Embedded TDD

For Cambridge Software Crafters
13 March 2024

Brice Fernandes
brice@fractallambda.com

Logistics and Wifi

🚽 Behind the main screen to the right of the corridor.

🛜 Wifi is **The Bradfield Centre** password is **Ca3Br1d5e**

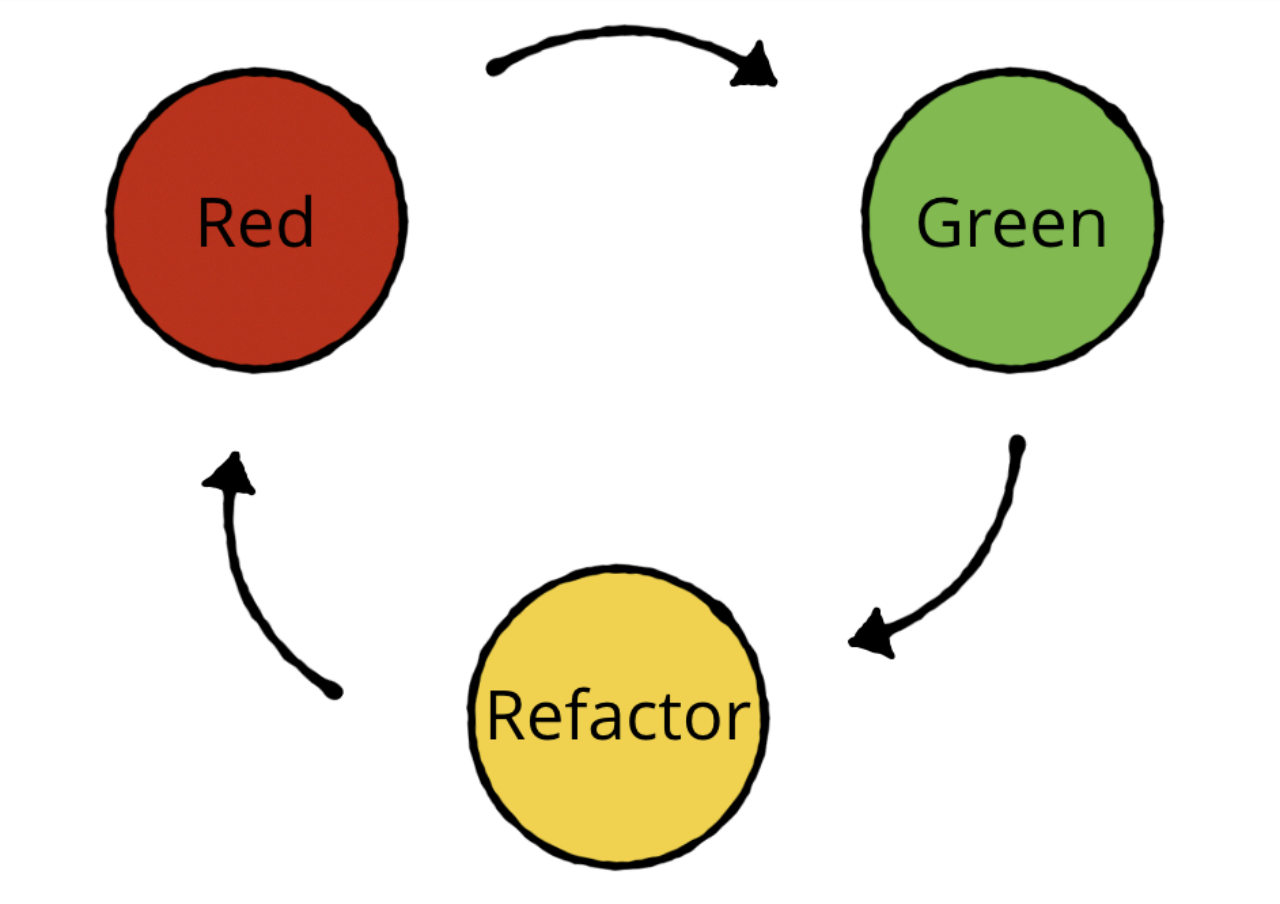
🚨 We do not expect alarms. Assume a fire alarm is real and make your way to the car park.

