

Summative Assessment: e-Portfolio Reflections

E-Portfolio

Link to my e-Portfolio: <https://gi0rgi-k.github.io/eportfolio/index.html>

By clicking on “Secure Software Development” on the main page, you will be sent to this module’s page of my e-Portfolio:

https://gi0rgi-k.github.io/eportfolio/ssd_module.html

Here, you can find descriptions of what I have learned in each Unit of the module.

Under each Unit, you can click on “LEARN MORE,” and you will be redirected to a respective folder of my SSD Github repo, where you will find all of the artifacts and tasks I have accomplished for the respective Unit.

This is a direct link to the Github repo:

<https://github.com/gi0rgi-k/SSD-UoE/>

Professional Skills Matrix

Skills	Skill Level Grade	Skill Level Grade
	<i>Beginning of Module</i>	<i>End of Module</i>
Time management	4	4
Critical thinking and analysis	4.5	4.5
Problem-solving	4.5	4.5
Communication and Literacy skills	3	4.5
IT and Digital	4	4.5
Numeracy	4	4.5
Critical Reflection	4.5	4.5
Python OOP	3	4
Software Design (UML)	2	4
Academic Writting	4	4

Skill Level Grade: 1-5

Figure 1. The increase in skill level after the SSD Module.

Reflective Essay

As a Product Manager of Data Platforms with a non-technical background, my approach to learning technical concepts essential for communicating with engineers and designing technical systems has primarily been through on-the-job experience. I enrolled in the Secure Software Development (SSD) module to complement this with a theoretical and academic understanding of security frameworks.

This reflection captures the knowledge, skills, and personal growth I have gained throughout the SSD module, structured using Rolfe et al.'s (2001) model.

Additionally, I incorporate elements of the Gibbs Reflective Cycle where relevant (Galli, 2022).

What?

The SSD module provided me with a foundation in secure software design and development principles. It combined theoretical understanding with practical applications through coding exercises, group assignments, and the development of an eShop application.

1. **Security Principles:** The module emphasized various industry best practices such as OWASP Top Ten vulnerabilities (OWASP, 2021), ISO/IEC 27000 (Meriah & Rabai, 2019) and GDPR compliance (GDPR, 2022), directly aligning with my role as a Product Manager of Platforms, where security is critical.
2. **Hands-On Experience:** I gained technical proficiency in secure coding, testing, and development. Through coding exercises, these are some, but not an exhaustive list of security features I implemented:

- Authentication and MFA.
 - Session management.
 - Regular expression (regex) input validation.
 - Role-based access control (RBAC).
 - Leveraged Flask APIs for company ID verification (Palletes, 2010).
 - Threading for running the API with CLI for a better user experience in the App.
 - Logging.
 - Encryption.
3. Collaboration and Agile Practices: Collaborating on the Design document as a group emphasized the significance of effective communication, project management, and teamwork. By utilizing Trello and following Agile ceremonies, I experienced firsthand how an iterative development process allows for a faster feedback loop, enhancing efficiency by enabling the group to quickly align on necessary adjustments (Parabol, 2024). In addition to holding weekly meetings for in-depth discussions, we used Google Drive to collaboratively edit shared files and WhatsApp for quick communication. This approach mirrored Agile principles, where short iterations and prompt communication streamline fast adjustments, in this case for the design document.
4. Modeling Tools: Learning and applying UML diagrams enhanced my ability to design and communicate complex workflows. This skill was valuable during group assignments and is equally applicable at my work for communicating with engineers and various stakeholders.

5. Secure Testing and Tools: Tools like linters (flake8, pylint, etc.) and vulnerability scanning taught me how to detect and fix security flaws. This also aligns with my work, where ensuring high code quality and security are essential.

So What?

This module profoundly transformed my perspective on software development.

Beyond its professional relevance, it contributed to my personal growth by equipping me with new skills and a framework for approaching application development with a security-first mindset.

