**Unit 2: UML Modelling to Support Secure System Planning**

Welcome to Week 2, during which we will gain some practical skills in the creation of UML flow charts. Flow charts are used during the design stage of the SDLC and can be used from that point onwards to support communications between stakeholders and to act as documentation of the final product created.

**Objectives:**

* + Gain an awareness of the opportunities for integrating software security development practices at each stage of the agile SDLC.
  + Understand the range of ways in which software security can be problematic.

**Outcomes:**

* + Research the academic literature (e.g., published by IEEE, ACM, Elsevier or ScienceDirect) on software development which follows an agile process and prioritises security.
  + Become familiar with the language and concepts of a security standard which has been published by the ISO/IEC.

**Reflection:**

Diagrams of all shapes and sizes can be found in typical UML models, each showing a different perspective on the system you're modelling.

Software development can be thought of as a three-stage process: conception, design, and development. The Waterfall methodology, the Agile methodology, and the Iterative approach are all examples of SDLC frameworks.

At every stage of the software development life cycle, a secure SDLC ensures that security-related activities are carried out. Penetration testing, multiple code reviews, and architecture analysis are all examples of this type of activity. Consequently, vulnerabilities are reduced or completely removed, ensuring the final product's integrity (SYNOPSYS, 2018).

Every stage of the software development life cycle can be tampered with. Developers can detect flaws, vulnerabilities, and weaknesses in their products before they have a negative impact on the end user. Incorporating security into your SDLC reduces costs and shortens development times, while also delivering high-quality results.

Viruses, Trojan horses, Logic bombs, Worms, Agents, and Applets are just a few of the security issues that must be addressed when creating an application. Security flaws can be introduced by software developers either intentionally or unintentionally (Stewart, 2012).

Organizations can use the ISO/IEC 270001 family of standards, also known as the ISO 27000 series, to improve their information security.

**References:**

SYNOPSYS (2018). *What is the secure software development life cycle? | Synopsys*. [online] Software Integrity. Available at: <https://www.synopsys.com/blogs/software-security/secure-sdlc/>.

Stewart, J. (2012). *CISSP Certified Information Systems Security Professional Study Guide Sixth Edition*. [online] Canada: John Wiley & Sons, Inc. Available at: https://www.limswiki.org/index.php/Software\_development\_security [Accessed 26 Jun. 2022].