**Jesus Yanes**

**January 29, 2015**

**NSS 1501**

**SDL Comparison**

**Chaos Model**

The Chaos Model is designed to fix the most important issues first. This set up, inspired by Go strategy, works in a similar way to how programmers would work to fix bugs and implement features towards the end of development. It allows the user to put issues in a list by priority and fix them individually.

It also allows the code to be viewed as an entire project or individual lines of code. This helps to verify smaller pieces of code and then move forward to the next set of lines. This model breaks code up as the whole project, systems, modules, functions, or lines of code.

**Security Development Lifecycle**

This method from Microsoft approaches things in a different way. In the design phase, security is addressed early to reduce vulnerabilities. Using threat modeling, scenarios are introduced to help identify these threats effectively, determine the magnitude of the risks caused by the threats, and create solutions. Any code or tools used that are deemed unsafe are addressed early on and corrected. Testing of attacks are used to determine if the proper security measures are in place. The software is certified and released, while an Incident Response Plan is created for users to report any vulnerabilities that were missed during development.

**Comparison**

These two models both have the security of the software and users in mind, but handle them in different ways. While the Chaos Model suggests prioritizing and fixing the major issues as they arise, the Security Development Lifecycle addresses them from the beginning. They will continue to have concerns about them for the entire development cycle and continues to do so after release.

Both have good ideas. I like the idea of prioritizing the issues and fixing the most important first. But, on the other hand, testing and preparing for vulnerabilities early on will reduce the amount of security issues that will occur later on.