Unit Testing for C++

BoF session facilitated by Honey Sukesan

Disclaimer

All the views, thoughts, and opinions I expressed in this session belong solely to the presenter, and not necessarily to the presenter's employer, organization, committee or other group or individual.

Good housekeeping

- Please keep your webcams on as much as possible.
- Mute yourself you are not speaking.
- Please don't talk over each other.
- Please maintain mutual respect.

It's self introduction time.

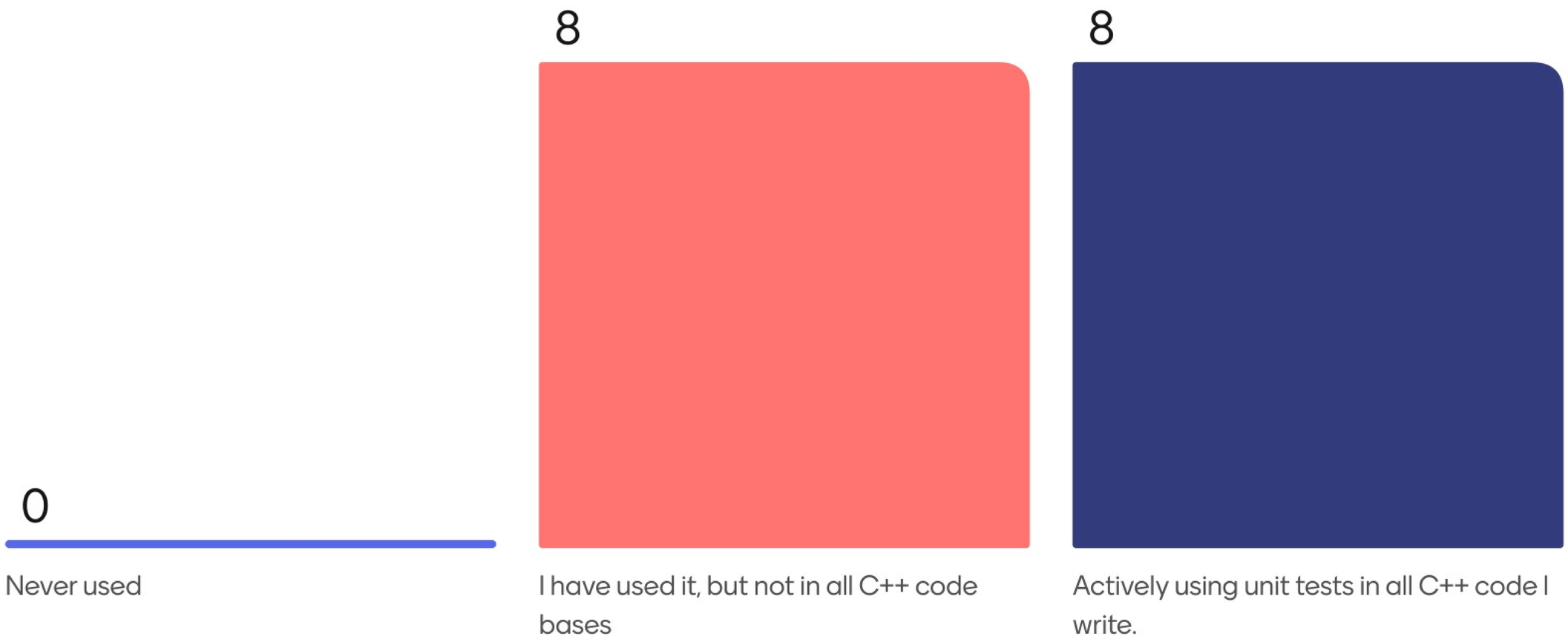
What is a BoF session?

- “Birds of a feather flock together"
- Informal discussions around a shared interest or a topic without any specific agenda.
- They are not meant to be presentations
- Everyone is welcome to share their thoughts.
- BoF proposer act as a facilitator for the discussions

Instructions

Mentimeter

Are you using unit testing in your C++ codebase?



1

16

Mentimeter

What is that one word coming to your mind when you think about unit tests?

