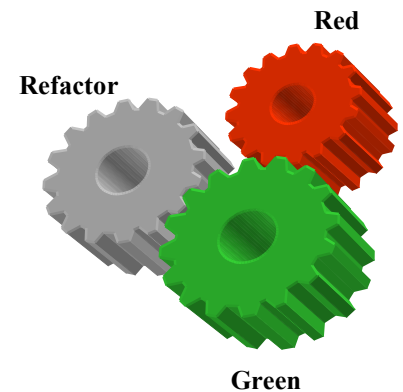


# Test Driven Development

Test Driven Development (TDD) is a rhythmic development practice that requires a high level of individual discipline and focuses on using automated tests as a lever to shape not only the quality but the design of the associated production code.

TDD is codified through a simple set of steps, often summarised as “red, green, refactor”. See the diagram below for more details on each step.



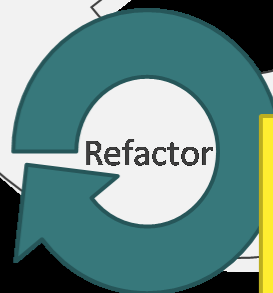
1. Write the test
  2. Compile the test
  3. Make the test compile
  4. Run the test
- *Tests shape the design*



5. Write the implementation
  6. Run the test
- *Write just enough code to make test pass*
  - *Focus on simplicity*



Cycle time of minutes not hours



7. Improve design quality
- *Not always needed*
  - *Rule of Threes*
  - *Behaviour preserving*



**REFERENCES:**

- For a more definitive description of Test Driven Development go to Wikipedia: [http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development).
- The seminal reference on TDD is Kent Beck's "Test Driven Development by Example". The examples are predominantly in Java, but the concepts are very transferrable to other languages.