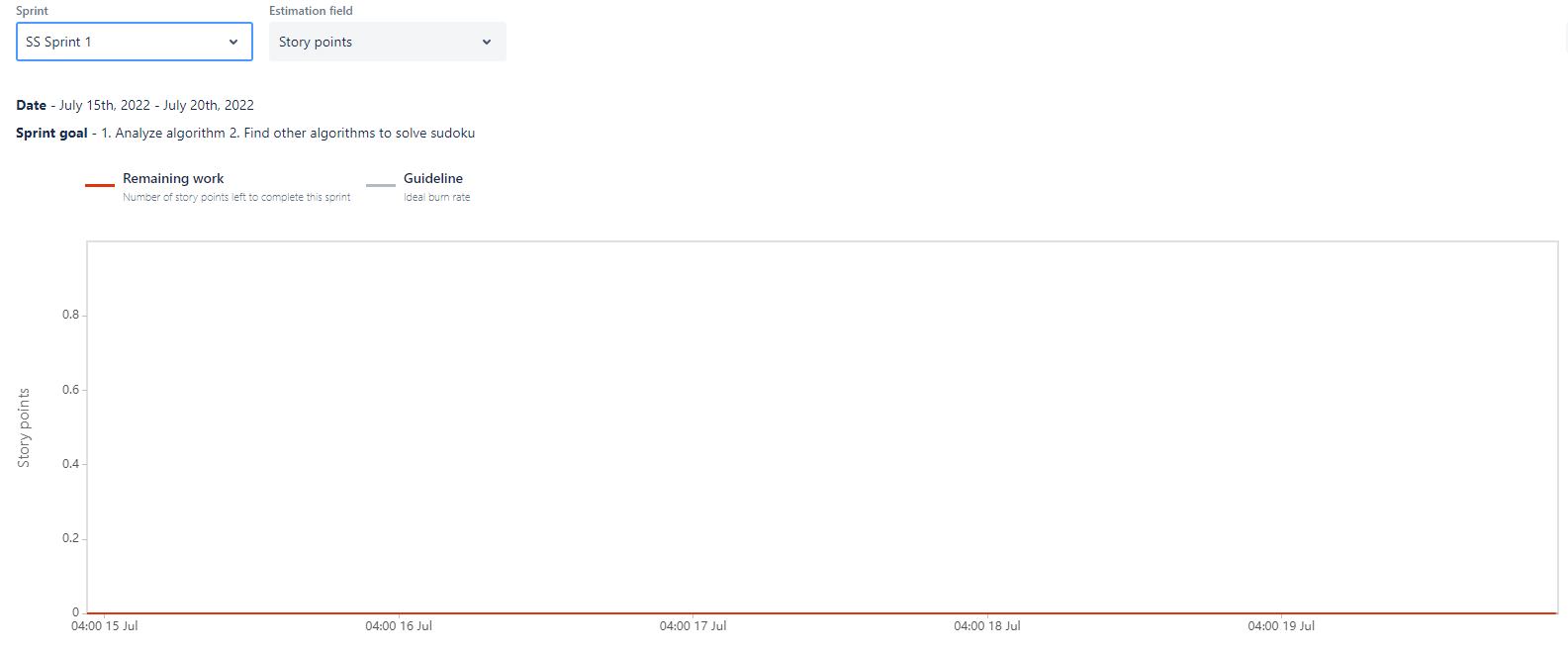
1. Project Management Log

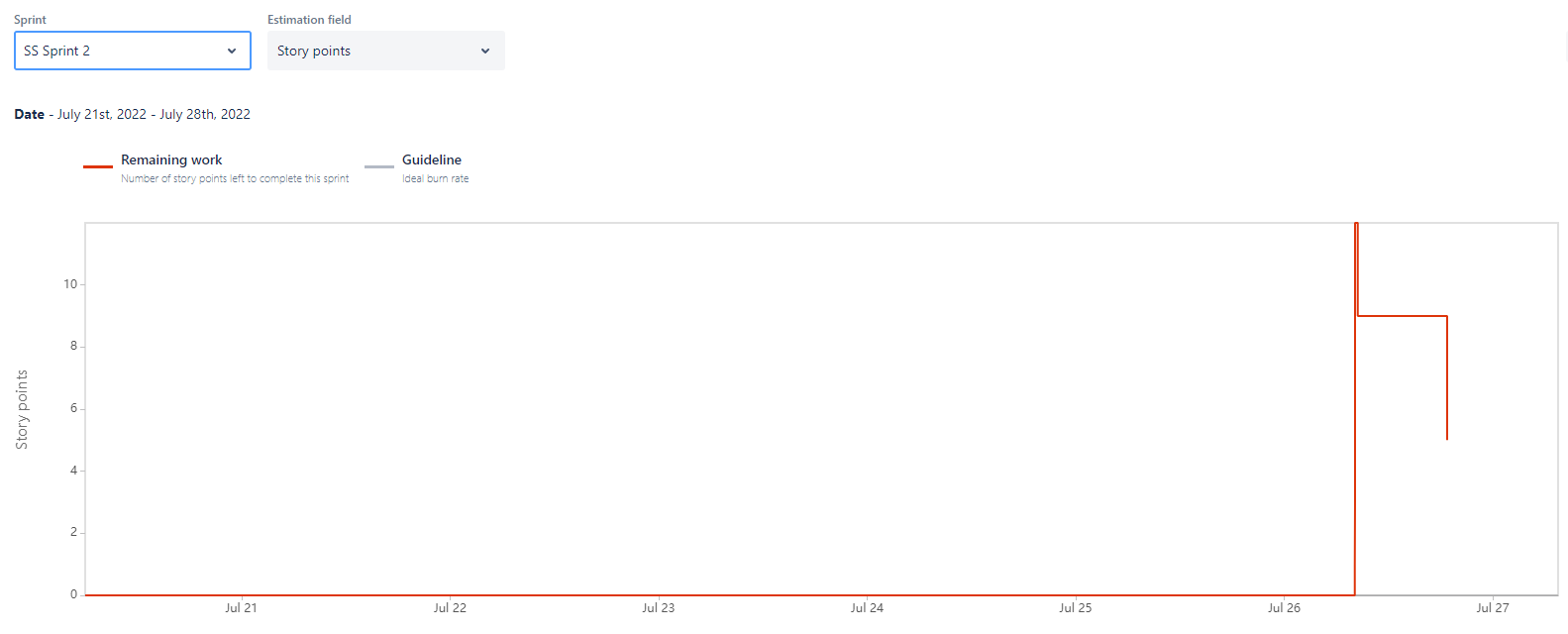
