

Testing Business Logic



Michael Van Sickle

@vansimke



Introduction



The Testing Pyramid

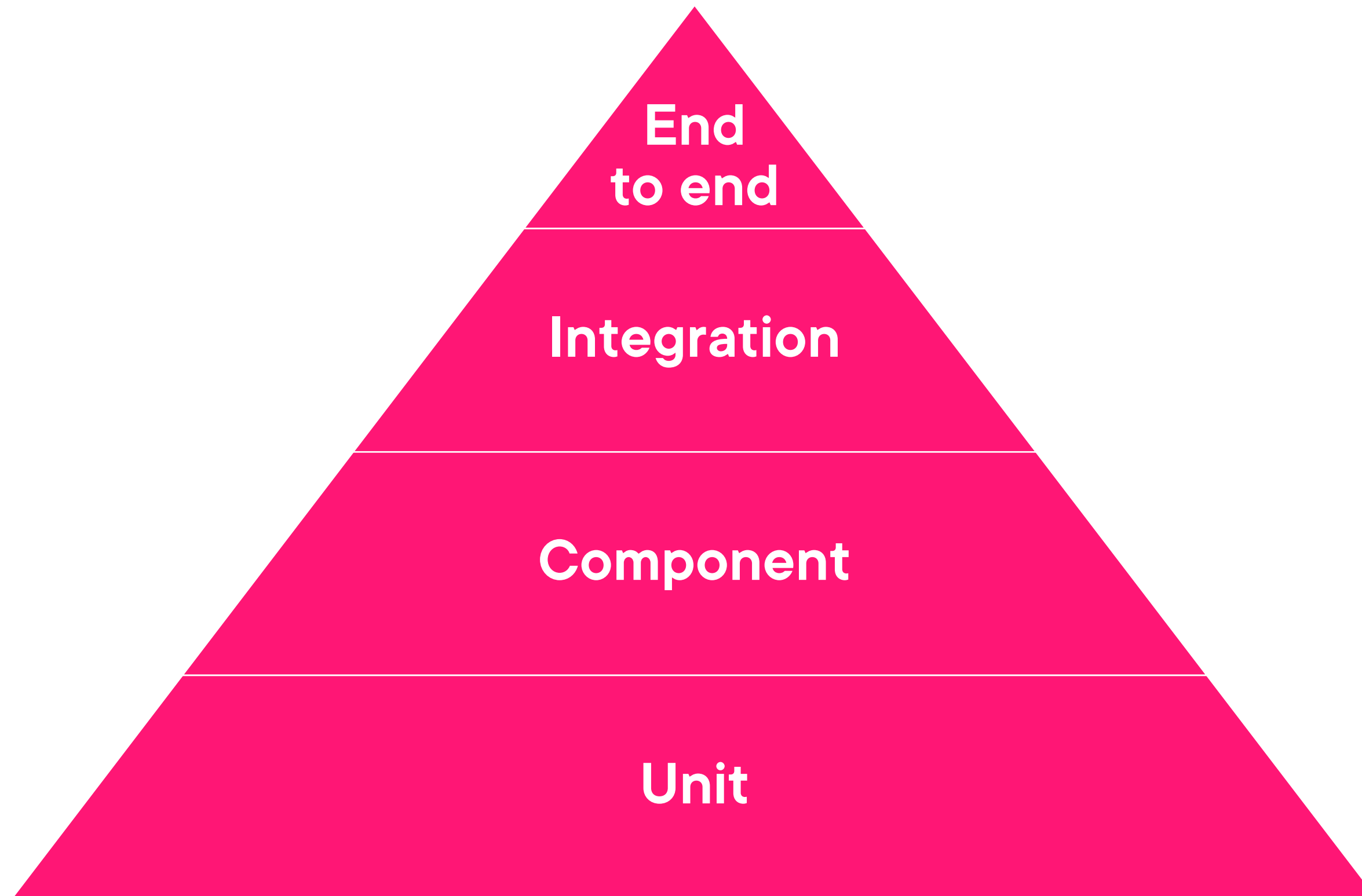
The Testing API

Black and White Box Testing

