Quality Control

Dear Utah Valley University:

We at After Innovation Education pride ourselves in the quality of our products. We hold it as one of the key components of software engineering, and take it very seriously. This purpose of this paper is to describe how we at Innovated Education control the quality of our products and ensure the highest quality of the deliverable to our clientele. This paper will describe how we keep the code simple, ensure project requirements are met and exceeded, create a user friendly graphical user interface, ensure the code is portable, flexible, and testable, and how we handle errors.

First and foremost, we at After Innovation Education keep our code simple and easy to maintain. We can't always be there to continually maintain and innovate on the program after it's turned over to our valuable clients. Therefore, we prioritize simple and easy to read code. We ensure the code remains modular by maintaining a high standard of loose coupling and high cohesion. That way when changes inevitably need to be made to ensure you keep up to date with the best in program quality, you can meet those needs quickly with minimal effort and cost to you.

Next, we ensure project requirements are met and exceeded. This starts with understanding your unique requirements at every step of the project. We ensure that strong communication lines are maintained during planning, building, and testing. During planning phases we would like for a member of UVU staff to sit in on the planning meetings to describe your unique needs to our team of highly trained engineers. We will make plans on the spot to ensure those needs are met by our team and to ensure that you are totally satisfied with the final project on the first iteration.

Third, we ensure the graphical user interface (GUI) is entirely user friendly. Writing a program is more than keeping code simple and doing the bare minimum. We exceed all expectations and requirements, and this is one of many areas we go above and beyond. Maintaining a simple user interface keeps your users happy and coming back to the program time and time again. As an educational software, it's important that your students have an easy time using the software, and an easy time completing their homework and the projects you give them. Everything from page design to layout even as granular as font readability is a consideration as we design the perfect user interface.

Fourth, similar to keeping easily maintained code, we maintain a high standard of portability. Without relying a lot on API's, we ensure that the code can be run on many different systems without relying on dependencies. This is an important point of software maintainability. This ensures that if your hardware, servers, or frameworks change, you can continue to run your software with minimal downtime.

Fifth, by ensuring the code's testability, we can ensure its high standard of quality for many years to come. By maintaining the program's modular nature, and well planned architecture, we can ensure that even when you have taken over maintenance of the program, you can continue to test new features based on a strong foundation of good, testable code. Having testable code is important so you can pinpoint where any errors crop up or where new changes need to be made. When a moving piece fails, it's important that you know exactly where that piece failed and how it failed. Therefore we also establish our code with strong test cases built in so minimal work is needed on your part. Our thorough structure is top priority, and we take all angles into account.

This leads me to my sixth point. Our code always has strong error handling mixed in. We build our code on a firm foundation, and this project is no different. With strong error handling, you can ensure that users know the bounds of the program as well as how to use it properly. No guesswork involved. Not during testing, and not during end user experience. With strong error handling, you can ensure, similar to testability, that your code remains at the highest standard of quality for years to come.

Finally, we ensure that the code is flexible. Due to its modular nature, our code maintains a high degree of flexibility. That way a year from now, two years from now, five years from now, or ten years from now, you can adapt the code to fit whatever new architecture or curriculum you find yourself using. When we create a program, you can rest assured that your investment will take you a long, long way into the future. No matter what happens, transferring our simple code to your new structure, or evolving the program to evolve to conform to your very unique changes in circumstance and needs, will always be a simple process.

In conclusion, we at After Innovation Education take our quality control very seriously. It is our hope that this paper has proven the fact that it is always at the forefront of our minds, and that we put quality at the highest rung of priorities. By choosing us, you have ensured a top quality program that other educational institutions will turn green with envy over. Quality is important to you, and it's important to us. Our programs are always simple and easily maintained. We always ensure that the project requirements are met and exceeded. Our graphical user interfaces are always user friendly and easy to understand. Our code is highly portable and easy to transfer if needed. Our programs are always testable, ensuring a high quality as you evolve it to fit your needs. With strong error handling and a high degree of flexibility, it is our hope that you will always find use for this program. As you continue your noble venture of teaching the up and coming youth of tomorrow, it is our privilege to work with such an esteemed establishment and hope that you will call on us again in the future for all of your programming needs.

-After Innovative Education (IED)

Daniel Espinel *Erik Manley* *Santiago Rameriz* *Anthony Peterson*