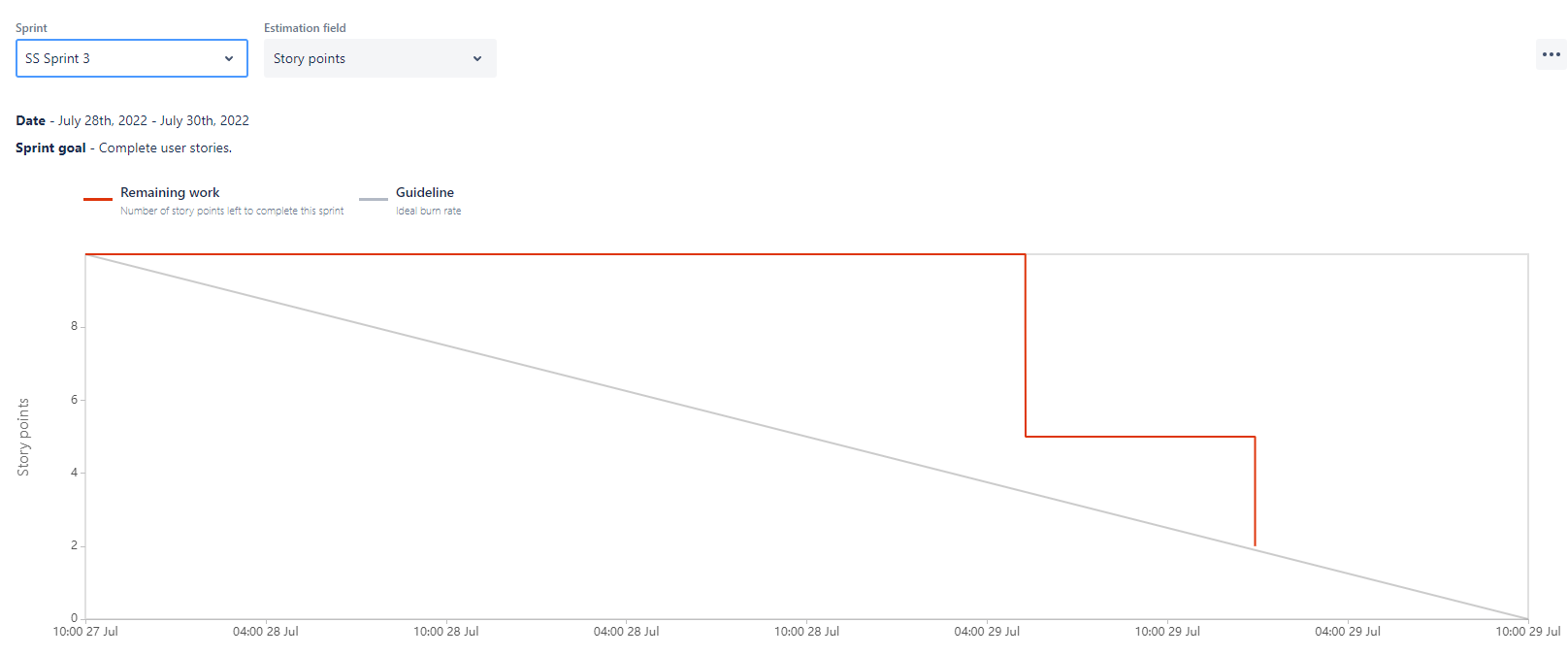
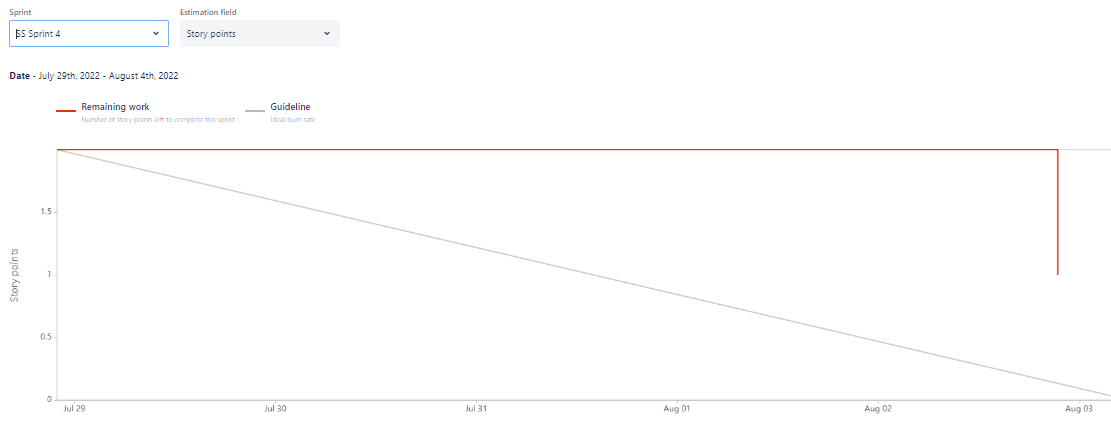