Plan for this evening

1. TDD Refresh
2. What we mean by "embedded"
3. Embedded craftsmanship practices
4. The Katas
    1. LED Driver Kata
    2. Interrupt Kata
5. Recap

Intro

Why this talk?



TDD Loop

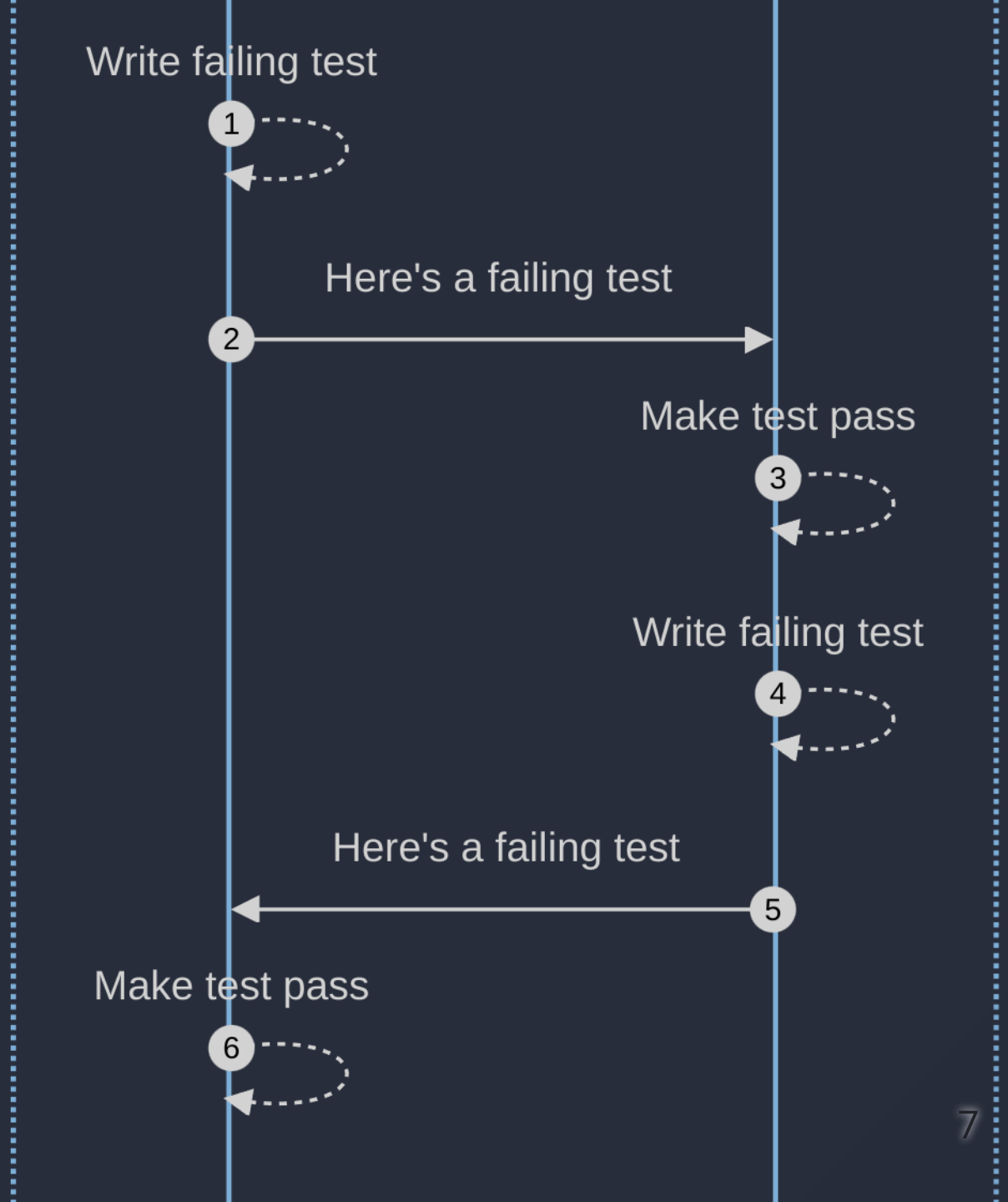
Write a failing test

Make the test pass

Refactor the code

```
TEST(LedDriver, ArrangeActAssertExample){

    // Arrange
    LedDriver_Create(&virtualLeds);

    // Act
    LedDriver_TurnOn(4);

    // Assert
    TEST_ASSERT_EQUAL_HEX16(0x08, virtualLeds);

    //Teardown
    LedDriver_Destroy();
}
```