1. Deeper Appreciation for Security: Before this module, I often viewed security as an obstacle to rapid feature release at work. Now, I understand that security is a shared responsibility across all roles, not just the Security team. By embedding security at every stage, rather than treating it as an afterthought, I can better support engineers and align with legal and compliance teams, which will also help with creating a more collaborative culture across teams. While thinking about this, I came to learn that the concept that every role is accountable for security, is widely known and studied (Ross, 2021). This mindset will be very valuable in my future career growth.
2. Enhanced Communication and Collaboration:
 - UML diagrams and flowcharts offered a structured method for communicating complex ideas during the creation of the design document for the eCommerce platform and in collaborative discussions (Rumbaugh et al., 2004). I focused on developing detailed flowcharts,

which also facilitated the creation of class diagrams and was instrumental in writing Python code aligned with the app's detailed functionality and edge cases from the diagrams. Given the need to communicate with diverse stakeholders at work, such as engineers, legal teams, clients, and management, mastering the visualization of designs through UML is immensely beneficial, since it minimises misunderstandings about the intended outcome of the project.

- Collaboration with team members emphasized the value of respect, flexibility, and compromise in group settings. Balancing diverse perspectives and collectively reaching final decisions through discussions and voting gave a strong sense of team belonging and made the entire experience enjoyable. Despite division in tasks according to our expertise, we effectively built upon each other's contributions. For example, the flowchart was derived from features outlined in the use case diagram, and the class diagram was subsequently created using both of these as references. Although different team members developed these diagrams, they were perfectly aligned in terms of features and requirements. Such success in teamwork will be valuable as a standard to strive for when I will be building new teams for various projects at work.

3. **Strengthened Technical Skills:** The hands-on exercises in this module enhanced my ability to understand and implement secure features, making me more effective in guiding technical decisions:

- For example, the eShop application helped me understand the technical aspects of security mechanisms. This allows me to better evaluate system security in my current role.
 - Learning about linters and GDPR-compliant coding equips me to identify potential vulnerabilities early, improving both code quality and regulatory adherence. This is a major benefit, as I will not have to simply rely on engineers or legal teams' expertise on security and regulatory topics while I focus solely on functionality, but can collaborate with them and make sure that software is compliant and developed with secure principles.
4. Personal Growth: The module expanded my technical confidence, particularly in Python coding and designing systems with security in mind. I can now communicate more effectively with some of the engineers I work with as I understand their code and the standards they need to follow better.

Now What?

Looking forward, I am set on applying the lessons from this module to both my professional practice and ongoing personal development:

1. Promote Security by Design: I will advocate for security-first approaches in every project. I aim to build systems that are secure by design rather than treating security as an afterthought.
2. Foster Collaboration:
 - I plan to strengthen partnerships with Security and Legal teams, building trust and shared accountability for secure system design.

- UML diagrams will allow me to bridge the gap between technical and business teams, improving both system design and stakeholder alignment.

3. Apply Practical Coding Knowledge:

- I will advocate for tools like linters and vulnerability scanners to improve code quality and security.

References

Galli, F. (2022) Gibbs' cycle review. Emotions as a part of the cycle. e-Motion

Revista de Educación Motricidad e Investigación 92-101.

DOI:10.33776/remo.vi19.7224

GDPR. (2022) Complete guide to GDPR compliance. Available from: <https://gdpr.eu/>

[Accessed on 24th November 2024].

Meriah, I., & Rabai, L. B. A. (2019) Comparative study of ontologies based iso 27000 series security standards. Procedia Computer Science 160: 85-92.

OWASP. (2021) OWASP Top Ten 2021: The Ten Most Critical Security Risks to

Web Applications. Available from: <https://owasp.org/www-project-top-ten/>.

Pallets (2010) Flask. Available from: <https://flask.palletsprojects.com/en/stable/>

[Accessed 1 December 2024].

Parabol (n.d.) 45 User Story Examples to Inspire Your Agile Team. Available from:

<https://www.parabol.co/blog/user-story-examples/> [Accessed 15 November 2024].

Parabol (n.d.) *45 User Story Examples to Inspire Your Agile Team*. Available from:

<https://www.parabol.co/blog/user-story-examples/> [Accessed 15 November 2024].

Rumbaugh, J., Jacobson, I. & Booch, G. (2004) The Unified Modeling Language

Reference Manual. 2nd ed. Addison-Wesley

Rolfe, G., Freshwater, D. & Jasper, M. (2001) Critical reflection in nursing and the

helping professions: a user's guide. Basingstoke: Palgrave Macmillan.

Ross, R. (2021) Security Is Everyone's Responsibility. Available from:
[linkedin.com/pulse/security-everyones-responsibility-ron-ross/](https://www.linkedin.com/pulse/security-everyones-responsibility-ron-ross/).