*Posted on December 18th, 2013*

**COMP 361 – Moving on to an Object-Oriented Approach**

There are two main approaches to system development: traditional and object-oriented. Regardless of the approach, the goal is to design a fully-functional and effective information system that serves the need of a client. However, the approach taken can have an effect on any number of development aspects such as budget or resources. The company in the example provided is planning on switching from a traditional approach to an object-oriented one. Changing course so drastically in the middle of development is a major challenge in itself, and will significantly increase the time and budget of the project because it would involve such a major redesign of the project. However, if we disregard this fact, we can see that there are some advantages and disadvantages of adopting the object-oriented approach versus a traditional approach.

The major advantage of an object-oriented approach is that it is highly adaptive. In the traditional approach, each phase of system development is independent of the others. In a paper by Mohammed and Munassar, they state, "With the traditional approach, will be expected to move forward gracefully from one phase to the other[1]." This means it phases can not be returned to and modified as project needs change. On the other hand, they mention that the object-oriented approach "tend[s] not to be too prescriptive[1]." This means developers are free to return to the different phases of project planning to make changes. This leads the object-oriented approach to be much more adaptive, especially when projects involve new technology or all the project needs are not determined at the project outset. However, in an article by Donald Burleson he mentions, "many bread-and butter information systems... may not benefit from the object oriented approach [2]." For well established systems with well defined technology in place, there may be little or no gains and switching to a new approach will be a detriment.

The object-oriented approach has other drawbacks as well. One of the biggest drawbacks to adopting an object-oriented approach is the lack of training in use and design. According to Burleson, many programmers are not skilled in object-oriented languages and techniques, even though object-oriented programming languages have been on the market for years [2]. He states that "object-oriented languages comprise less than 1% of systems today [2]." This means that making the switch to this approach will either require a significant amount of training, or hiring of new personnel, both of which cost a lot more money.

If I were to make a recommendation to the company mentioned in the Unit 2 Discussion, I would hesitate in wholeheartedly endorsing the change to an object oriented approach. I would assume the company already has well-established procedures and methods for development. Therefore, making a change will involve a significant amount of investment. However, it would seem that the adaptive nature of the object-oriented approach would allow the company to be more dynamic. They can work on new systems and integrate new technology more easily. For the long-term health and profit of the company, it would seem that making the change to this approach would be in their best interests.

**References**

[1] Nabil Mohammed and Ali Munassar, "Comparison between Traditional Approach and Object-Oriented Approach in Software Engineering Development," *International Journal of Advanced Computer Science and Applications 2, no. 6, (2011): accessed December 7, 2013,* [http://thesai.org/Downloads/Volume2No6/Paper%2010-Comparison%20between%20Traditional%20Approach%20and%20Object-Oriented%20Approach%20in%20Software%20Engineering%20Development.pdf](http://thesai.org/Downloads/Volume2No6/Paper 10-Comparison between Traditional Approach and Object-Oriented Approach in Software EngineeringDevelopment.pdf).

[2] Donald Burleson, "Advantages and Disadvantages of Object-Oriented Approach," *Oracle Tips by Burleson Consulting,* accessed December 7, 2013, <http://www.dba-oracle.com/t_object_oriented_approach.htm>.