Solid Principles.

The Single Responsibility Principle   
A functional unit on a given level of abstraction should only be responsible for a single aspect of a system’s requirements. An aspect of requirements is a trait or property of requirements, which can change independently of other aspects.

The Open Closed Principle

The Liskov Substitution Principle

The Dependency Inversion Principle

The Interface Segregation Principle

Principles behind the Agile Manifesto

Our highest priority is to satisfy the customer

through early and continuous delivery

of valuable software.

Welcome changing requirements, even late in

development. Agile processes harness change for

the customer's competitive advantage.

Deliver working software frequently, from a

couple of weeks to a couple of months, with a

preference to the shorter timescale.

Business people and developers must work

together daily throughout the project.

Build projects around motivated individuals.

Give them the environment and support they need,

and trust them to get the job done.

The most efficient and effective method of

conveying information to and within a development

team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development.

The sponsors, developers, and users should be able

to maintain a constant pace indefinitely.

Continuous attention to technical excellence

and good design enhances agility.

Simplicity--the art of maximizing the amount

of work not done--is essential.

The best architectures, requirements, and designs

emerge from self-organizing teams.

At regular intervals, the team reflects on how

to become more effective, then tunes and adjusts

its behavior accordingly.