**Module 2 Assignment - Software Development Life Cycle (SDLC)**

Ryan Rooney

SDEV120

Gabe Haynie

11/3/2024

The Software Development Life Cycle (SDLC) is a key component to any software developer who plans on creating a program for a company. This method allows us to systematically go through a set of rules that we must follow which in turn reduces the cost, improves the quality, and reduces production time for software development (Stackify Team, 2024). There are six key stages to the SDLC, the first being requirement analysis, planning, software design, software development, testing, and finally deployment. Each of these stages is completed in order.

Requirement analysis is the first stage, and this step usually involves customer input as well as any other input from stockholders or other analysis. Documentation known as SRS (Software Requirement Specification) is used to specify what the client wants to achieve (GeeksForGeeks, 2024). After the problem has been addressed, planning is put into action. This stage is usually considered to be the most important stage in the entire SDLC as careful planning can reduce bugs and security issues further down the line. This is also where we can determine how feasible the software is and if it’s worth pursuing with the resources required. The next stage is software design, and this is where the SRS is useful when developers design the project’s architecture. Multiple designs are created and put together in a document known as Design Document Specification (DDS) (GeeksForGeeks, 2024). The DDS is then shown to the client to check if it fulfills the requirements. Once the design has been chosen, development of the project occurs. During this stage, developers choose a variety of software tools based on multiple factors. Some of these include what programming languages to use, and any other programming tools such as compilers and interpreters (GeeksForGeeks, 2024). Once the product has been developed, testing occurs to see how well the program executes. This is also the time to spot any bugs or security issues that may show up. If the program does not fulfill the requirements of the SRS or doesn’t work as intended, it is then fixed and retested. Finally, after the program is accepted, deployment of the program occurs so users may start to use the product. If stakeholders are happy with the product after being used in an industrial environment, the program can either be slowly released or released in full depending on what the company wants. Maintenance can be done to the program until the end of its usefulness to the company.

There are multiple models that go off the SDLC concept. Each model has its own usefulness and place. Five of the most popular SDLC models include the waterfall, agile, iterative, spiral, and V-shaped model (GeeksForGeeks, 2024). The model I’d like to discuss is the iterative method. This method is a take on the classic waterfall method which has fallen out of favor. Instead of going through each stage of the SDLC and waiting until the end of the project for feedback, the iterative method provides feedback at every stage. This combines the ease of the waterfall method for developers with feedback at every stage from the client.

Prototyping is a method used during the SDLC when stakeholders need a more visual representation of what the program needs to do. This is done usually at the beginning of the SDLC and includes prototypes to give visual representation. Developers create a trial version of the product which then shows how the program could look.

Something that’s important to note when dealing with the SDLC and programming in general is modularization. Modularization allows us to critically think about a complex problem each stage individually. This is incredibly useful as breaking up tasks allows more than one person to work efficiently at each stage. Another benefit would be simply being less overwhelmed by such a large problem. Taking each smaller problem of a larger problem and solving it is much more manageable in the long run. This is especially important as having a cohesive program design is necessary for programmers. When each stage of the program is working in tandem due to good programming habits, better code is produced.

In conclusion, the SDLC is an invaluable methodology used by developers for any program. The benefits to SDLC include reduced cost, improved quality, and faster times to completing programs. There are multiple SDLC methods to choose from each with their own benefit. Finally, program design can benefit from modularization as having cohesive programs is essential for developers.

**Citations:**

GeeksForGeeks. (2024, September 3). *Software Development Life Cycle (SDLC).* <https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/>

GeeksForGeeks. (2024, September 3). *Software Development Life Cycle (SDLC).* <https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/>

Stackify Team. (2024, July 10). *What Is SDLC? Understand the Software Development Life Cycle.* <https://stackify.com/what-is-sdlc/>