# Description of what we did in project: 3-5 pages

* 1. Team Forming

Our team consists of four members. Each member self-evaluated his/her programming language competency as follow:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **C/C++ - CGI** | **Java - JSP** | **C# - .NET** | **PHP** | **Python** | **Ruby** |
| **Dung Dam** | 1 | 2 | 2 | 5 | 4 | 3 |
| **Anh Nguyen** | 3 | 1 | 0 | 0 | 3 | 0 |
| **Duc Nguyen** | 0.5 | 3.5 | 2.5 | 1 | 1.5 | 0 |
| **Van Trinh** | 3 | 1 | 4 | 0 | 3 | 0 |

The programming language competency increased from level 0 to level 5 with level 0 denoted that he/she never used that programming language before and level 5 denoted his highest competence in that language.

This evaluation table helped us to decide which technologies and platforms should be used for this project.

* 1. Methodology Brainstorming
     1. Scrum process

In this project, we chose Scrum process as our development methodology. Scrum is an iterative, incremental process for developing any product or managing any work. Scrum produces a potentially shippable product at the end of every iteration. In Scrum, you break a task into a series of iterations, called Sprints, which get you ever closer to the goal. The team acts in a self-empowered and coordinated manner to ensure investment of a common goal and commitment to that goal.

There are several reasons why we chose Scrum. First, we all would like to learn more about Scrum with hands-on experiences. Second, our team and our project have characteristics matching with ones required by Scrum.

* Small teams: Scrum flourishes with small, face-to-face teams of 12 or fewer members. In our team, there are only four members.
* Small project: Large projects may not fit the Scrum methodology, which are targeted to smaller teams.
* Iterative fashion: Scrum works in an iterative fashion, in successive cycles, without an overall and upfront, must-do plan. In this project, we wanted to build our system with a few features first and increasingly added new features. We did not need to consider all aspects or plan all features of our project from beginning to end before the project even started.
* Collocation: In Scrum, this term indicates a preference that everyone is in the same location. This is also applicable to our team.
* Key personal quality: In addition to programming skills, we self-assessed that we possessed key qualities required for a Scrum team such as being experienced, motivated, committed, responsible, and independent programmers, as well as being team players.
  + 1. Team structure

A Scrum Team consists of three components: the Scrum Master, the Development Team and the Product Owner. We assigned roles as below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Description** | **Members** | **Note** |
| Scrum Master | A person who is responsible for implementing Scrum methods, values, and practices. He keeps the project on track and moving forward. | Dung Dam | He has experiences with Scrum process when working in his previous company |
| Development Team | The Development Team consists of professionals who actually implement tasks. | All members |  |
| Product Owner | The Product Owner represents for the customer. He knows what the customer wants for the product under development to do. | All members |  |

Though Scrum prefers separated people hold different roles, in this project we decided that all team members would hold the roles of the Development Team as well as the Product Owner. As the Product Owner, all members would be able to discuss and contribute their ideas about the features of the system.

* 1. Software Development Process
     1. Product Backlog

At the first step, we, in the roles of the Product Owner, wrote down all requirements we can think of for our product. These consist of functional and non-functional requirements. They go into a Product Backlog in Scrum process. The items in Product Backlog are ranked in order based on both business values and development effort. Tying the Product Backlog and the business value of each listed item is the responsibility of the Product Owner. The estimated effort to complete each backlog item is, however, determined by the Development Team. This Product Backlog is almost always more than enough for a first sprint. It then can grow and be changed as more is learned about the product and its customers. Top-ordered Product Backlog items drive immediate development activities.

Table of Product Backlog and priority (with 1 is the highest priority)

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Priority |
| 1 | Search shortest duration routes with multiple results | Given a user’s start time, starting point and ending point, the system will return multiple results which include bus routes that the user can take based on the shortest duration criteria. | 1 |
| 2 | Search earliest arrival time routes with multiple results | Given a user’s desired arrival time, starting point and ending point, the system will return multiple results which include bus routes that the user can take based on the earliest arrival time criteria. | 1 |
| 3 | Displays multiple results | Display multiple results in graphic user interface, display details when user clicks on a result. | 1 |
| 4 | Search place by name | User inputs a name that they want to find. The system will return all places that match that name. | 1 |
| 5 | Highlight bus stops on map | Highlight bus stops on map with marker. | 1 |
| 6 | Web service for communication between back end and front end | Provide a web service that allows the back end and the front end to communicate with each other. | 1 |
| 7 | Sort search results | Sort search results based on one criteria. | 2 |
| 8 | Responsive design | The system will change User graphic interface based on which devices are used to access the system, for example, displaying in different resolution depending on if the device is a smartphone or a tablet. | 2 |
| 9 | Auto detect user’s current position | Auto detect user’s current position based on GPS signal or network signal. | 2 |
| 10 | Highlight routes on map | Highlight segments of routes on map. | 3 |
| 11 | Store previous search results | Store previous search results for a user. | 3 |
| 12 | Compare 2 search results | Compare two search results for a user. | 3 |
| 13 | Intuitive search results | Intuitive search results. | 3 |
| 14 | Real-time tracking & re-planning | Real-time tracking & re-planning. | 3 |
| 15 | Suggesting nearby restaurants/bars/… | Suggesting nearby restaurants, bars, etc. along user’s bus route. | 4 |
| 16 | Real-time bus tracking | Track bus’s position in real-time manner. | 5 |

We decided that this list would be re-evaluated after each sprint to add new features or remove existed features if needed.

After the creation of Product Backlog, Scrum team will iterate through Sprints.

