|  |
| --- |
| Johns Hopkins |
| Module 13 Assessment |
| Development versus O&M |
|  |
| **Brian Loughran** |
| **8/14/2020** |

|  |
| --- |
|  |

**Problem Statement**

Review the development and O&M phases of the life cycle.  How are they the same? How are they different?

**Assumptions**

N/A

**Computations**

N/A

**Discussion/Conclusions**

The development phases of the project life cycle are the phases that everyone thinks about when starting a software project. The development life cycle starts with defining the project, which involves determining key stakeholders, what the project should look like as a final product, and how the interface should look. After definition comes the planning phase, which involves scheduling, staffing, and estimating time to completion. This is followed by designing and building, which consists of coding, creating interfaces, building functionality and integrating systems to create a product. After designing and building comes a testing phase, which involves bug tracking, software repairs, and security reviews. Following testing is the review phase which can include peer reviews, tech reviews, and design reviews to ensure the product is ready for market. Finally, the deployment stage is actually pushing the product up to be consumed in the market (whether that be internal or external facing). All of these phases are critical to developing a software product and are the basics components of the development phases of the software life cycle.

In contrast stand the phases of O&M, which stands for operations and maintenance. O&M is a less ordered process, allowing operations to happen in almost any order. One of the actions that can be taken during O&M include performing code updates. Code updates can start either with a software discrepancy report or an engineering change proposal, which feed into coordinating resources and performing the change, followed by either a peer review or design inspection approval, which after testing result in a code update. Another action that can occur during O&M include insertion of new hardware, insertion of new COTS, developing new software versions, and contributions to I&T activities.

There are great deals of similarities between the development and O&M phases of the software life cycle. Many of the actions that are taken in the development phase are also taken in the O&M code update phase, with action definition, planning, design and building, testing and review happening just like in the development phase. The hardware identification process is very similar between the O&M phase and the design phase, with both trying their best to deliver the best hardware system possible for the business. Similarly, COTS platforms are constantly evaluated in both the development and O&M phases to provide the best solution to the business needs.

However, while there are many similarities between the two phases, there are also differences. For example, while many of the same activities are performed between the development and O&M phases, typically the cycle from definition to planning to design and build to testing to review to deployment is much faster for the O&M phases of the lifecycle than the development phases, simply because fixing a bug in a system is typically easier than creating the whole system. Another difference is in reviewing hardware and COTS for the product. During development, all hardware and COTS are on equal footing and choosing one over the other can be a simple decision about which is better. However, during the O&M phase, any proposed new hardware or COTS needs to be compared to what is currently being used, and the projected benefit of the new solution needs to outweigh the cost of transferring from the old solution. This introduces the possibility that a better solution may not be chosen because of the business cost of switching from the current solution. Another difference between development and O&M is that O&M is almost always performed on a stable system, and escapes project start-up politics.

Clearly both development and O&M are important parts of the project life cycle. Customers want products that work both on release and for the long haul, so providing an excellent product to customers through both cycles is paramount. While there are many differences between development and O&M, many of the differences are surprisingly minute, allowing a good software engineer to be able to thrive in both environments.