28 responses

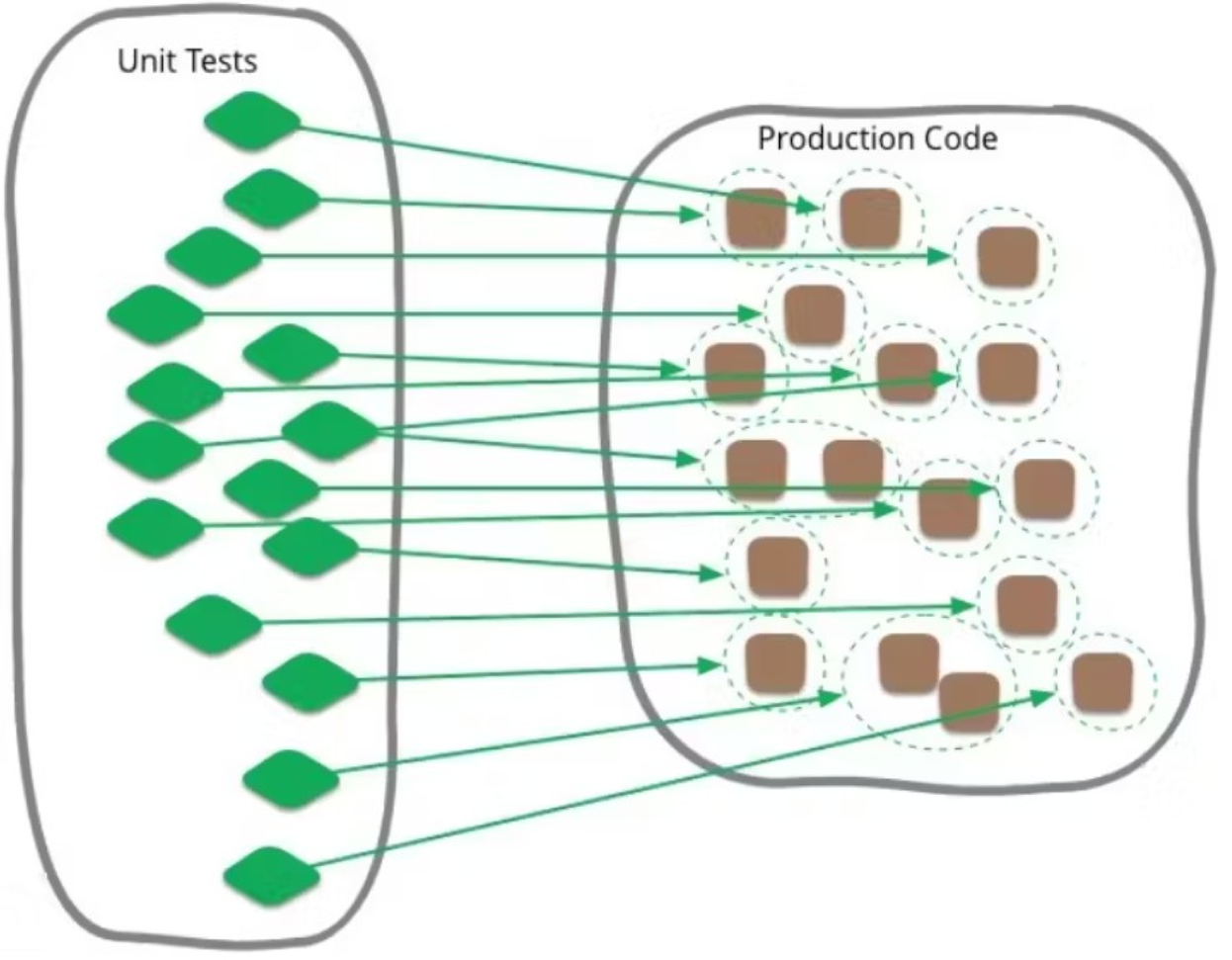
2 16

"I've said this before, and I'll say it again. Despite the fact that your company may have a separate QA group to test the software, it should be the goal of the development group that QA find nothing wrong."

– Robert C.Martin (Uncle Bob), The Clean Coder.