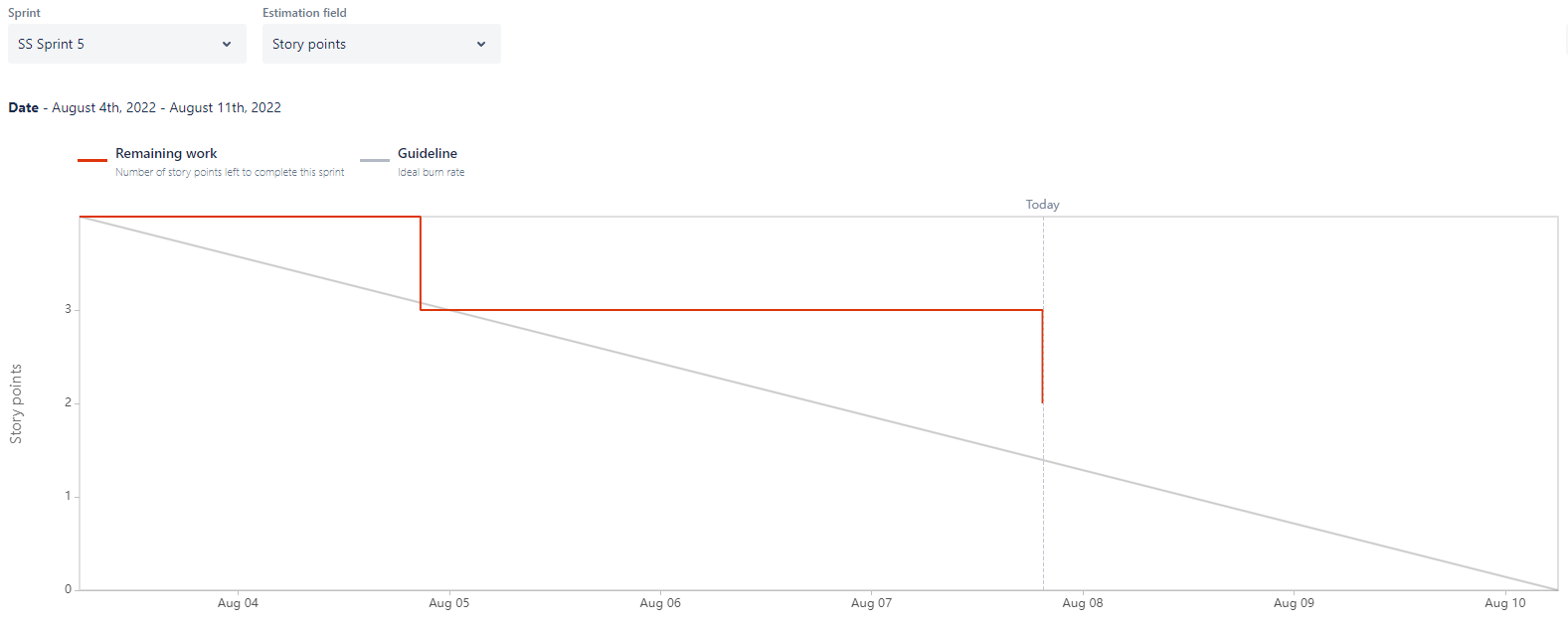
**Project management discussion**

