

# Test-Driven Development

Let the tests be your guide

**It starts  
with  
the test**

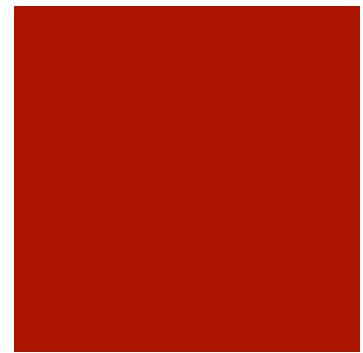


# Benefits of TDD

- All the benefits of unit testing plus ...
- It becomes hard to create errors in the first place
  - You almost have to try to create errors
- Better matches the business requirements
- It does what it is supposed to do
  - It must because we only write tests that satisfy requirements
- Encourages communication with the business
- It eliminates gold plating



Red,  
green,  
refactor



The red phase

# You want to write positive tests and negative tests

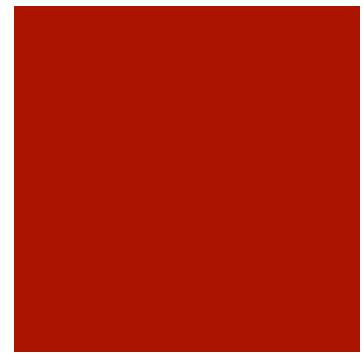
- Negative tests involve values that are outside acceptable ranges.
  - They should fail
  - You're testing to make sure that they do
- Positive tests are ones that should pass



The green  
phase



The refactoring  
phase



Once finished you pick up a user story and start again writing a new failing test



# Summary

- Unit testing is the cornerstone of TDD
- TDD = Red, green, refactor
- Red = write a failing unit test
- Green = make the unit test pass using a naïve implementation
- Refactor = improve the code quality
- TDD results in huge benefits like better quality code that better matches the business requirements, encourages communication and eliminates gold plating