Unit testing

- Act of testing the correctness of your code at the smallest possible unit: the function.
- Unit tests are small, automated, stand alone executables that perform unit testing on your code.



```cpp
// rectangle.h
#ifndef RECTANGLE_H
#define RECTANGLE_H

class Rectangle {
public:
    Rectangle(double width, double height);
    double area() const;
    double perimeter() const;

private:
    double width_;
    double height_;
};

#endif // RECTANGLE_H

// rectangle.cpp
#include "rectangle.h"

Rectangle::Rectangle(double width, double height) : width_(width), height_(height) {}

double Rectangle::area() const {
    return width_ * height_;
}

double Rectangle::perimeter() const {
    return 2 * (width_ + height_);
}
```

Example rectangle class

```cpp
// rectangle_test.cpp
#include "rectangle.h"
#include <gtest/gtest.h>

TEST(RectangleTest, DefaultConstructor) {
    Rectangle rect(5.0, 10.0);

    EXPECT_DOUBLE_EQ(5.0, rect.area() / 10.0); // Check width
    EXPECT_DOUBLE_EQ(10.0, rect.area() / 5.0); // Check height
}

TEST(RectangleTest, AreaCalculation) {
    Rectangle rect(4.0, 6.0);
    EXPECT_DOUBLE_EQ(24.0, rect.area());
}

TEST(RectangleTest, PerimeterCalculation) {
    Rectangle rect(4.0, 6.0);
    EXPECT_DOUBLE_EQ(20.0, rect.perimeter());
}

TEST(RectangleTest, ZeroDimensions) {
    Rectangle rect(0.0, 0.0);
    EXPECT_DOUBLE_EQ(0.0, rect.area());
    EXPECT_DOUBLE_EQ(0.0, rect.perimeter());
}

int main(int argc, char **argv) {
    ::testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```

Example unit tests for class rectangle

Mentimeter

Which are the C++ unit test frameworks you are familiar with / are using?

catch2

catch, Cppunit

gtest / gmock 👑

GTest/GMock

catch2

catch2

ApprovalTests also 👑

boost testframework

15

Mentimeter

Which are the C++ unit test frameworks you are familiar with / are using?

Gtest

Catch2, gtest

BoostTest, GTest, GMock

catch2

catch2 gtest

Great, catch, mstest, boost test

BOOST GTest

Cantata

15

Mentimeter

Which are the C++ unit test frameworks you are familiar with / are using?

Python (for Mocking)

googleTest, catch2, doctest

Ut

Doctest

