

Shokunin Sessions

Refactoring Legacy Code

Legacy Code

“To me, legacy code is simply code without tests”

Michael Feathers

What am I doing here?!

What will we do?!

Today's Agenda

- Pairing - 25 min, Ruby
- Discussion
- Pairing - 25 min, Java
- Discussion
- Pairing - 25 min, Javascript
- Discussion / Conclusion
- Retro

Ground Rules

Be Present

No sexting, no double screen pairing

Clear Your Mind

We're seeking to understand, not to be understood

Everyone is Equal

Even Adam

Failure is Encouraged

As long as you learn from it

Get into pairs!

Round 1

Ruby

Trivia Codebase: the obvious

- It's a trivia game
- Example has 3 players
- There is the concept of rolling dice
- There is the concept of player location (on a board?)
- There are different categories of question
- Users answer questions
- Correct answers earn players gold
- Wrong answers send players to the penalty box
- There is a way for players to get out of the penalty box
- It looks like the first user to reach 6 gold coins wins

Reflect

Golden Master

- Find the way the system delivers its outputs
- Find a way to capture the output of the system without changing the code
- Find a pattern for the outputs
- Generate enough random inputs and persist the input / output
- Write a system test to check the SUT against the previously persisted data
- Commit the test
- Check test behaviour and coverage
- If not enough behaviours are covered, go to step 3

Get into pairs!

Round 2

Java

branch: golden_master

Reflect

Seams

- A way of altering behaviour of code without touching the production code
- Types:
 - Preprocessor
 - Link
 - Object (dependency injection or subclass and override)
- In javascript you can replace object functions on the fly!

Get into pairs!

Round 3

Javascript

branch: seams

Reflect

Retro