**Quality Assurance Process Summary (15 marks)**

The Team A practically went through several difficulties with the development of the PHP MVC framework, however, with much effort and team work, we tried hard to detect, prevent and execute few approached to combat both development and adverse issues regarding the project.

| Items | Steps taken | What still has to be done |
| --- | --- | --- |
| **Test Plan** | * Though not much testing were executed during the initial development of the framework, several general PHP methods were used to test database connections, unnecessary variables and artifacts that might cause error to future problems | * Should have written test plans for each sprint * Each team should build small and testable code at the beginning of the sprint * A team member should be delegated to track and monitor progress of tests |
| **Versioning** | * Team A did setup a Github account and also created a repository for the team to collaboratively, push and pull different version of the code that were being updated * Team members also have their own branches to avoid run their own test and code before merging into the group repo | * Though each team members having their own Github branch, it’s better to limit too many branches and only work together rather than separately * Possibility of a branch for each user story, once it’s done it’s merged into the main * Branches for each weekly sprint |
| **SCRUM Meeting** | * The framework went through much collaboration among team members, onsite and offsite. And as much as possible unsafe and unnecessary code or features was eradicated before it becomes code. By having this approach we minimize problems that may run out of control | * SCRUM master should be aware of their roles and a proper task allocation sheet should have been drawn so each members task can be monitored |
| **Planning** | * Team A group setup a feature list where person and features of the framework is created and allocated to team members * Sketches were also used and modelling were also drawn to demonstrate workflows of logic of the framework core structure | * Too much effort was put into planning, however, prioritization of features of the framework should be given first with concise procedures |
| **Test Driven**  **Development (TDD)** | * TDD was used a mid-term of the project. Test was integrated with structure inheriting MVC pattern. * Several test cases for product, categories, and customers, database connection and clean URL. | * TDD should be executed at the start with simple test and all team members should contribute to test cases |
| **Security** | * Login were encrypted using MD5 API’s * Data were well sanitized * DB connection was encapsulated into a JSON format * Abstract classes were inherited by child classes (Controllers, Models, Views) and acted on Separation of Concerns and DRY * SOLID principles were adapted with robustness to protect SQL injection | * Input field data validation * Build security at the start of the project |