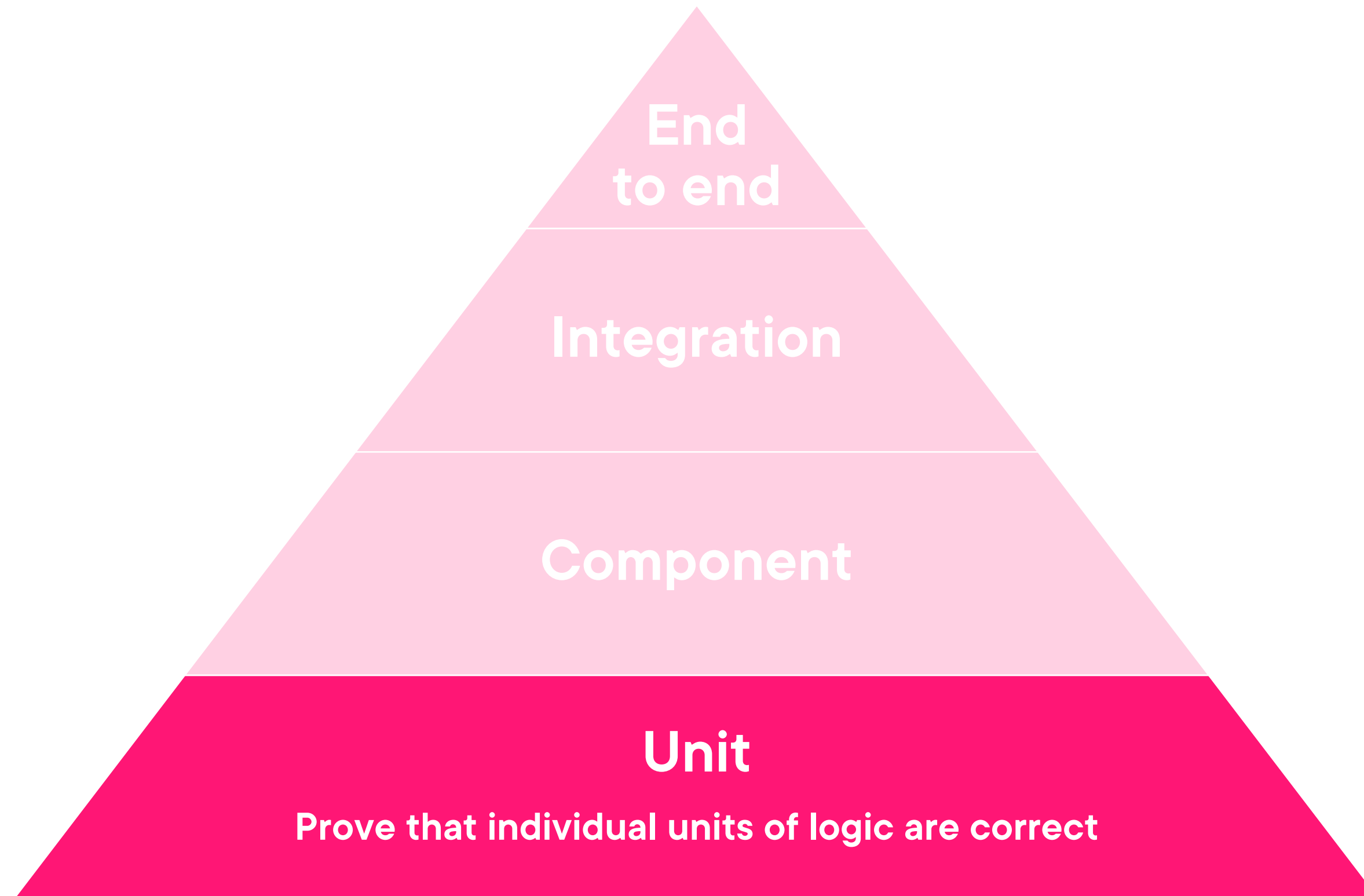
Targeting Test Runs



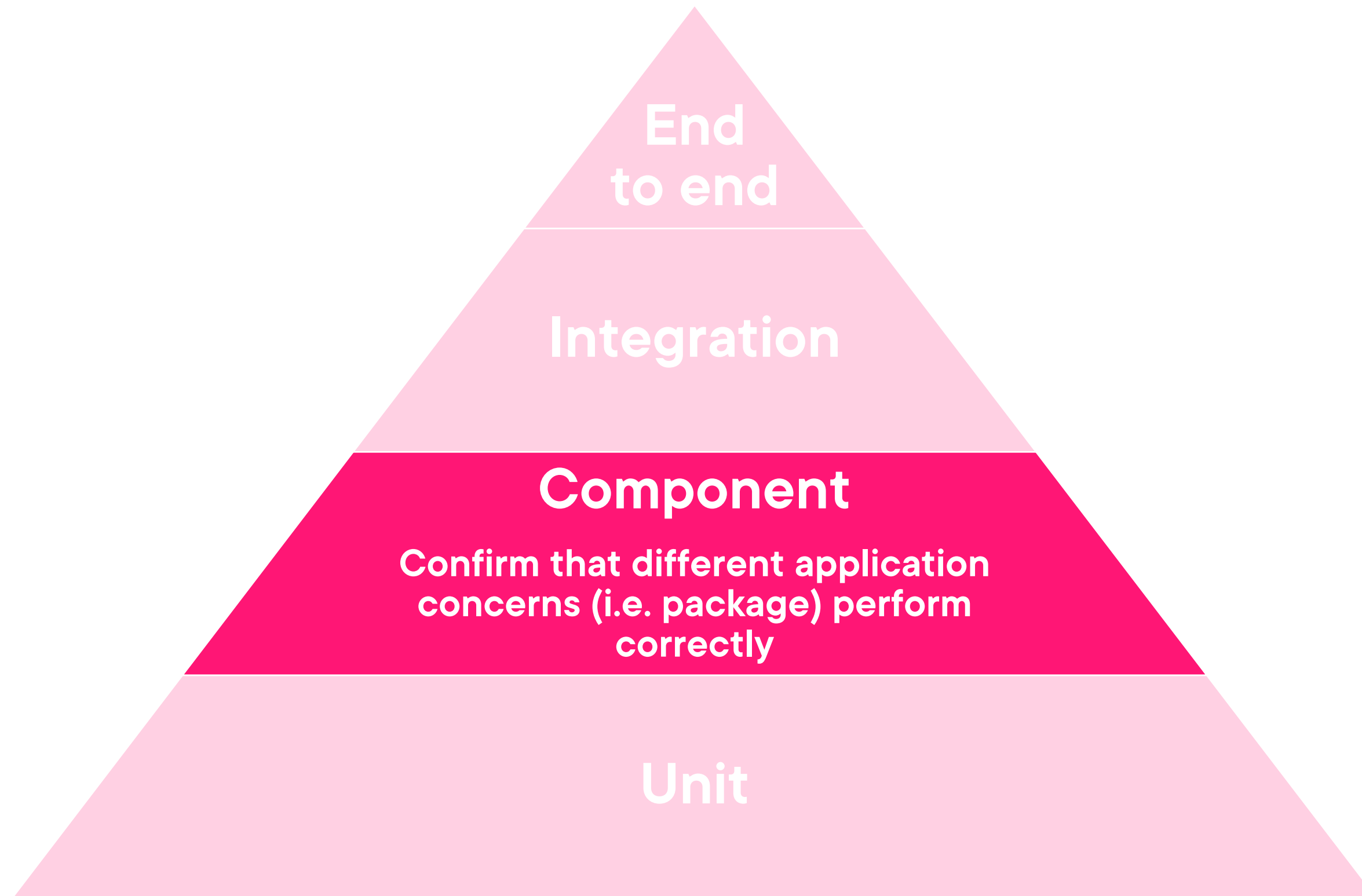
