*Agile Principles, Patterns, and Practices in C# by Robert C. Martin, Micah Martin*

Chapter 8: The Single Responsibility Principle (SRP)

"cohesion" (Demarco 1979, Page Jones 1980)

* The functional relatedness
* Of the elements
* Of a module

"reason to change" == "a responsibility" == "an axis of change"

Classes with multiple responsibilities are "coupling" those responsibilities.

* Changes to one responsibility
* Impair the ability to do the other responsibility.

Examples of "responsibilities" aka "axes of change"

* Draw a rectangle VS calculate the area of a rectangle
* Manage connections VS communicate over connections
* Business rules VS persistence operations

Why coupling responsibilities is bad:

* Dependent applications must include the code for all the responsibilities instead of only the code for the responsibility of interest.
* Making a change to one responsibility requires recompiling the code for all the responsibilities.

Adhering to single responsibility requires a balance between:

* Rigidity: having to recompile an application, when there are changes to parts of a dependency that the application does not use.
* Needless complexity: introduces fixes for problems that are unlikely to occur.

Principles:

> A class should have only one reason to change.

> An axis of change is only an axis of change if the change actually occurs.

In many ways, all the other OO principles are re-articulations of the SRP.