Ping Pong TDD



What I mean by Embedded

Embedded constraints

- Resource constraints (RAM, CPU)
- Lack of standard libraries
- No or limited filesystem
- Limited Interface (serial? UART, SWI)
- No Operating System
- No standard library
- Direct hardware access
- Lack of MMU/PMMU

Special pains

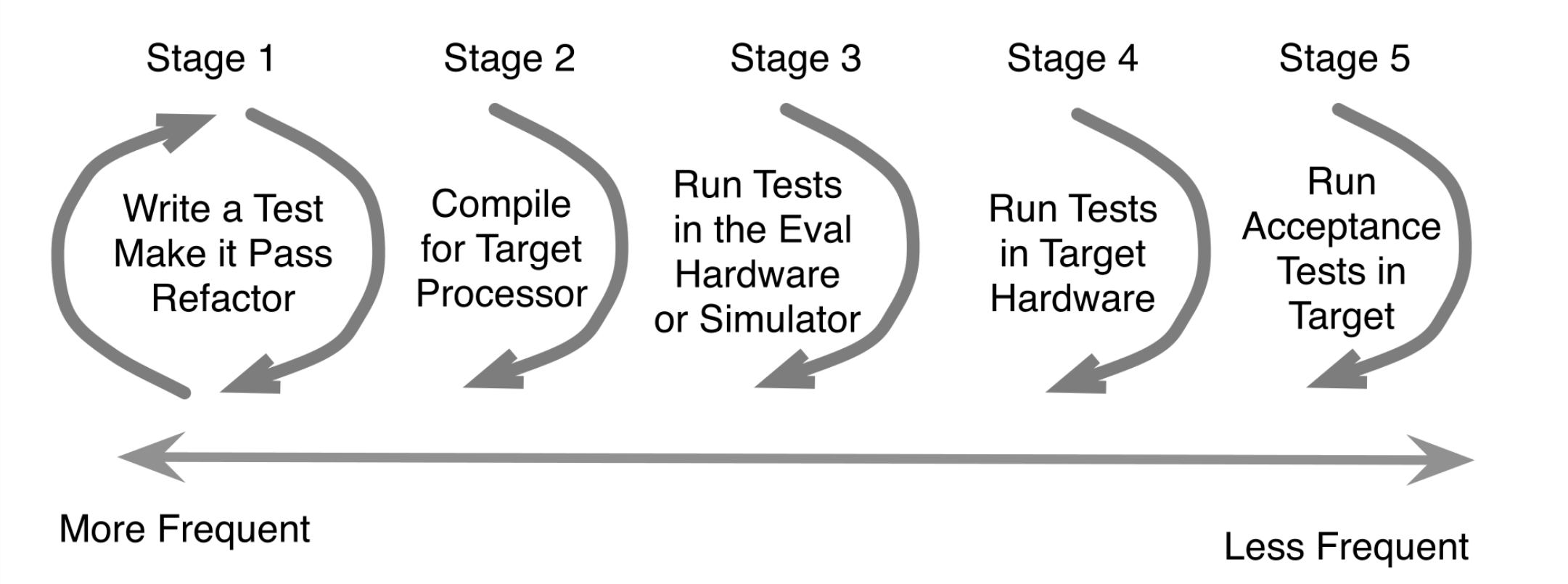
- Late hardware delivery
- Hardware scarcity
- Hardware bugs
- Long target compile times
- Long target setup and upload time
- Compiler licenses

Craftsmanship for Embedded

Dual targeting

- Dual targeting
  - Simulate hard-to-duplicate conditions
  - Target bottleneck
  - Running the test suite on the target

Embedded TDD Cycles



CI and automated HW tests

Advanced Mocking

Advanced Mocking

1. Mock the clock
2. Test doubles

1. Code structure & Link time substitution
2. Function pointer substitution
3. Syntactic substitution (preprocessor)

Simulators

SOLID

1. Single Responsibility Principle
2. Open Closed Principle
3. Liskov Substitution Principle
4. Interface Segregation Principle
5. Dependency Inversion Priciple

