Quality assurance, control Group 2

Making sure that the result of our project will be high quality product is something that we must start working towards from the beginning. There are several different ways to assure high quality results. To assure external quality towards customer we have decided to try pair programming, test driven development and continuous integration.

Pair programming is a nice way to code, because it gives members of the group a chance to exchange ideas and come up with various ways to solve a single problem. In pair programming finding and fixing bugs also happens a lot faster, two people are more likely to spot a mistake than one individual. Test driven development is basis of quality assurance for our project. TDD allows us to build software that compiles. First, we write the simplest test, which captures main point of testable method. Then we write the method itself and make sure it compiles. Next, we modify test and actual method, so they would correspond more to the result we want to get. If all goes as planned, eventually we should end up with piece of software that does exactly what is expected of it. Conditions integration is also an important part of quality assurance. We push and pull code from GitHub repository very often. Every time a part of system is coded, the code is uploaded by creator and downloaded by other group members. That way we assure that each group member has the latest version of the project and people do not write code that doesn’t match with already existing code. Most of our group’s work will be done at a specific location where everyone must show up. We work as a team not as individuals. That way the communication inside the team is way better, clearer and simpler, which also helps with assuring high-quality results.

Quality control is also an important factor in our project. Testing the product when we finally get it will be crucial. We must make sure that all user stories, different scenarios, all attributes and overall appearance work. We will compare every user story to the product to see if every one of them can be done. We will test all input field, so even if given incorrect information, the system still knows how to continue working. Overall appearance is something that cannot be overlooked either. Product must have appearance that is appealing, simple and yet complex enough to keep it interesting. Testing all these factors on product is how we plan to implement quality control in our project.

In software it is also essential to keep high maintainability, reliability, usability and reusability. For code to be easily maintainable it is useful not to use many layers of inheritance. It’s also a good idea not to put several short lines of code into the same line. Easily maintainable code has clear communication lines inside the system. UI shouldn’t be able to directly call DB in one method if in every other method UI has to call Business layer first. It also helps to ensure reliability of the code, if method needs to be changed. Another way to keep up reliability is to reduce the number of errors in code and to keep different parts of code separated. Putting all methods that deal with users, products and services into the same class reduces reliability and makes it harder to read and use this piece of code. To make sure one doesn’t lose points in reusability it is advised to write simple methods that other bigger methods can call instead of writing smaller method into ever bigger method. The simpler and shorter those simple methods are, the more they can be used in various other methods. For example, a method that only searches for a person based on first name, can be called from far more methods than a method that searches for a person based on first name, last name and email together. To make software as user-friendly from developer’s point as possible, it is not good idea to make too long pieces of code or to merge methods.