Rodolfo Diaz

Prof. Wu

CSE 455: Software Engineering

Lab: Wednesday, 7:30 – 8:50 pm

Exercise 1.2:

The most important difference between generic software product development and custom software development lies within the ownership of the software specifications. In the case of generic software products, the ownership of the product specifications lies with the client/customer. On the other hand, in a custom software development, the ownership of the system specification is the developer themselves.

A complication that this can bring with generic software products is the client/developer specification agreement. If the client wishes to alter/add any specification, negotiations must be made between the client and developers. In the case of custom software development, the developer themselves can freely choose to alter/add/remove any feature they wish without negotiating about the software specifications.

Exercise 2.4

The usage of incremental development within a business software system is crucial because business software systems rely on a well-planned process. When altering/adding/removing features from said system, the process must be carefully thought out because these system are often complex. The act of software evolution is much more frequent, as opposed to Real-time systems because business systems must change with demands from their customers or competitors.

Real-time systems cannot use an incremental development approach because it wouldn’t be as stringent to what is needed. Real-time systems cannot be easily altered because they these types of systems depend on various hardware components. Furthermore, real-time systems are usually built upon a well-planned process, something that incremental development can deliver on.

Exercise 3.4

Some primary advantages within Extreme programming are the usage of stories. These advantage allow developers to easily see and understand stories, which can lead to constructive feedback among development teams. Stories can also present an incremental approach to extreme programming, opening the doors to a systematic way of implementing functionalities so that users/clients can understand.

However, stories consequently present problems that may be unavoidable. One problem can be that the representation of stories can lead to a complex architectural design due to possible vagueness within the stories. Also, stories can focus more on the functional requirements instead of the non-functional. This issues can bring problems with the end user/client. Lastly, stories can sometimes present an unfinished or incomplete understanding of the requirements. Perhaps an issue might arise within the story, but vagueness can make it difficult for developmental teams to find and fix.