printf??

fmt

Code review 😀

Boost, gtest, faekit

15

Mentimeter

Which are the C++ unit test frameworks you are familiar with / are using?

static_assert

short, explicit

15

Conference talks

CppCon 2018: Phil Nash “Modern C++ Testing with Catch2”

Another conference talk

Kevlin Henney - Programming with GUTs - Meeting C++ 2021

Another one on good unit tests

Unit testing terminologies

- A test double is an object that can stand in for a real object in a test
- The most common types of test doubles are stubs, mocks, and fakes.
- A stub has no logic, and only returns what you tell it to return.
- It can be used an object is needed to return specific values in order to get your code under test into a certain state
- A mock has expectations about the way it should be called, and a test should fail if it's not called that way
- A fake doesn't use a mocking framework.
- It's a lightweight implementation of an API that behaves like the real implementation

Mentimeter

What do you think as the properties of good unit tests?

- Good naming
- short runtime
- reproducability
- It should work
- Small
- Stable
- fast
- The one that exists

11

Mentimeter

What do you think as the properties of good unit tests?

- Well defined responsibilities
- Cover all edge cases
- Documents use
- Test only one thing
- Runs fast
- requirements based
- Coverage?
- positive and negative testing

11

Mentimeter

What do you think as the properties of good unit tests?

- Deterministic and reproducible
- Test public api
- Easy to read
- 100% coverage
- Test public interface
- easy to read
- Same code quality as production code base
- Mocks real run

11

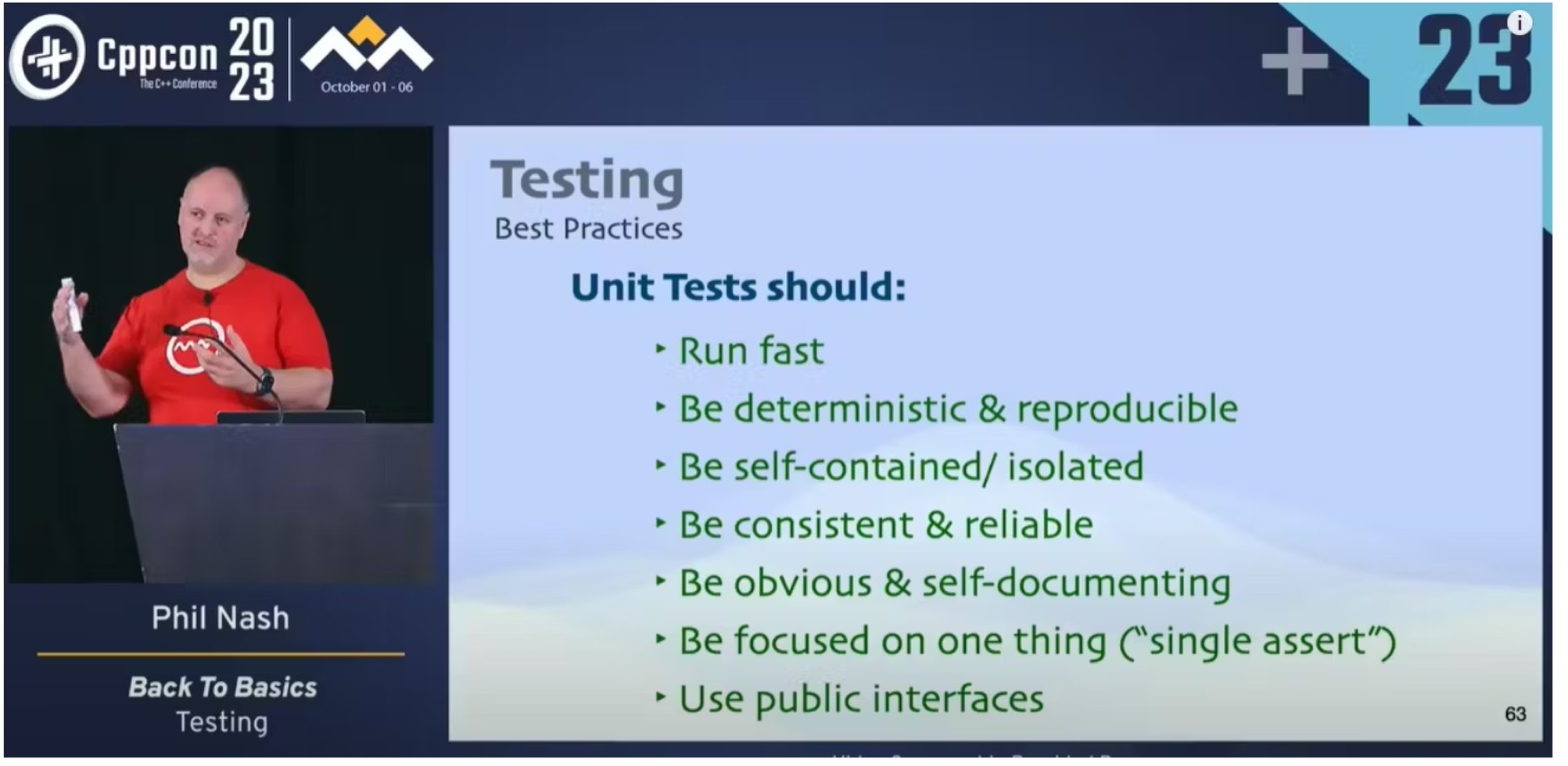
Mentimeter

What do you think as the properties of good unit tests?

Fails for good reasons

reasoning in TDD is quite difficult, solveable with practice

11



Unit tests best practices

What are the challenges you face in writing unit tests and how can it be solved?

Mentimeter

How do you handle complex dependencies in unit tests?

Mentimeter

What metrics do you use to measure the effectiveness of your unit tests?

Quick recap

Thank you for your time.