Unit 2

# Reflection

In this unit, consideration was given to integrating security into agile SDLC. I found the activity related to this module rather fascinating for two reasons: (1) most agile process do not consider security in their phases and (2) the Microsoft Security Development Lifecycle seems to be the best match for *development teams*. It was thought-provoking to contemplate how each phase of the standard agile process should be adapted to incorporate security considerations. I found the biggest takeaway mapping security concerns to an agile process to center on

- Security Requirements.
- Trust Boundaries/Attack Surface.
- Risk analysis.
- Static code analysis.
- Vulnerability testing/penetration testing.

Analysing these security concerns I could not agree more about their relevance to real-world security concerns (given my experience as a software engineer and architect). I feel enriched by this knowledge because of its *relevance.*

Next, a blog post was generated that centered on the language and concepts of ISO security. This blog post was aimed at cementing an understanding of the role *people* play in cyber security attacks. To me, the biggest concern is the need for *policy* to handle security events, and also the need for *education* to ensure an organisation's work force is well-equipped to handle the ever-growing range of cyber security attacks.

Our team continued to communicate using Slack, and we held another team meeting to both review and feedback on the initial class diagrams generated by the team's designated architect. I thoroughly enjoyed the openness expressed by the architect to take onboard the feedback provided by the team and their willingness to also defend and explain various design decisions they took. The collaboration between team members is good and professional, with each providing feedback designed to assist each other.