Module Two-Journal Two

Tammy Hartline

05/11/2023

CS-320 SNHU

**Dynamic and Static Testing**

* **What is static testing?**

Static testing is a type of software development testing that is best done before any actual programming code is written. It is used to detect and correct defects more efficiently and before execution testing. One of the major beneficial features of static testing is the ability to prevent design and/or coding defects, through identification of inconsistencies, redundancy, missing or contradicting information, and inaccuracies. It can also increase development productivity by decreasing development time and costs, as well as decreasing dynamic testing costs and time. Another feature static testing offers is a reduction in the total cost of quality for the product’s lifetime, from less failures occurring after the product is released into production. (A.K.A. Software maintenance costs could be reduced due to the product being designed to have more automated maintenance for example.) Lastly, it can also improve communication among team members due to the necessary participation in the review process.

* **What is dynamic testing?**

Dynamic testing requires testing on the actual product, post design and program writing. This is because dynamic testing is done by executing code, such as the hardware or final software. Something in the reading worth stating is that building a dynamic test that finds defects is often hard, if not impossible to accomplish.

* **What are the differences between static and dynamic testing?**

There are several differences when comparing static to dynamic testing. But, the most significant difference is that static testing is done against work products, without execution of the code, whereas dynamic testing requires access to the code and the ability to execute the program. When reading about static testing, I think my mind almost immediately went to pseudocode, but at a higher level. Where each stage of the program is gone over in tedious detail and described, documented, and modeled. When thinking of dynamic testing, my thoughts go more toward verifying that the final product performs and does what it is expected of the product, based on its underlying code development. The main benefit that dynamic testing offers, is that it can be used to ensure the software meets specified functional and non-functional requirements and can then be used to detect defects that are typically undiscoverable until execution and runtime.

* **Why is it important to use both static and dynamic testing?**

It is important to use both static and dynamic testing to ensure that development time is spent as productively as possible. Developing a product without static testing can be compared to creating a school project based solely on the title of the assignment. Meaning, without context, research, planning, modeling, and ensuring that you have a clear understanding of the task at hand and all the requirements you are expected to meet, you may still end up with something that is functional, but doing the *bare minimum* for anything is never a goal one should strive for. Once static testing is in its final stages, it can then be used with dynamic testing to improve the efficiency, quality, and lifetime durability, while also reducing development costs by preventing costly errors, which will ultimately produce a higher quality product and increase your client’s satisfaction.

**References/Citations:**

Brian Hambling (editor), Peter Morgan, Angelina Samaroo, Geoff Thompson, & Peter Williams,

*Software Testing, An ISTQB-BCS Certified Tester Foundation Guide, Fourth Edition,* Chapters 3 & 4