What is a Use Case Anyway?
When do we use one?

Use Cases are a funny thing. They are used by many different people for many different things. They can be used to document business processes, functional requirements, software specifications, high level views, low level views or whatever we chose to use them for.

First of all Use Cases are not UML and UML is not Use Cases. When people think of Use Cases they sometimes think of the graphical Use Case model that can be developed using the UML artifacts. Yes, UML can be used to represent a Use Case model, and the behavior diagrams (activity, sequence and state) can be used to represent the Use Case scenarios. These UML diagrams are not the only way to represent Use Cases. Use Cases can also be represented using simple plain narrative language. You can use unstructured paragraphs to create a freeform Use Case or sequentially numbered steps to create a more structured Use Case. Use simple plain language to create a narrative Use Case.

More generally, a Use Case is a story about the interactions between two entities. These entities are referred to as Actors. An actor can either be a person, group of people, a role or even another system. One actor initiates a request and another responds. For example a customer (actor) could *place order* with another actor such as a clerk or an electronic system. The collection of all the Use Cases are all the functions for that system.

A Use Case can be represented by a series of steps known as a **scenario**. A scenario lists all the steps required to satisfy that request for the actor. Additional steps can be listed for alternative scenarios which are needed to respond to errors or alternative choices. For example, an error condition for the *place order* Use Case might be **product out of stock.**

Use Cases are said to be used to document requirements. This statement is somewhat controversial because of the difference of opinions of what a requirement is or even the different types of requirements (e.g business, functional, non-functional, etc.).

My opinion is that a business requirement should be documented as a required capability and stated in the purest sense, without any indication of any interaction or behavior. An example would be: **The customer must be able to place an order during normal business hours**. A Use Case could then identify and document the function required to satisfy this requirement (e.g. *place order*).

The steps in each Use Case would describe all the actions required to perform a specific function. An example of the *place order* Use Case is: 1. Customer searches catalog; 2. Customer calls store; 3. Customer places order. Since these types of statements are describing **how** the system should work, it may appear that we are designing the solution, which in fact I believe we are. However, the industry tends to classify these types of statements as Functional Requirements, which should be the next level requirements definition after the Business Requirements.

Use Cases were originally designed to document software specifications and, as such, tend to be able to describe how the solution would provide a specific function. Use Cases can also be used to document an existing business process or system. This exercise will aid the Business Analysts if they are being asked to create a new solution similar to the existing one.

In summary, Use Cases are used to tell a story about the interaction between two actors, usually a person and a system. Use Cases should be used to document the Functional Requirements of a system and not the Business requirements. Use a narrative step by step description to document a Use Case. Use UML Use Case Models, Activity, Sequence or State diagrams where necessary.