Learning diary

Module 1

The first lecture is a nice introduction to requirements engineering in software development. I learned what requirements engineering is, why is it needed, the different types of requirements, the skills needed to analyze them, and how to establish product vision and scope. I plan to become a cloud software architect in the future, and I believe the knowledge from this course will help me a lot. Designing a complex software system will be easier if the specification and requirements are well planned beforehand.

Lecture 2 highlighted the approach needed for identifying and prioritizing requirements, emphasizing the challenges in making requirements clear due to varying perspectives between stakeholders. For my role, understanding these can help me navigate project needs more effectively, which is crucial when designing cloud solutions that must meet both immediate and long-term goals.

Lecture 3 was about the importance of documentation and traceability in project success. We studied best practices for writing clear and verifiable requirements and the structure of the Software Requirements Specification (SRS). I can apply these practices in my own work, particularly as systems become more complex and interconnected in the cloud.

Lecture 4 brought sustainability into the picture, introducing frameworks like the Sustainability Awareness Framework (SusAF) and the Framework for Sustainability of Software System Design (FSSSD). Sustainability isn’t always the top priority, but with cloud infrastructure impacting resources, this perspective helps me consider efficient design choices that support long-term environmental goals.

Module 2

Reflecting on the experience of designing and using the sustainability-focused product backlog template, I learned that creating a backlog is an iterative process that closely aligns with Agile development. The process involves continuous refinement, requiring flexibility to adapt to new inputs and changing project dynamics. One of the key takeaways from this exercise is that no universal backlog template can cater to every project or scenario. Each project has its unique goals, constraints, and stakeholders, making it essential to design a backlog that reflects these specific factors. This fact has deepened my understanding of the role of a project manager, who must consider multiple factors when creating a functional and effective backlog.

One of the advantages of the new template is its ability to promote sustainability awareness among team members. By explicitly documenting sustainability impacts alongside user stories, the template encourages developers and stakeholders to think beyond immediate deliverables and consider long-term effects.

However, there are challenges associated with this approach. Adding sustainability-specific sections and indicators can make the template more complex, which may slow down initial adoption or overwhelm users not familiar with sustainable software development. Furthermore, designing and refining a custom backlog template is a time-intensive task, especially when balancing multiple priorities and incorporating feedback from various stakeholders.