

# Business Rules

**The most independent and reusable code in the system**

- Rules & procedures to make / save the money
- Entity - an object or software module with high-level concepts
  - Having a set of *Critical Business Rules* operating on *Critical Business Data*
  - The interface of entity consists of the functions that implement the Critical Business Rules that operate on that data
- Use Case - application-specific business rules describe low-level concepts
  - Input, output, and the steps to produce output
  - How and when the critical business rules within the entities are invoked
- High-level entities know nothing of low-level use cases; instead use cases must know about the high-level entities

# Screaming Architecture

## Always the highlights of USE CASES

- The theme of an architecture
  - The architecture of a software application should scream about the use cases of the application
- The purpose of architecture
  - Good architectures help focus on the structures that support use cases without committing to frameworks, tools, and environments.
  - All decisions about frameworks, databases, web servers and other issues and tools can be deferred.
- Frameworks are tools, not ways of life
- Testable architectures
  - Test use cases without any of the frameworks in place