

# Test Driven Development (TDD)

Best Practices Workshop, March 25-26 2019, Hampton VA

Carlos Cruz  
Jules Kouatchou  
Brent Smith

NASA GSFC Code 606/610 (ASTG/GMAO)  
Greenbelt, Maryland 20771

# What is TDD?

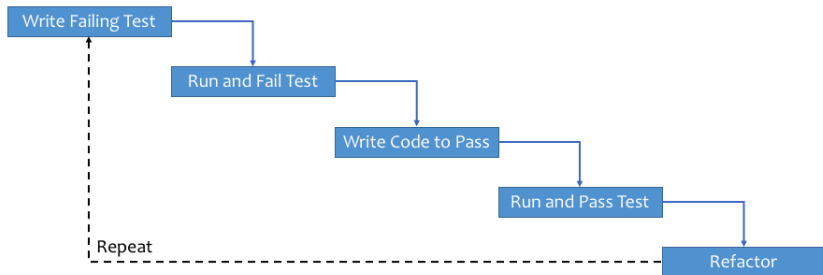
An **agile** methodology (a style of programming) where coding, testing, and design are tightly interwoven and in which unit **T**ests **D**rive the **D**evelopment of the production code.



# How does TDD work?

Have requirements handy.

Break up your task into functional units, then for each unit:



# Red Green Refactor



## Some useful unittest constructs

- `setUp()`: Let's you keep access any data that you want to keep around for all (or most of) the tests.
- `tearDown()`: The opposite of `setUp()`
- `fail()`: Force a test to fail. Useful in TDD when you're still in the process of writing your tests.
- various asserts:
  - `assertEquals()`
  - `assertIsNone()`
  - `assertListEquals()`
  - `assertDictEquals()`
  - `assertRaises()`
  - many more...



# Python Demo



## FAQ

- How can I write all my tests, when I don't even know all the implementation details?
- What makes a good unit test?
- How many tests should I have?
- How should I order my tests?
- How can I avoid code repetition within my tests?