* Generally, we managed to assimilate well and work together in a team effectively. As most of us are new to Agile methodology, we needed some time before becoming more familiar with how to run the scrum meetings.
* During the first two scrum meetings, we were unsure with the differences between the scrum review and sprint retrospective. As such, we focused most of our attention on the product, and less on how well we were working together. However, as the project continued, we realized the importance of the scrum retrospective, as it helped our group communicate and coordinate better.
* As we proceeded through the various sprints, there were issues that surfaced which required resolution.
* During the first sprint, we discovered that not everyone was 100% clear about the requirements of the product after the sprint planning. Hence, some of the user stories not being completed correctly and we had to bring forward most of the user stories to the following sprint. This slowed our velocity significantly. As such, for the second sprint planning, we made sure that everyone was clear on the product requirements before starting the sprint.
* As we started the programming aspect of our product, we learnt that there were differences in the levels of programming skill among our group members. This affected the velocity of the project as some tasks were outside the abilities of some group members. We resolved this by changing the way we allocated the work. For example, tasks that required more programming were taken by members who are better at programming, while the others focused on other tasks like report writing. Also, we started working in pairs and we realized that such an arrangement helps us to complete the tasks better.
* As we are all full-time working adults with other personal and family commitments, we had to learn how to manage our time effectively and coordinate with each other in order to get work done. We had some physical meetings as well as online meetings to discuss the project. For better communication, we used mostly Whatsapp and MS Teams to coordinate our work.

**Sprint Burndown Charts**

* Sprint 1
* Sprint 1 showing no user stories at all, because we are trying to familiarise with the Jira and Github.

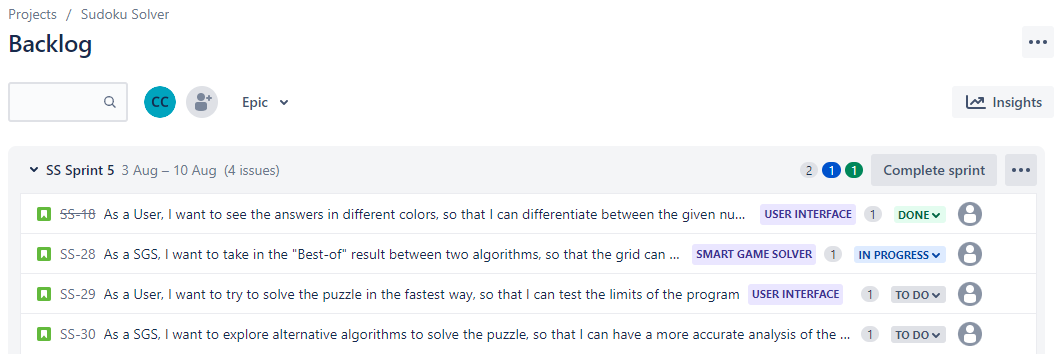
[](https://user-images.githubusercontent.com/56427412/181765259-b5d6426b-f6ef-40b6-94af-10803d052941.png)