The Katas

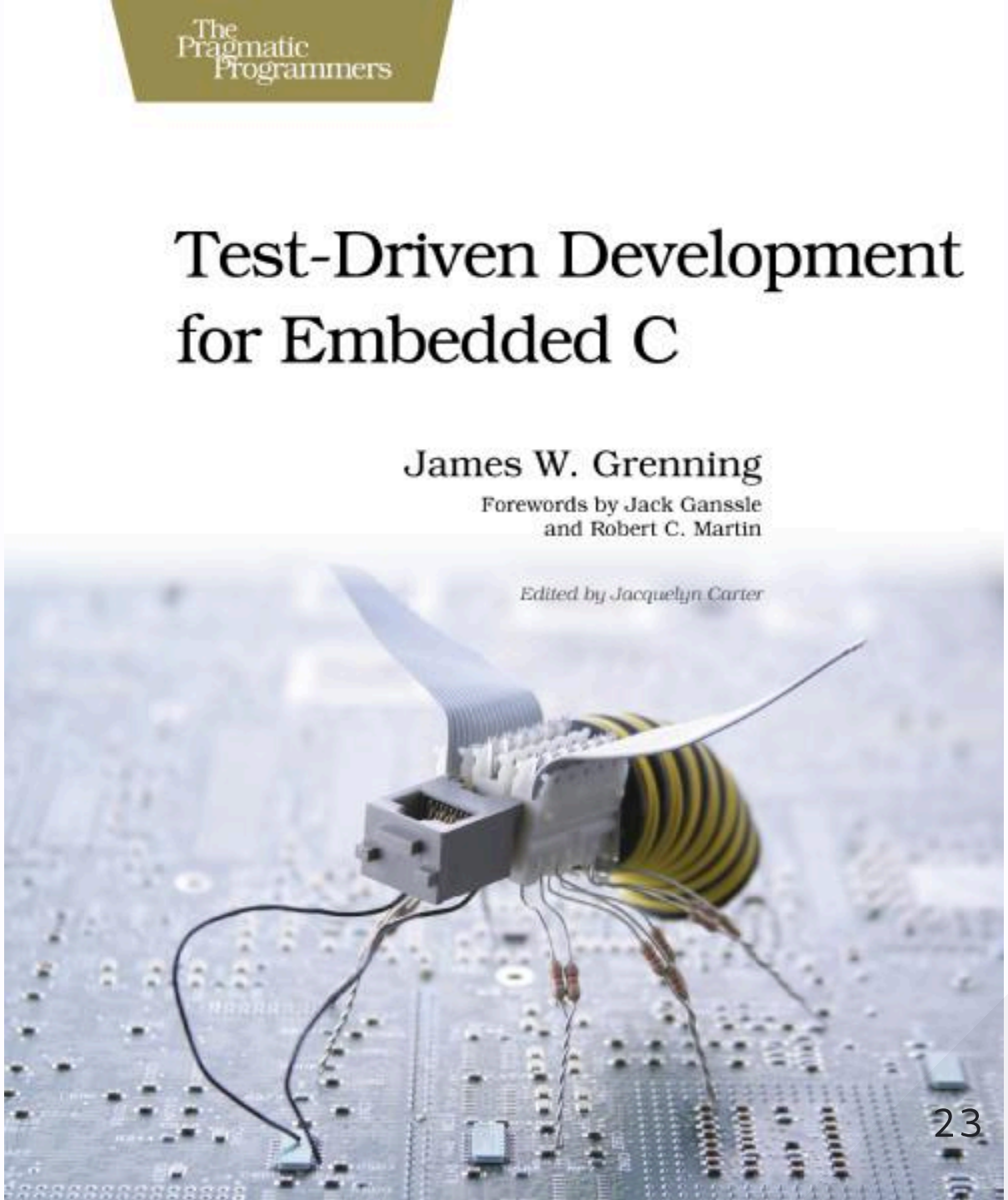
LED Driver Kata

Interrupt Kata

Recap

What we learnt

Further Reading



Thank you

🙏

Q&A

❓