Metrics to track:

- Number of Parameters per Method
    - Process for measuring : Use one of the many extensions for Visual Studio with this capability.
    - How we will use it : If our methods are taking too many parameters (>7), they may be doing too much and refactoring may be necessary.
    - Why we will track it : to make sure we don't have methods that need to be refactored into multiple methods.
- Cyclomatic Complexity for each method
    - Process for measuring : Use the built in code metrics in Visual Studio 2010
    - How we will use it : If the cyclomatic complexity is too high (>5), we should probably break our methods down into more basic methods that call eachother.
    - Why we will track it : A high cyclomatic complexity makes the code (usually unnecessarily) hard to understand. Simpler code is easier to debug.
- Lines of code per class (excluding comments and whitespace)
    - Process for measuring : Use the built in code metrics in Visual Studio 2010
    - How we will use it : Make sure each individual class isn't too long. A class with too many lines (>800) may indicate the need to split that class up.
    - Why we will track it : if a single class gets too big, we are likely not taking advantage of our object oriented capabilities.
- Lines of code per method (excluding comments and whitespace)
    - Process for measuring : Use the built in code metrics in Visual Studio 2010
    - How we will use it : If a method gets too big (>50 lines), we will pull out simple chunks of code into helper methods.
    - Why we will track it : if you throw too much logic into one method, it is very hard to debug and not well designed software. It is also much harder to test.
- Minimum amount of commenting on each method
    - Process for measuring : Use one of the many extensions for Visual Studio with this capability.
    - How we will use it : If we don't have at least a basic description of the method and its return value, then we will add it.
    - Why we will track it : Commenting makes code easy to read and helps others understand what is going on. This is good coding style to put in comments and is generally accepted as being useful.
- Number of characters per line
    - Process for measuring : Use one of the many extensions for Visual Studio with this capability.
    - How we will use it : If our number of characters per line gets higher than 80, we will break that line into 2 lines.
    - Why we will track it : our coding style guidelines specify <=80 characters per line for readability.
- Ratio of tests passed to tests attempted
    - Process for measuring: ratio is given when tests are run
    - How we will use it : if our percentage is below 90%, we need to pass more tests that we attempted
    - Why we will track it : If we don't pass tests, we don't have the functionality we set out to code.