Reflective comments on the success of the project

Technical Perspective

At the early stage, the marking algorithm of the system was represented using Java Code. However, it will be more straightforward to implement the algorithm using PHP at the beginning (PHP is the straight scripting language running inside of the web system to ensure the interactive between database and website). After the general structure is set up, we meet with some incompatibilities when trying to embed the JAVA algorithm into the system. Fortunately, by adjusting the algorithm's framework, we succeed overcoming this problem.

It is rational choice for us using HTML as the main [markup](http://en.wikipedia.org/wiki/Markup_language) code for creating our [web](http://en.wikipedia.org/wiki/Web_page) site, because it can be supported by almost any browser and offers the way to create [structured documents](http://en.wikipedia.org/wiki/Structured_document) by implementing structural [semantics](http://en.wikipedia.org/wiki/Semantic) for text. The compatibility of web scripting language should be considered as well. Due to the excellent expansibility, HTML is really an ideal choice.

For the selection of scripting language to support web dynamic development, we reasonably choose the PHP and JavaScript. PHP code can be straightly embedded into our [HTML](http://en.wikipedia.org/wiki/HTML) source document without using an outer file to run the data and provide the perfect link between the user interface and the database level. Basically our system is running based on the interaction of user actions and database query. It gains great convince in making use of PHP as the main script language.

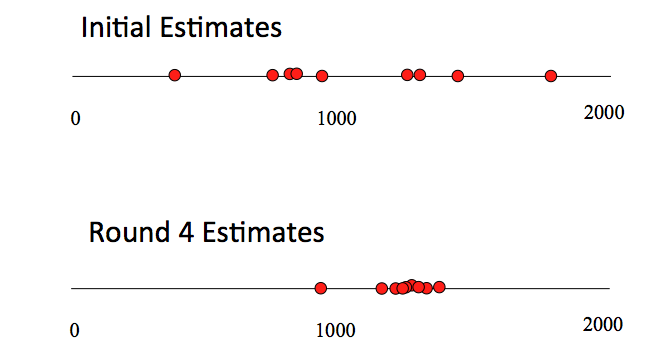
There are some dynamic graphic functions in the system. However it tends to be quite slow if large numbers of users trying to see the diagrams, because it takes huge CPU for web server to interpret these files. Hence we implement the JavaScript to solve this issue. JavaScript is executed at the client side, ensuring that the code is running on the user's processor instead of the web server. It saves large strain and bandwidth for our server, as the data does not need to be processed in the site's web server and sent back to the user consuming local as well as server bandwidth, which guarantees the accessing speed requirement.

We each had a MySQL account linked to our Computer Science usernames – this meant that each person’s MySQL was different. This meant that to resolve this issue, the team member that was responsible for creating the database had to export it from his MySQL account and then upload it to the repository ，which means that each of the other members had to import that file into their MySQL so they could work with the current database. It is pretty inconvenient but we keep doing this for a long time at the early developing stage. Afterwards, we realized that it could be resolved if there was a shared MySQL account for the group that the team could have logged into. We absorbed lesson from this mistake that every small decision, which is concerned about developing efficiency, should be well considered.

Although most of the functions have been realized in the end, there still remain some parts unimplemented. Such as the “Database Export”(because generally the users do not need any exporting documents), as well as some advanced graphing functions (it may take time for users to further learn these mathematic tool to help analysis the data). The system is developed with the concept of “Simple And Stupid”, so some unnecessary functions are excluded.

Project Management Perspective

Job Equitable Distribution: At the early stage, we always come across the problem for unequal work distribution due to the difficulty for predicating the time cost for different tasks. So we cannot ensure the concurrent of each progress. To alleviate this problem, the “Parkinson’s Estimate” is implemented. For each remaining tasks, we use an unlabelled distribution plot to carry on. Each member point out their estimation and explains the rationale behind them. The process is repeated several times until the estimated time converges.



At the middle term period, we found that the progress of our project is behind the estimation. To accelerate the development, we have scrum meeting every day and implement the Agile Software Development method. Basically, we predict how many works will we finished during this period (using the Parkinson’s Estimate). Afterwards, the big job is split into small tasks (sprint backlog). These tasks will be handle one by one with very short developing cycle. However the general development is an iterative progress. This flexible characteristic makes it suitable to adapt all kinds of changes during development.

Task Scheduling: Since we could not feasibly all work on all sections at the same time, it became necessary for us to divide out the workload. This was resolved by having each member of the group states what their strengths were (coding in specific languages, writing skills etc.). We then all sat down and discussed how splitting it into certain skill areas, and then attributing those skills to team members who felt strong in that area. Another minor problem was that whilst there was an algorithm necessary to be built, it was only needed near the end. This was resolved by having the team member set to work on that temporarily re-assigned to help out in the database.

Language Barriers: Within the group we had a team member whose native language was not English, so this led to a few mix-ups early on as we tried to understand each other. This was resolved as we learnt how to make each other understand better.

Generally, the group has a constructive team approach to solve problems. There is trust, respect and full participation by all members. Sometimes some issues like misunderstandings and poor communication, different expectations as well as ambitions and priorities, ambiguities about responsibilities may arise. But we finally come over them with regular self-reflection and communication with each other.