

TDD cycle

Test-Driven Development cycle and why to use it

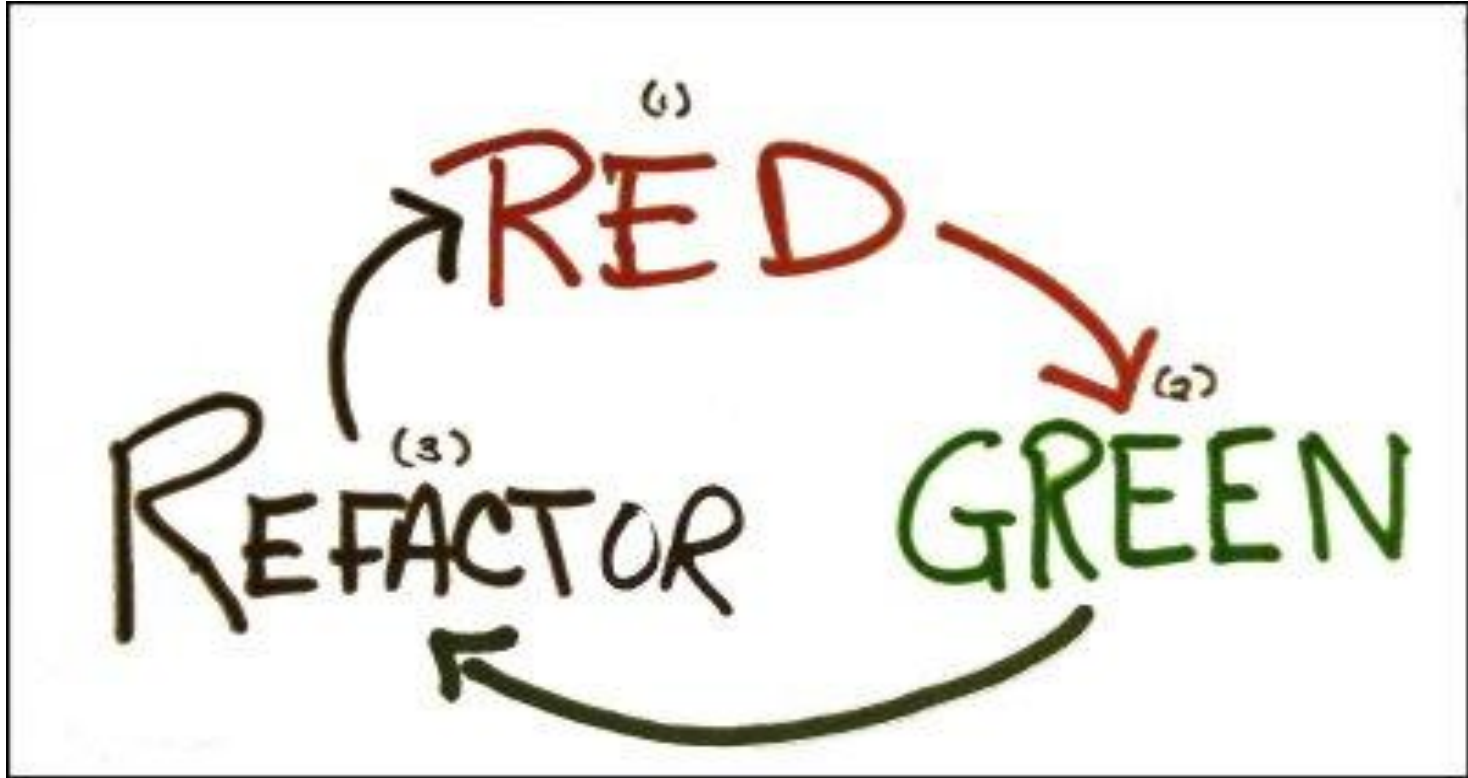
In the beginning God created...

- Test First Design (1999)
- **Around 2000** first **TDD**:
 - Part of extreme programming
 - Adopted by Scrum
 - All agile and nonAgile

Thou shalt not...

- 3 laws:
 - You are not allowed to write any production code until you have first written a failing unit test.
 - You are not allowed to write more of a unit test than is sufficient to fail - and not compiling is failing.
 - You are not allowed to write more production code than is sufficient to pass the currently failing unit test.

...there is no revelation...



...righteous will be rewarded with prosperity...

Benefits:

- Certainty
- Defect rate
- Courage
- Documentation
- Design

...righteous will be rewarded with prosperity...

Certainty:

- Change of code
- Passed tests => near certainty
 - Enough to ship!

...righteous will be rewarded with prosperity...

Defect rate:

- IBM, Microsoft
- Sabre, Symantec
- 2x-10x defect rate reduction

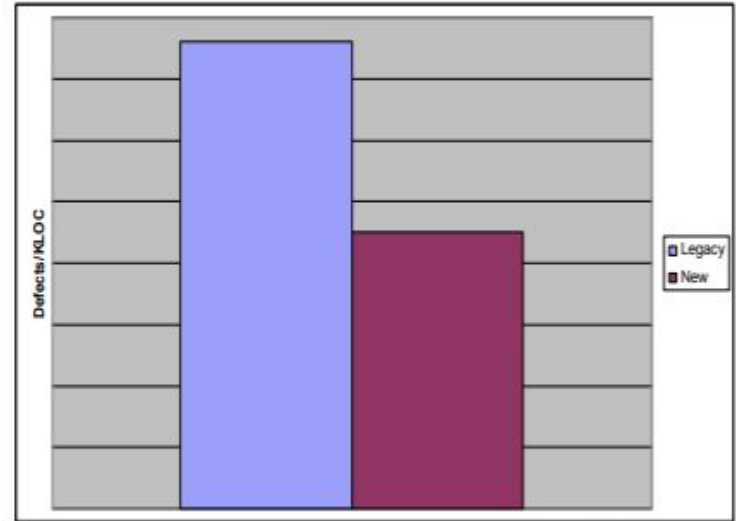


Figure 3: FVT/Regression Defect Density

...righteous will be rewarded with prosperity...

Courage:

- To change/improve code
- Refactor
- Can keep with clean code
- “Left the code in better shape”

...righteous will be rewarded with prosperity...

Documentation:

- Tests = Self documented
- Low-level documentation

...righteous will be rewarded with prosperity...

Design:

- Unit Tests ergo:

Simple, single responsible, isolated function

...no other Gods before me...

- Certainty
- Defect rate
- Courage
- Documentation
- Design

=> **Unprofessional** not to use it!

...there is no righteous man...

NOT a:

- Religion
- Magic formula
- Still can write a bad code
- Not always efficient

...they had no root, they withered away...

The Clean Code (Robert C. Martin)

The Clean Coder (Robert C. Martin)

<https://blog.cleancoder.com/uncle-bob/2014/12/17/TheCyclesOfTDD.html>

<https://dev.to/yurishenrique/5-test-driven-development-tips-from-the-clean-coder-35mn>

?

...