Metrics

1. Lines of code
   1. Includes only lines of executable code
   2. Monitored by Visual Studio
   3. To try to sniff out large classes and methods and short classes
2. Cyclomatic complexity
   1. The number of linearly independent paths through a program execution
   2. Monitored by Visual Studio
   3. Ensure that we have a reasonable number of test cases for our program
3. Class Coupling
   1. Number of classes coupled to other classes within the project
   2. Monitored by Visual Studio
   3. Ensure that our project uses a high cohesion, low coupling approach
4. Maintainability Index
   1. The maintainability index as calculated by Visual Studio
   2. Monitored by Visual Studio
   3. To ensure that we create easily maintainable code
5. Depth of Inheritence
   1. Measures the depth of a class or method from the class diagram
   2. Monitored by Visual Studio
   3. To determine if there is too much indirection.
6. % of tests passed
   1. The percentage of our total tests being passed at the end of each week
   2. Monitored by NUnit
   3. To ensure that we are meeting the goals that we set the previous week in writing tests
7. % of methods, classes, and tests with a description comment
   1. The percentage of classes, methods, and tests with a comment describing what they do
   2. Counted by hand
   3. To ensure good code documentation