



Module Reflection

Course: MSc Computer Science

Module: Secure Software Development (Computer Science)

Assignment: ePortfolio

Date: Sunday 31st October 2021

Student ID: 126853

E-Portfolio Links:

GitHub Pages: <https://eportfolio.kieronholmes.me/modules/secure-software-development>

Module Reflection:

The Secure Software Development (Computer Science) module of the MSc Computer Science course has introduced us to the concepts of developing maintainable code, which is produced in line with security best practices. In addition, this module has provided a valuable introduction to many crucial topics within software development, including the approaches to software development (Agile/Waterfall), Linters/Style Checking Tools and the OWASP Top 10.

The first aim of this module required us to identify and manage security risks as part of a software development project (UoEo, n.d.). Throughout this section, we were introduced to a selection of the most common vulnerabilities, in the form of the OWASP Top 10, which included common issues such as the lack of logging, proper access controls and SQL injection vulnerabilities (OWASP, n.d.). From this, we learned the technological measures that can be implemented to develop an application free of the main vulnerabilities.

The second aim of this module was to analyse development problems and determine methodologies, tools and techniques that can be used to solve them (UoEo, n.d.). Within this section, we learnt that although there are two distinct development methodologies (Agile/Waterfall), the Agile methodology is far more prevalent within software development scenarios as it allows you to develop in independent sprints

and change previously completed work. As a team, we had used this methodology within our Unit 11 assignment to assist us in timely producing a high-quality product.

To implement the Agile-style workflow in a remote development situation, we chose to hold a team meeting and split the software development project into distinct sections, which could be worked on in the form of 'sprints', which were listed as Issues with the 'enhancement' tag within GitHub. This allowed team members to choose areas in which they wanted to expand their knowledge, which would then be pulled into the main repository for testing and submission.

The third aim of this module was to create a solution meeting a design brief (UoEo, n.d.). This aim required us to select a scenario to develop a technical solution, enhancing the currently lacking security mechanisms. In this case, our team chose to select the CERN Large Hadron Collider experiment, providing a repository for researchers and employees to store work-in-progress research material, preventing malicious users or those from other institutes from gaining access to the resources. This section has helped me understand the importance of sticking to a design brief within a commercial environment. In addition, this section taught me the importance of creating a comprehensive design brief that can be used as a solid base for creating a code-based project.

The final aim of this module required us to develop and implement the skills to be effective members of a development team (UoEo, n.d.). This section required us to develop a code-based solution as a team, utilising collaboration tools such as GitHub and Codio to achieve the desired outcome. For example, the team I was in chose to use GitHub combined with the PyCharm IDE to develop – whilst implementing a

branching policy to require that all code contributed to the repository was peer-reviewed to ensure reliability/consistency (GitHub, n.d.).

Implementing Branch Policies and automated workflows/testing was incredibly important with our final code project. It ensured that all three team members (with distinct coding styles) produced easily readable and maintainable code, which was tested in line with a set of test cases based upon the design brief document.

During the 12 units of this module, I actively participated in the Discussion Forums and individual assignments, sharing the knowledge I have learned during this module or through my professional career. Although I work as a Web Developer for my day job, I had not come across or used the Python Django framework before starting this course. Therefore, a series of Documentation pages and team discussions proved to be a precious resource when learning how to implement security measures within this framework. Furthermore, all of the knowledge learned during this module will be useful when working on future Python projects where security is paramount.

A large element of this module focused on developing and contributing as part of a group, showing a diversity of opinions and prior knowledge, which can help prevent a developer from getting tunnel vision (ProgrammerInterview, n.d.). However, members within our team also had a limited understanding of the Django framework, meaning that the speed of development was reduced due to the different learning paces. In such cases, I suggested holding a short team meeting to produce a deliverable covering one of our 'sprint' items. This proved to be a handy tool, as it

allowed all members to provide real-time feedback on code, suggesting alternatives where they would be more efficient.

Throughout this module as a whole, I have completed various tasks to demonstrate my understanding of the Secure Software Development topics taught within this module, applying my knowledge to a real scenario based on the CERN Large Hadron Collider project. During this module, my understanding of the Django Python Framework has been enhanced and the practical application of the Agile/Waterfall software development methodologies. This knowledge will be transferred into future projects I develop or manage and integrate with upcoming modules.

References:

GitHub. (n.d.) Managing a Branch Protection Rule. Available from:

<https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/defining-the-mergeability-of-pull-requests/managing-a-branch-protection-rule> [Accessed 31st October 2021].

OWASP. (n.d.) OWASP Top Ten. Available from: <https://owasp.org/www-project-top-ten/> [Accessed 31st October 2021].

ProgrammerInterview. (n.d.) What is tunnel vision in software design and engineering?. Available from: <https://www.programmerinterview.com/american-vocabulary/what-is-tunnel-vision-in-slang/> [Accessed 31st October 2021].

UoEo. (n.d.) Secure Software Development (Computer Science) – Module Home. Available from: <https://www.my-course.co.uk/course/view.php?id=7062§ion=0> [Accessed 31st October 2021].