Week 4 Assignment: Software Development Life Cycle Versus Security Development Life Cycle

Shaun Hoadley

CST316: Information Security Management

Professor Amr Elchouemi

August 2, 2021

* **Describe the four phases of the software development life cycle: the project initiation phase, functional design phase, system design phase, and testing phase.**

As we learned in previous courses, and refreshed in chapter nine of our course textbook by Wills, the software development lifecycle (SDLC) is the lifespan of software from conception to it’s decommission. For the most part, the SDLC consists of several phases including planning and requirements, architecture and design, test planning, coding, testing and results, and release and maintenance. Each of these phases is pretty self-explantory. Planning and requirements consists of coming up with an idea of what problem the software is supposed to address, how it will address it, what functionality is needed, and any other requirements such as how it will look, company objectives, and legislative or privacy requirements. Architecture and design is where those requirements are then organized to show how the system will work as a whole and how data flows through it. This is commonly done using UML models as a standard way of design and understanding. The next phase is test planning. Test plannining consists of developing the test criteria to ensure the software will meet the needs for the project. Next, comes the coding, or programming, of the software. After the software has been coded, it moves to the testing and results phase. The testing and results phase uses the methods developed in test planning against the actual software to ensure it works as intended and efficiently. Finally, release and maintenance is where the software is handed over to the client, whether that is a specific company or the general public, for use, and support for the software takes over for any issues or updates that may be required after release.

* **Compare the following software development life cycle models: the waterfall and the agile cycle.**

In older linear programming languages such as BASIC, COBOL, and FORTRAN, as well as in assembly language, these phases have to be performed one at a time, meaning the next phase cannot begin until the previous phase has been completed. This cascading of the phases is better known as the waterfall design because it moves in one direction like a series of waterfalls.

With the emergence of object-oriented programming languages, agile development methods also began to emerge. Agile development has the same (more or less) phases in the SDLC as waterfall development, but some of the phases can begin as other phases are still going on. With object-oriented programming, functions and features are built modularly, allowing testing to begin on the completed modules while work continues on the other modules. Additionally, agile development allows for rapid deployment of software by releasing in versions that allow a client to get up and running with a barebones release while future releases add in the extra functionality they wanted but were not strictly required to be operational.

* **Describe the phases of the security development life cycle. Describe which lifecycle model is more secure based upon your experience or opinion. Justify which would be more secure.**

Though it was challenging to make out what the security development lifecycle is from the course text by Wills, essentially what I gained from it is the addition of security measure phases within the existing SDLCs. Alongside the planning and requirements phase, incorporating a security requirements phase. With the architecture and design phase, add an architecture security analysis phase. In the test planning phase, developing security testing plans as well. In the testing phase, you can use code scanning tools and implement the security tests previously planned.

As for which lifecycle is more secure, it is my opinion the security development lifecycle is the more secure lifecycle. My justification for this opinion is simply because it adds in security focused design and testing to the regular software development lifecycle.

**References**

Wills, M. (2019). [(ISC)2 SSCP Systems security certified practitioner: Official study guide](https://ashford.instructure.com/courses/87741/modules/items/4439845) (2nd ed.). John Wiley & Sons.