Non-Functional Requirements

Notes taken from:
Object Oriented Software Engineering text;
Art of Software Architecture (Stephen Albin)
Use Case Analysis (Bittner and Spence);
Use Cases – Requirements in Context (Kulak and Guiney)
Some notes from Rational Software Corporation slides
Other personal notes

Use-Case Analysis Steps – Here's where we are:

- Supplement the Use-Case Descriptions
- For each use-case realization
  - Find Classes from Use-Case Behavior
  - Distribute Use-Case Behavior to Classes
- For each resulting analysis class
  - Describe Responsibilities
  - Describe Attributes and Associations

→ Non-Functional Requirements – We have a:

- good understanding of the nature of the analysis classes and their responsibilities
- collaborations required to support the functionality described in the Use Cases via analysis classes

Need to address the non-functional requirements

IBM-Rational calls these "analysis mechanisms."

Non-Functional Requirements

- The **purpose** of "identifying non-functional requirements" is to get a handle on these absolutely necessary requirements that are normally not 'functional.'

- During Analysis, this information is speculative.
- Will be refined later.
- Philosophy: Capture now; Realize later…
- (Essential points in design…)