Joseph Silva Jr.

01/10/2021

SNHU

IT 315

Module 1-2 Activity

**Use-Case Driven:**

A User-Case Driven is used to affect the behavior of the system by obtaining user cases as the primary modeling tool. The use cases allow the programmers, who are writing the code for the system, to identify the requirements by having the system communicate with the programmers. The use cases describe to the programmers how users interact with the system in order to perform an activity such as search for information. The Use-Case Driven is used to obtain data in order to make the proper first steps to help the programmers develop a system for the users’ requirements.

**Architecture-Centric:**

An Architecture-Centric is used for analysis and design of the project by creating the specifications, construction, and documents for the system. The Architecture-Centric breaks these jobs down into three different models: Functional, Structural, and Behavioral. The Architecture-Centric helps to describe the behavior of the system whether it be messages, user’s perception, classes, relationships, attributes, or methods.

**Iterative and Incremental:**

An Iterative and Incremental has the software project go through many cycles or increments. These cycles allow the software project to be continuous tested and refined throughout the life of the project. The system analysts build up the three architectural views in order to understand the user’s problem. The analyst uses the functional, structure, and behavior models in order to evolve the system to discover the user’s requirement.

**Models:**

The three models that an Architecture-Centric process produces are Functional (External) Model, Structure (Static) Model, and Behavioral (Dynamic) Model.

**Description:**

* Functional Model – Describes the behavior of the system from the perspective of the user.
* Structure Model – Describes the system in terms of attributes, methods, classes, and relationships.
* Behavioral Model – Describes the behavior of the system in terms of messages passed among objects and state changes within an object.

**Diagram:**