* Sprint 2
* Sprint 2 shows we putting some user stories on 26 July and managed to done some work. [](https://user-images.githubusercontent.com/56427412/181765357-82fbbb16-6377-41b1-b0ed-2681c0ee44a8.png)
* Sprint 3
* Sprint 3 shows we started with 10 story points and about 2 days later, we managed to complete some of the user stories which reduce to 5 story points, at the end of the day, we managed to further reduce the story points to 2. [](https://user-images.githubusercontent.com/56427412/181765435-af2940fd-fd4f-4cc2-8d37-47d4dd16a683.png)
* Sprint 4
* Sprint 4 shows we started with 2 story points and about 5 days later, we managed to complete one story point, and concentrate most of the time to do MVP. [](https://user-images.githubusercontent.com/56427412/183281039-765ef003-a4a4-46b3-b0cf-50b0b141d9f1.png)
* Sprint 5
* Sprint 5 started with 4 story points, and done one of the story point on 5 Aug and another one on 7 Aug. [](https://user-images.githubusercontent.com/56427412/183407949-3f771dd5-e53d-4fdb-a1a2-af16f1893d57.png)

**Burndown-Charts Discussion**

* The burn-down charts show how many tasks are completed in the sprints and how much work remains. We can improve our velocity in future sprints by clarifying tasks and providing a reasonable time estimate.

**Product Backlog**

[](https://user-images.githubusercontent.com/56427412/183280885-8095511f-6147-40c3-a2ce-ebbf559e9b67.png)

1. Team Effort Log

**Kong Chee Wee**

**Sprint 1 (week 2) :**

* Try to understand the goals of the project and checking on youtube's tutorial video to familize with Jira and Github.
* Discuss among team members and decide which game product we choose for the project.

**Sprint 2 (week 3) :**

* Assigned by scrum master to do coding on the Smart Game Solver, partner with Yong Kiong.
* Research on internet for alternative yet faster algorithms to solve the Sudoku grid.

**Sprint 3 (week 4):**

* Testing on solving Sudoku level using the optimized backtracking algorithm. The algorithm is efficient compared to the traditional backtracking method, and we decided to use the algorithm as our Smart Game Solver's algorithm.

**Sprint 4 (week 4) :**

* Assigned by scrum master to do coding on the user interface by inserting "Solve Puzzle!" button on the game's front page.
* Volunteered to help prepare MVP documents, primarily on Team Effort Log.

**Sprint 5 (week 5) :**

* Discussion among team members on improving the product after gaining feedback from the client.
* Help to prepare final product documentation and presentation slide.

**Chio Yong Kong**

**Sprint 1 (week 2) :**

* Confused with the project's requirement, I firstly thought of creating a game only, corrected by team members after discussion. Familiarizing with Jira only, as I already know how to use Github.
* Discuss among team members and decide which game product we choose for the project.

**Sprint 2 (week 3) :**

* Assigned by scrum master to do coding on the Smart Game Solver, partner with Chee Wee.
* Research on internet for alternative yet faster algorithms to solve the Sudoku grid.

**Sprint 3 (week 4):**

* Testing on solving Sudoku level using the simple algorithm, the algorithm is not fast enough compared to the traditional backtracking method. We keep it for alternative's algorithm.

**Sprint 4 (week 4) :**

* Assigned by scrum master to do coding on the user interface by inserting "Easier" or "Harder" button after the game analytic report.
* Volunteered to help prepare MVP documents, primarily on Requirements And Risk Log.

**Sprint 5 (week 5) :**

* Discussion among team members on improving the product after gaining feedback from the client.
* Help to prepare final product documentation and presentation slide.

**Benjamin Leong Jia Juin**

**Sprint 1 (week 2) :**

* Try to visualize the final output of the project and testing the features of Jira and Github.
* Discuss among team members and decide which game product we choose for the project.
* Create Jira and Github repository and invite all team members to join in.

**Sprint 2 (week 3) :**

* Assigned by myself to do coding on Game Level Generator, partner with Nelson.
* Coding on solving the Sudoku Grid using traditional backtracking method.

**Sprint 3 (week 4):**

* Done coding for the screen display of the Sudoku grid and implemented the backtracking solving algorithm.
* Testing and debugging session for the GLG.

**Sprint 4 (week 4) :**

* Assigned by scrum master to do coding on game difficulty selection's algorithm.
* Volunteered to help prepare MVP documents, primarily on Product Demonstration.

