

Testing



Michael VanSickle

@vansimke



Introduction



Why write tests?

What should you test?

How does Go support testing?



Why Write Tests?



Initial Build

Ensure correctness of application



Production

Identify errors before users are impacted



Enhancements

Prevent new development from breaking older features



Why Write Tests?



Why Write Tests?



What Should You Test?

Correctness

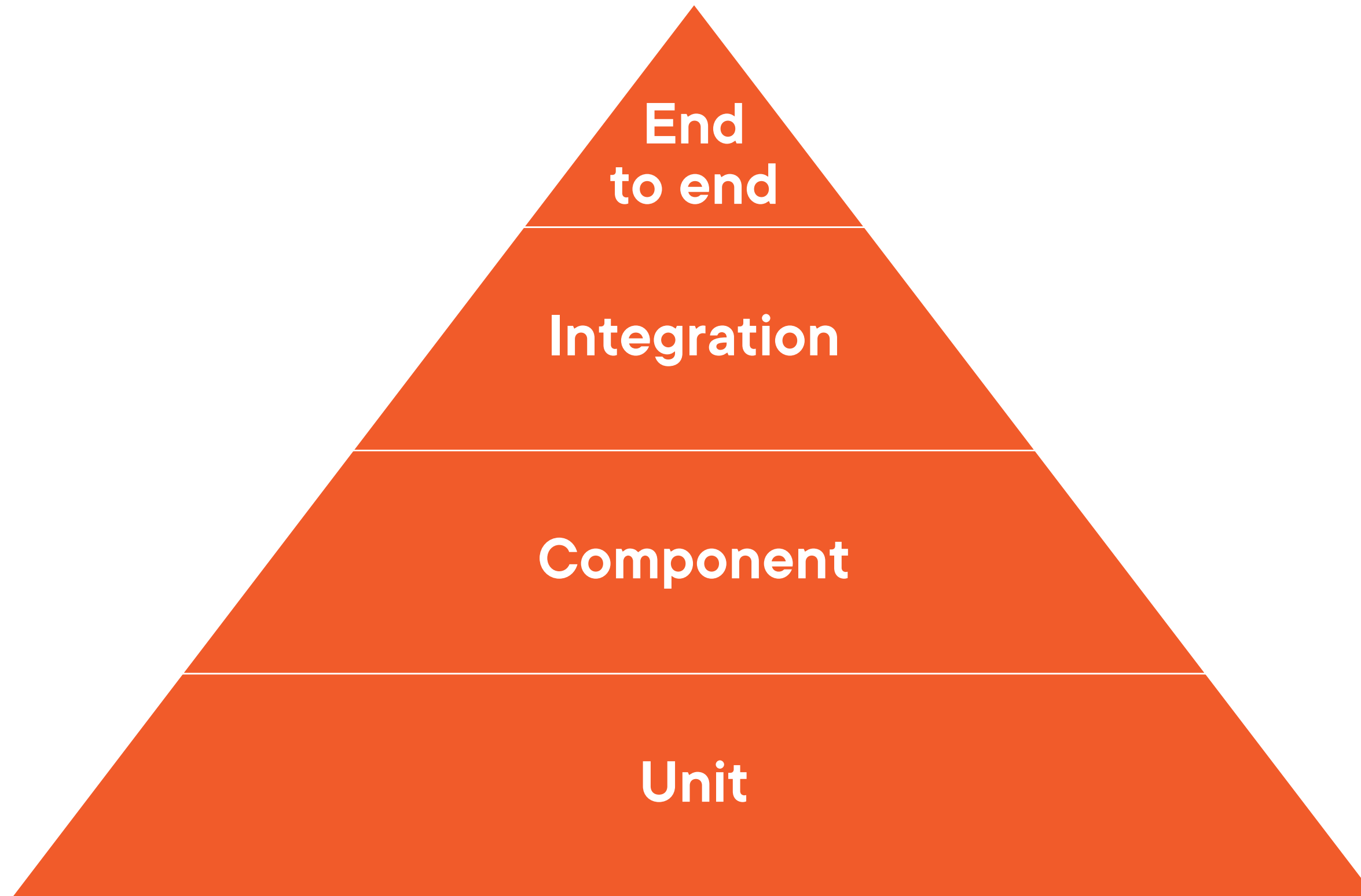
The program does what it should

Performance

**The program doesn't consume
excessive resources**



What Should You Test?



What Should You Test?

Prove that individual
units of logic are
correct

Unit

Confirm that different
application concerns
(package?) perform
correctly

Component

Validate that entire
program works as
expected

Integration

Demonstrate that
entire system works
correctly together

End to End



How Does Go Support Testing?

Correctness

Test

Fuzz

Example

<https://pkg.go.dev/testing>

Performance

Benchmark

Resource usage

Profiling

Tracing

<https://go.dev/doc/diagnostics>

Testing Go Applications



Demo



write a simple test



Summary



Why write tests?

What should you test?

How does Go support testing?



Course Plan

**Variables and data
types**

**Creating and
debugging**

Control flow

**Code organization
and error
management**

Concurrency

Testing

