Analyze if this requirement will affect any code that was introduced by the above requirement. Can the two types of requirements, rate policy and coin validation, be varied independent of each other?

• Reliability:
The pay station has been refactored with respect to Coin aspect for the above requirement, and new rate calculation can be handled only by "Rate Strategy interface" and adding more classes implementing the rate strategy.

• Run-time binding:
The binding between the pay station and its associate rate calculation can be changed at run-time.

• Separation of responsibilities:
Responsibilities are clearly separated and assigned to easily identifiable abstractions in the design. Here we have rate strategy interface and classes implementing it to support different types of rate calculation. We also have payment strategy interface and classes implementing it to support different coin types

• Separation of testing:
As the responsibilities have been separated I can actually test rate calculations and core pay station functionality independently.

• Variant selection is localized:
The code that decides what particular variant of rate calculation to use is in one spot only. The code that decides what particular variant of coins to use is in one spot only.

• Combinatorial:
We can introduce Rate calculation without interfering with the coin calculation variability.

Liabilities:
• Increased number of classes and interfaces.
• Clients must be aware of strategies. The selection of which rate policy and coins to use is no longer in the pay station but still someone has to make this decision. Variant selection is moved to the client objects.

This requirement will not affect any code designed for the previous requirement as these two requirements are designed using 2 different interfaces and classes implementing those interfaces.

The two requirements rate policy and coin validation can be varied independent of each other. This is because there is no direct interaction between "Rate Strategy and Payment Strategy interfaces". Implementation can be altered according to the requirement.