* + 1. Sprint Iteration
       1. Sprint Planning

We started a Sprint with a Sprint Planning Meeting, in which we made a plan of the work to be performed in this Sprint. In this face-to-face meeting, the plan is usually created based on the collaborative work of the entire Scrum Team. The meeting includes two parts:

* Part One: determined what would be done in this Sprint. In this part, we focused on selecting the high-priority backlog items from the Product backlog, moving them to the Sprint backlog, and defining the Sprint goal(s). We defined a set of acceptance criteria for each task.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Sprint | Duration | Sprint backlog |
| 1 | Sprint 1 | 21/10 - 03/11 | Search earliest arrival time routes with multiple results  Display multiple results  Highlight bus stops on map  Search place by name |
| 2 | Sprint 2 | 11/04 – 11/17 | Search shortest duration routes with multiple results  Responsive design  Sort search results |
| 3 | Sprint 3 | 10/18 – 12/01 | Daily update bus route data  Data analysis and processing  Store previous search results |

* Part Two: estimated backlog items. First, we estimated how much time should be spent to complete a task. We started with the highest-priority tasks first, estimated the time it would take, and then continued with the lower-priority tasks. In this project, we logged time spent on each task in a shared Google document file. Then, we used that time and working information from last Sprints for this sprint estimation. By this way, all members would not be too ambitious for setting their predicted productivity too high. After that, each team member considered how much time he or she would be able to contribute to this sprint. In this part of Sprint planning, we also discussed about the solution for such tasks involving data structures, algorithms, technologies, platforms, and programming languages. After all of the above were done, each member committed to tasks. From then until the next sprint, there would be no change in the sprint requirements.
  + - 1. Daily Scrums

We decided that we would work together twice per week, on Wednesday (from 17:00 to 19:00) and Saturday (from 13:00 to 17:00), at Evans library. At the beginning of each day, we spent 15 minutes on reporting the current progress. Each team member reported on what they had done in the previous days, what they would be working on until the next meeting, and what roadblocks they were facing. After that, it was the time for development.

* + - 1. Development

CODING

* For revision control, we used GitHub, a web-based hosting service for software development projects that uses the Git revision control system. This tool helped us to easily manage code changes of individuals and the whole team as well as effectively merge code from different branches. Before working on a specific task, each member pulled the latest source code from GiHub. Then, after completing a task and getting code review, we committed and pushed the new source code back to GitHub.
* Pair Programming: In this technique, two team members usually work together at one workstation. One, the driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in. The two members switch roles frequently. We used this technique in complex features such as building the bus route search algorithm.
* All members logged our working time and duration into a Google document file. That information was used for assessment and time estimate for future tasks.
* To make a highly visible, real-time picture of the total work, we used Trello, a collaboration tool that organizes projects and ideas into boards. In Trello, we categorized statuses of a feature into catalogues: Product Backlog, Known Issue, Defining Requirements, To-do, In-progress, Testing, and Done. In the Sprint Planning, we added tasks with its description and time estimate to Trello, then assigned members to tasks. All team members updated the status of the tasks frequently so that at any point in time, we could easily sum up the total hours of effort spent, and estimate the total work remaining.

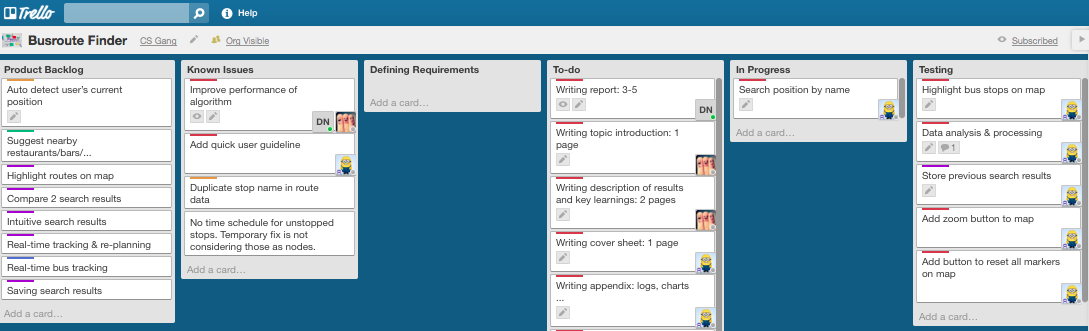


Figure : Trello's graphic interface

COMMUNICATION

Communication is an important part of Scrum process. For internal communication, we used Facebook chat and Google hangout, which is free and easy to use for online discussion and remote working. To reach out to the external source, we chose email as the main communication tool. For example, to obtain the bus schedules, geo data of all bus stops, and so on, we sent emails to the Transportation Services, kindly requested for those information.

TESTING

Each member was responsible for unit-testing each functions he or she ever developed, including writing and running the tests. Then, we performed an integration test when integrating the back-end to the front-end via a web service. Finally, we carried out a performance test. It started with small data and simple cases, then with heavy data and complex cases.

DEPLOYING

We deployed our system to Amazon Web Services. The system can be accessed at http://aggiehack.dngdiary.com

* + - 1. End-Of-Sprint Review

Sprint Review: concerned with “what was done” in a Sprint. Each task is enumerated, along with its acceptance criteria, to see if those criteria have been met. We ran a demo of the tasks that had been chosen to work on and determined if each was done or not. Those was not done would be pushed back to the Product Backlog.

Sprint Retrospective: captured on “how we did” in a Sprint. All team members discussed about the good, the bad, and the ugly of the Sprint and suggested our solutions.