**QA Concept**

In our quality assurance concept, we want to target 3 main things: the efficacy of our code, number of logging statements used in comparison to the total lines of code and three metrics we have chosen to analyze our code.

These 3 main things will help us understand our code better, thus making it easier to improve the project in the future. The total lines of code will also help us visualize how we’ve progressed through the analyzed points in time.

The *code coverage* will help us determine what parts of our code has and hasn’t been tested yet. The main components we have chosen to test in this part is our *ProtocolExecutor.class* and *SBListener.class.* The *ProtocolExecutor* exectues all commands from the client. For example, sending a message to all players and other such similar client/server communications. Testing the logging process of each client, testing the card sizes at the beginning of each game, testing that the cards are being dealt correctly at the beginning is all a focus of our tests for this class. These things are all a huge contributing factor to the success of the game.

The *number of logging* statements will show us how well we’ve used the logger and how often. This is important to see as logging statements help a lot for debugging at every stage, and thus shows us how well logged our program is and how it facilitates easy debugging.

The *3 metrics* we have chosen to analyse here are:

The *total lines of code* metric will show us which classes we have invested the most work in and will also show the average length of classes. This is very useful to have an oversight of all classes and to see if there are classes where things can be shortened/branched off. It will also help keep track of our project through the the timeline and to compare with our GANTT Project Timeline. It is a good way to see how well we’ve stuck to our project plan and if the parts of code were expanded around the time they were supposed to be expanded at.

We will also take a look at how many Logging statements we used (Apache Log4j2) to determine where and how well we used this library for debugging and logging purposes.

servLog (25.) – 66   
Client – 37

total: 103

***Metrics***

The data points that we have chosen are from the 5th of April, the 20th of April and the 26th of April. The 5th marked the end of MS3, the 20th was right after most of our GUI was completely implemented, and the 26th was the day before the MS4 deadline.