**UAGC CST-499 Week 3 – Discussion Forum 1 – Vicki Kelm**

**Describe some of the best software development practices - Explain how these practices can positively or negatively impact the quality of the software being developed**

“The ultimate goal of most software engineering projects is to produce a working program” (Tsui, Karam, & Bernal, sect. 9.1).

One way to improve code style without altering its behavior is to use **refactoring**. This tool provides a powerful technique in the production of good code. This method applies a change to the internal structure of software which makes it easier to understand and cheaper to modify without changing its observable behavior.

Another important best practice for software development is the recognized value of **code reuse**. “One of the best practices is to reuse as much existing high-quality code as possible” (Tsui, Karam, & Bernal, sect. 9.3.3). This method offers saving time and resource by improving productivity and quality. Experienced programmers utilize this practice by keeping a library of personal reusable code patterns.

**Debugging** is also important in the implementation process to locate and fix errors in code. These errors are typically discovered during testing but could also be found through code inspections and standard use of the program. There are four phases of debugging:

1. **Stabilization, or reproduction**: Purpose is to be able to reproduce the error on a particular configuration, and to find out the conditions that led to the error by constructing a minimal test case (Tsui, Karam, & Bernal, sect. 9.3.1).
2. **Localization**: Process involves finding sections of the code that led to the error.
3. **Correction**: Process involves changing the code to fix the errors.
4. **Verification**: Process involves making sure the error is fixed with no other errors being introduced with the changes in the code.

A useful technique is the utilization of **assertions** which relates to the concepts of preconditions and postconditions. Based on our text, “It is a good practice to make your preconditions explicit by the use of assertions”. Assertions are statements that check a condition and produce an error if the condition is not met. To catch many errors, the assertions should be explicit and executable.

**Performance optimization** is also important to almost any program and consists of modifying a software system to make it work more efficiently and execute more rapidly (Odhiambo, 2018). The optimization of performance can affect maintainability and readability. A cost-benefit analysis should be conducted prior to performance optimization as a programmer’s cost, the decrease in maintainability and the possibility of introduction errors need to be weighed (Tsui, Karam, & Bernal, sect. 9.3.3).

**References**

Odhiambo, D. (2018, September 24). *Performance Optimization in Software Development*. <https://medium.com/the-andela-way/performance-optimization-in-software-development-ae7952ab885e>

Tsui, F., Karam, O., & Bernal, B. (2018). *Essentials of Software Engineering* (4th ed.). Jones & Bartlett Learning. <https://platform.virdocs.com/read/2348054/11/#/4/2[ch06]/2/2,/3:0,/3:0>