**Main Changes**

**Abstraction, Encapsulation, Separation of Concerns – introduced from scratch as they were not present in the initial solution**  
The main functions (retrieval of discount tiers, the discount calculation and receipt generation) have all been abstracted, encapsulated and injected where needed via Autofac and a “Utility Support System” consisting of a variety of objects, enumerations and extension methods. Certain properties and types on the Bike, Line and Order objects have also been refactored in order to properly serve the changes above.

**Business rules on issuance of discounts that more closely resemble the real world**

1. Introduction of discount coefficient tiers based on inventory age. Why:

a. Many flooring lines require you to pay of larger and larger portions based on the age of certain items. This hurts cash, and nothing that hurts cash is good.

b. Inventory may depreciate with time and the introduction of newer models. This causes retail value to suffer thus leading to lower sales volume and lower gross margins. Once again, old units need to move.

2. Discount coefficient tiers based on the size of the order where the most appropriate measure of such size is the orders gross pre-tax amount (i.e. “Volume Discounts”). Why: Volume based discounts are standard practice.

Basically, via 1 and 2 above, you are ensuring that (a) older units move faster and (b) your customers are incentivized to place large(r) orders.

For example:

Aging based discount tiers

a. For all units over 20 days in inventory, discount = 5%;

b. For all units over 50 days in inventory, discount = 10%;

c. For all units over 90 days in inventory, discount = 20%;

Volume based discount tiers

a. For all orders over $ 1000, discount = 5%;

b. For all orders over $ 5000, discount = 10%;

c. For all orders over $ 10,000, discount = 20%;

The retailer can introduce as many tiers as needed.

The order calculating and receipt generating system will now dynamically accommodate any sort of bike / pricing combinations. The discount tiers are made available via a IMockDiscountRepository object. Why – such tiered settings are “internal” in nature and can be / should be managed without having to introduce any new order processing logic / code (i.e. they can be entered via a “form” and persisted in an internal data store from where they can be retrieved and used by the order processing system).

**Unit testing – excluded at this time.**