**Sprint 5 (week 5) :**

* Discussion among team members on improving the product after gaining feedback from the client.
* Help to prepare final product documentation and presentation slide.

**Nelson Tan Chun Soon**

**Sprint 1 (week 2) :**

* Try to understand the requirement of the project and try to familize with Jira and Github.
* Discuss among team members and decide which game product we choose for the project.

**Sprint 2 (week 3) :**

* Assigned by scrum master to do coding on Game Level Generator, partner with Benjamin.
* Research on internet for coding to solve the Sudoku Grid using traditional backtracking method.

**Sprint 3 (week 4):**

* Done coding for the screen display of the Sudoku grid and implemented the backtracking solving algorithm.
* Testing and debugging session for the GLG.

**Sprint 4 (week 4) :**

* Assigned myself to do coding on game analytic report.
* Volunteered to help prepare MVP documents, primarily on Project Management.

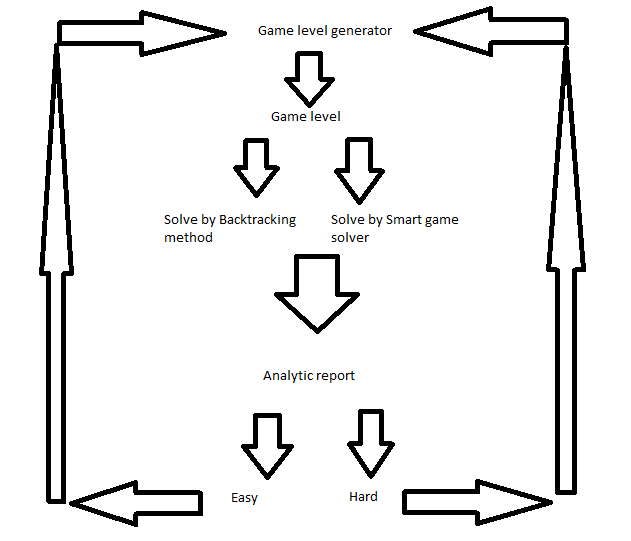
**Sprint 5 (week 5) :**

* Discussion among team members on improving the product after gaining feedback from the client.
* Help to prepare final product documentation and presentation slide.

1. Product Implementation Report

**Team Implementation Report**

**Technical Diagrams**

[](https://user-images.githubusercontent.com/56427412/183280065-c3e083e0-a3c4-4a2d-ac9f-163271ed764d.png)

**Technical Description**

* There are two main python files that are required for the program to be executed properly. This first python file is the Sudoku-Solver program and the second file contains a list of Sudoku grids.
* The Sudoku-Solver program is organized in a functional programming structure. It is mainly organized into seven sections:

1. Grid generation – This section contains only one function. This function will print a screen display of the grid, the given numbers and the button to proceed to solve the grid.
2. Execution of first algorithm and analysis – This section contains three functions. The first function tests the validity of the number in the cell. The second function allows the algorithm to traverse through the grid. The third function prints the solving process onto the screen, along with the indicators for algorithm analysis.
3. Execution of second algorithm and analysis – This section contains seven functions. The first function finds an empty cell in the grid. The next four functions test the validity of the number in the cell. The sixth function allows the algorithm to traverse through the grid. The final function prints the process onto the screen, along with the indicators for algorithm analysis.
4. Display of analytics report of both algorithms – This section contains one function. This function will gather the data from both algorithms and display it on the screen.
5. Selection of next difficulty level– This section contains one function. This functions will allow the user to choose “easier” or “difficult” for the generation of the new grid.
6. Generation of new grid based on difficulty selection – This section contains two functions. The first function will process the indicators by selecting the faster algorithm (depending on the indicators) and produce certain values that will ensure that the next grid is easier or harder, depending on what the user chooses. The second function will accept these values, and choose a grid from the list of grids given.
7. Main function –This main function will loop through the above six sections repeatedly.

* The second python file contains a list Sudoku grids that are stored in a nested list format. The Sudoku grids have been arranged based on their levels of difficulty. Each Sudoku grid has a specific index, which can be called in the main function.

