# Unit Tests for Learning

Darren Hale

# Thank you to our Sponsors!













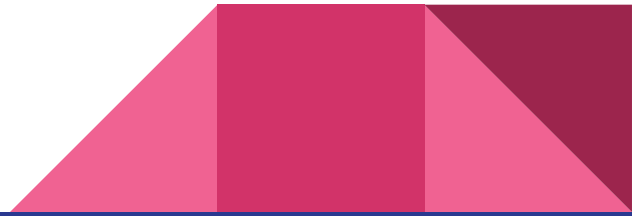
# About Me

My name is **Darren Hale**.

I create software.

I’m easy to find online:

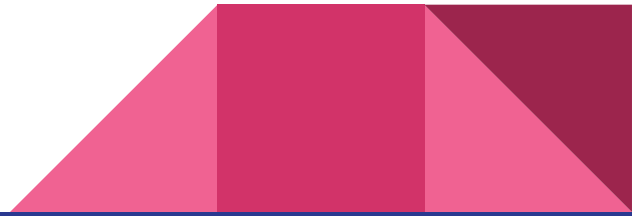
- LinkedIn:  https://linkedin.com/in/darrenhale
- GitHub:  https://github.com/haled
- Email:  darren.e.hale@gmail.com



# Why Unit Tests?

Majority Thinking…

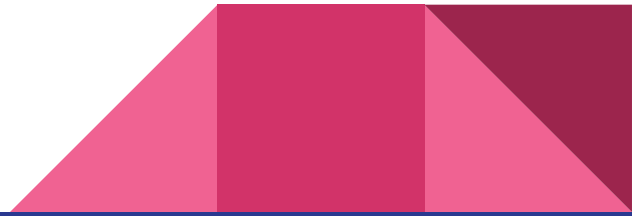
- Prove code does what it should
- Provide coverage for changes
- I’m “supposed” to



# Why Unit Tests, Really?

## IT’S A DESIGN TOOL

- Capture Requirements
- Discover Requirements
- Document Code
- Build Trust in the Code Base
- Create a safety net for changes
- Minimize costly debugging
- It creates better code
- They can help you learn new things



# What Makes a Good Unit Test

- Tests a single aspect of the code
- Asserts “what” more than “how”
- Does not require extensive set up
- Easy to read
- Easy to understand!
- Fits on a single screen

- A test can “talk” to you

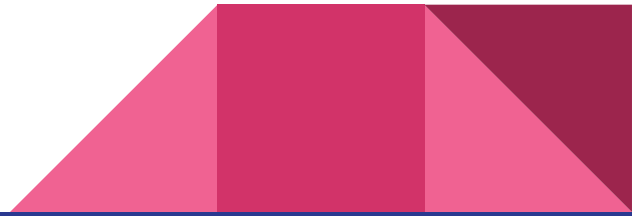
# How I Create Tests

I follow an agile approach:

- Don’t define every minute detail up front.
- Delay big decisions as long as responsibly possible.
- Allow questions to emerge.

# Assumptions

- Attempting to keep the exercise straightforward (simple) so we don’t get lost in code details.
- Trying to minimize third-party and external interactions.
- Interactive

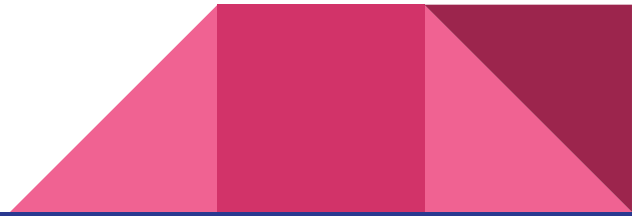


# Exercise

Create a command-line flashcard app that displays questions and prompts for answers from the user.

# Final Thoughts

- Martin Fowler has a lot of material on the testing pyramid to sort through unit tests, integration tests, and acceptance tests.
- Try to push external interactions as far to the edges as reasonable.
- Pair programming can help the tests guide design.



# About Me

My name is **Darren Hale**.

I create software.

I’m easy to find online:

- LinkedIn:  https://linkedin.com/in/darrenhale
- GitHub:  https://github.com/haled
- Email:  darren.e.hale@gmail.com