**Algorithms and Data Structures**

* The data structure for the Sudoku grid is a nested list. There are nine lists (of nine numbers) nested within a single list. The single list will serve as the Sudoku grid, which will be called by the Sudoku solver.
* The algorithms used to solve the Sudoku puzzles are backtracking algorithms. The second algorithm (used by the Smart Game Solver) is a more optimized version than the first. The backtracking algorithm traverses across the Sudoku grid, from left to right, and from up to down. The algorithm will visit every empty cell sequentially, starting from the cell in the first row and first column. At every empty cell, the numbers 1 to 9 will be incrementally placed into the cell, and the number will be tested for validity. If the number is valid, the number is fixed into the cell, and thee algorithm will move to the next cell. If the numbers 1 to 9 all fail to be valid, a number 0 is placed into the cell, and the algorithm will return to the previous cell and the number is increased by 1. This process will carry on until all the cells are filled.

**Imported Libraries**

* The libraries that were imported were pygame, time, json and random. Pygame is the main library used to construct the Sudoku game. The time library is used to calculated the time taken for the algorithm to solve the grid. The json library was used to create a copy of the grid. The random library is used to generate a random number.

**Known Issues**

* There was an issue with removing the “answers” from the grid when program moved the second algorithm. As a result, the second algorithm always received an already solved grid, and it took 0 seconds to solve the Sudoku grid. This was due to an issue with the memory address of the Sudoku grid. It was corrected by creating a copy of a grid in the computer’s memory address.
* When the backtracking algorithm was being executed to solve the puzzle, sometimes the numbers were not printed out properly to the screen. This resulted in the Sudoku puzzle looking incomplete, though the puzzle had been solved in the computer memory. This was due to the speed at which the numbers were printed. It was corrected by inserting a 2.5ms delay between each number print.
* For both the algorithms, the calculated “errors” were always higher than the “tries”. However, logically the “tries” ought to be more than the “errors”. This bug has not been rectified.

1. Product Testing Report

**Test report**

**Testing performed**

* This testing includes three important sections.

1. The Game Level Generator and Smart Game Solver should be able to solve a given grid with their own respective algorithms.
2. The algorithm should be able to be analyzed with certain measurements and indicators.
3. An easier (or harder) grid should be able to be generated with the input of the indicators from the previous grid.

* As of the end of sprint four, the product passes testing for all three sections.

**Automatic Testing:**

* Nil.

1. Product Context Report

# Product Context

## Legal

We would release our product under the GNU GPL (General Public License) - This Licence allows the 4 freedoms (elaborated under Ethical), and stipulates that any derivative of this software must retain these same rights.

GNU General Public License is intended to guarantee your freedom to share and change all versions of a program - to make sure it remains free software for all its users. The Free Software Foundation (FSF) uses the GNU General Public License for most of their software; it applies also to any other work released this way by its authors. We can apply it to our program too.

When we speak of free software, we are referring to freedom, not price. The General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they too receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, FSF had designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, FSF stands ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

## Ethical

We should refer to the Free Software Foundation (FSF) views of Ethical Computing.

A) For our Sudoku software to be ethical, it should not control the consumers in any way. “Free Software” is defined to be software that grants the user 4 fundamental freedoms:

1. The freedom to run the program as they wish, for any purpose
2. The freedom to study how the program works, and change it so it does their computing as they wish.
3. The freedom to redistribute copies so that they can help others.
4. The freedom to distribute copies of their modified versions to others.

B) Data Protection Act (DPA) 1998 - Applies to electronic and manually stored personal data (data allowing a person to be identifiable). DPA gives living and identifiable individuals the right to know what information is on record and to challenge it if appropriate. Each organisation is a Data Controller which is responsible for compliance. The people whose data is stored and used are called “data subjects”

Everyone responsible for using data has to follow strict rules called ‘data protection principles’. There are generally Eight Principles to follow - Personal data must be: 1 . Obtained and processed ‘fairly and lawfully’ - Must be with consent of the data subject and they now have to be notified to accept cookies or not. 2. Used for limited, specifically stated purposes. Data controllers must notify the Information Commissioner (a government body) of the personal data they are collecting and the purposes for which it is being collected. 3. Used in a way that is adequate, relevant and not excessive - E.g. do not ask for customers' address or martial status etc, if that’s not explicitly needed 4. Accurate - Kept up to date, but this can be impractical 5. Kept for no longer than is absolutely necessary 6. Handled according to people’s data protection rights 7. Kept safe and secure 8. Not transferred outside the UK without adequate protection

## Health and Safety

The Health and Safety (Display Screen Equipment) Regulations 1992 place specific requirements on employers with the aim of protecting workers from the health risks associated with DSE. These duties also apply to the self-employed.

These provisions would mostly apply to our staff, particularly our Sudoku developers. We emphasize that these provisons do not extend to our consumers. Consumers would have to exercise their own judgement and self-responsibility to prevent excessive use of our product, which may result in detrimental health problems.

The health problems associated with display screen equipment (DSE) work are upper limb disorders (including pains in the neck, arms, elbows, wrists, and fingers); temporary eyestrain (but not eye damage) and headaches; and fatigue and stress.

Only a small proportion of people who use DSE actually suffer ill health as a result. Usually these disorders do not last, but in a few cases they may become persistent and even disabling. There may be a sudden onset of symptoms in response to short-term activities, or development may be gradual and caused by longer-term poor provision or practice. Timely reporting, investigation and action to remove the cause of problems is vital to prevent worsening of symptoms and to promote recovery. The causes may not always be obvious and can be a combination of factors, but enough is known about the importance of some measures - for example, the need to sit properly and to have frequent breaks from the screen - to allow the risks to be tackled effectively.

Display Screen Equipment means any alphanumeric or graphic display screen, regardless of the display process involved. It covers PCs, laptops, tablets and smartphones as well as other methods of displaying data, such as CCTV screens.

A 'user' or 'operator' is a worker or self-employed person who uses display screen equipment as a significant part of their normal work. The Health and Safety Executive (HSE) specify that it applies to 'workers who use DSE daily, for an hour or more at a time not infrequent users or short-term use'.

Eyesight tests and corrective eyewear must be provided, by the employer, free of charge if required. A test should be carried out by an optician if a DSE user requests it. Employers can offer vision screening tests, but they cannot prevent a user opting for a full eyesight test instead. Where the test shows the need for corrective eyewear specifically for DSE use and to comply with the regulations, the employer must fund the basic cost.

Training and information must be provided to users and operators, explaining the risks of DSE use and how to arrange the workstation safely. It should also cover what to do if the user develops any work-related health problems.

Regular breaks aim to avoid risks of postural fatigue by introducing periodic short breaks or changes of activity away from the DSE.

1. Product Marketing Plan

**Marketing Plan**

**Customers**

* Our product's target audiences are those leisure players who are ages 13 and above, both male and female, regardless of ethnicity or religion, and anyone who is IT literate and has an interest in solving Sudoku puzzles. The group would mainly consist of students or retired elderly who have more time and would like to keep their minds sharp by solving puzzles. Income level would not be applicable since this is a readily available free product. We aim to capture the Singapore market, or any internet accessible countries.

**Economics**

* Its quite hard to estimate the total size and percent share of the market of Sudoku games in Singapore, as the game is relatively new compared to other puzzles such as Crossword and Word search puzzles. The fact that the game doesn't require any mathematical and wording ability makes it highly popular among leisure players, which raises the demand for its supply of more complex levels as the user can finish each puzzle in a short time. There is also demand from hardcore players who want to explore new or alternative solving techniques besides the old backtracking method.

**Competition**

* The other products on the market are Sudoku solvers. They simply accept a Sudoku grid, solve it with an algorithm and display the answer. Our product differs from them in a few ways. Firstly, our program can analyze algorithms by calculating the number of tries, errors and time taken to solve the Sudoku grid. Secondly, it can provides a report that compares between different algorithms. Thirdly, it can generate a new Sudoku level based on given parameters, and allows the user to choose an easier or harder Sudoku grid.

**Product, Price, Promotion, Place**

* For Product, Smart game solver and Game Level Generator are the main features included in our product. The SGS can generate a report to a player that compares two different algorithms by analyzing the number of attempts, errors, and time required to solve the Sudoku grid. At the same time, the GLG can create a new Sudoku level for the player after selecting either an easy or more difficult sudoku grid based on the analytic report. These two features will assist the player in gaining confidence in solving the game and thus increase our product demand.
* For Price, our product will be accessible in the app store or play store with an in-game advertisement, and As a method of revenue generation, in-game advertising involves the sale of ad space in our product.
* For Promotion, we would be adopting the online and word-of-mouth advertising method as it helps us reach a wider audience and tell the market about our product. Our vast number of levels generated using the game level generator and effective game-solving method would attract Sudoku players to try on our product, leading to discussion among them.
* For Place, the distribution of our product is mainly done online by application listing in Apple Store and Google's Playstore.

**Sales Forecast**

* We estimate a minimum of 500k downloads in the first 12 months upon release into the Apple store or Google's Playstore. A substantial revenue will be gained from the advertisement in-game after deducting the program license or application listing cost, promoting costs, and game/server maintenance cost.