The Testing Pyramid



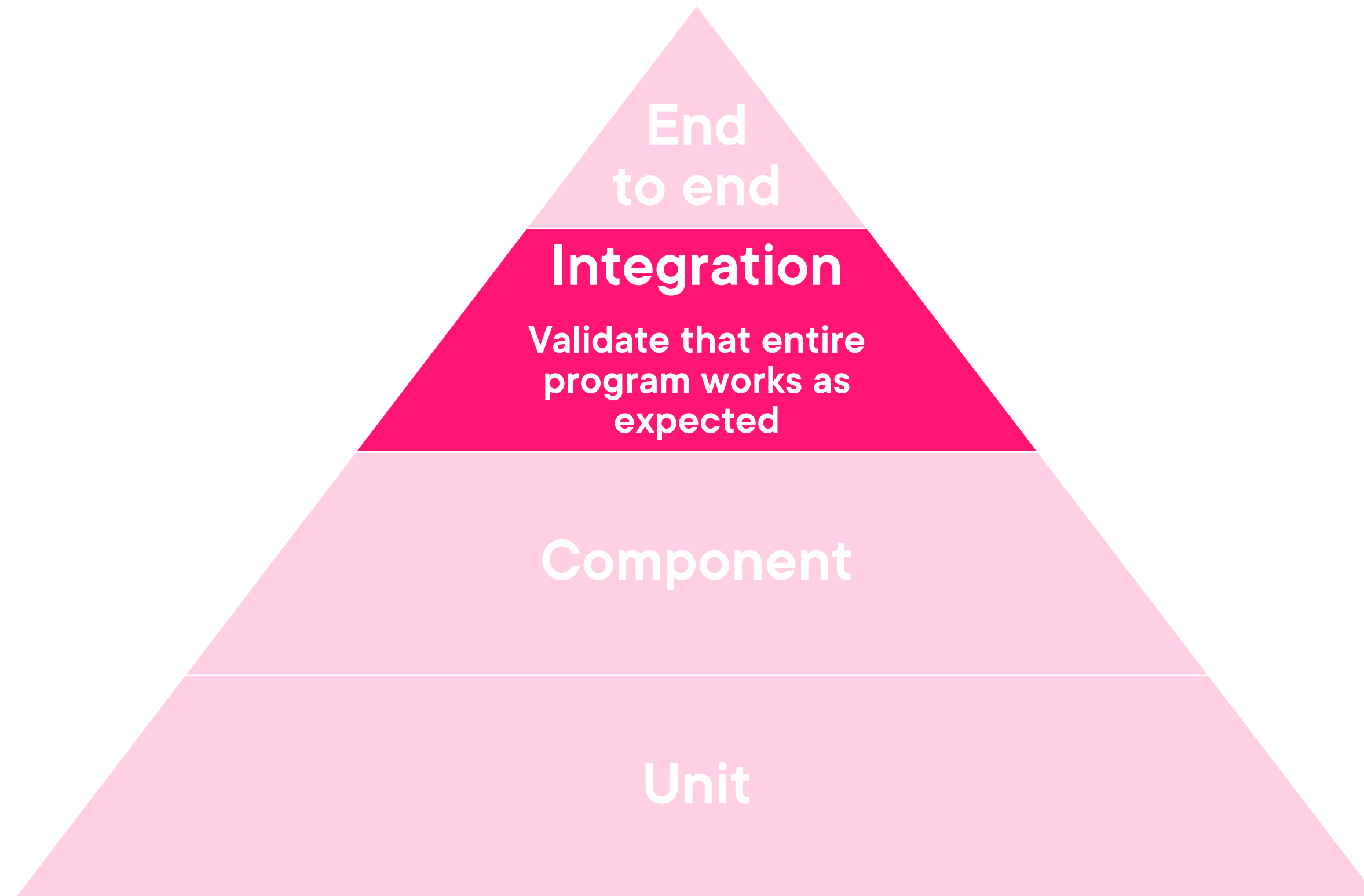
The Testing Pyramid



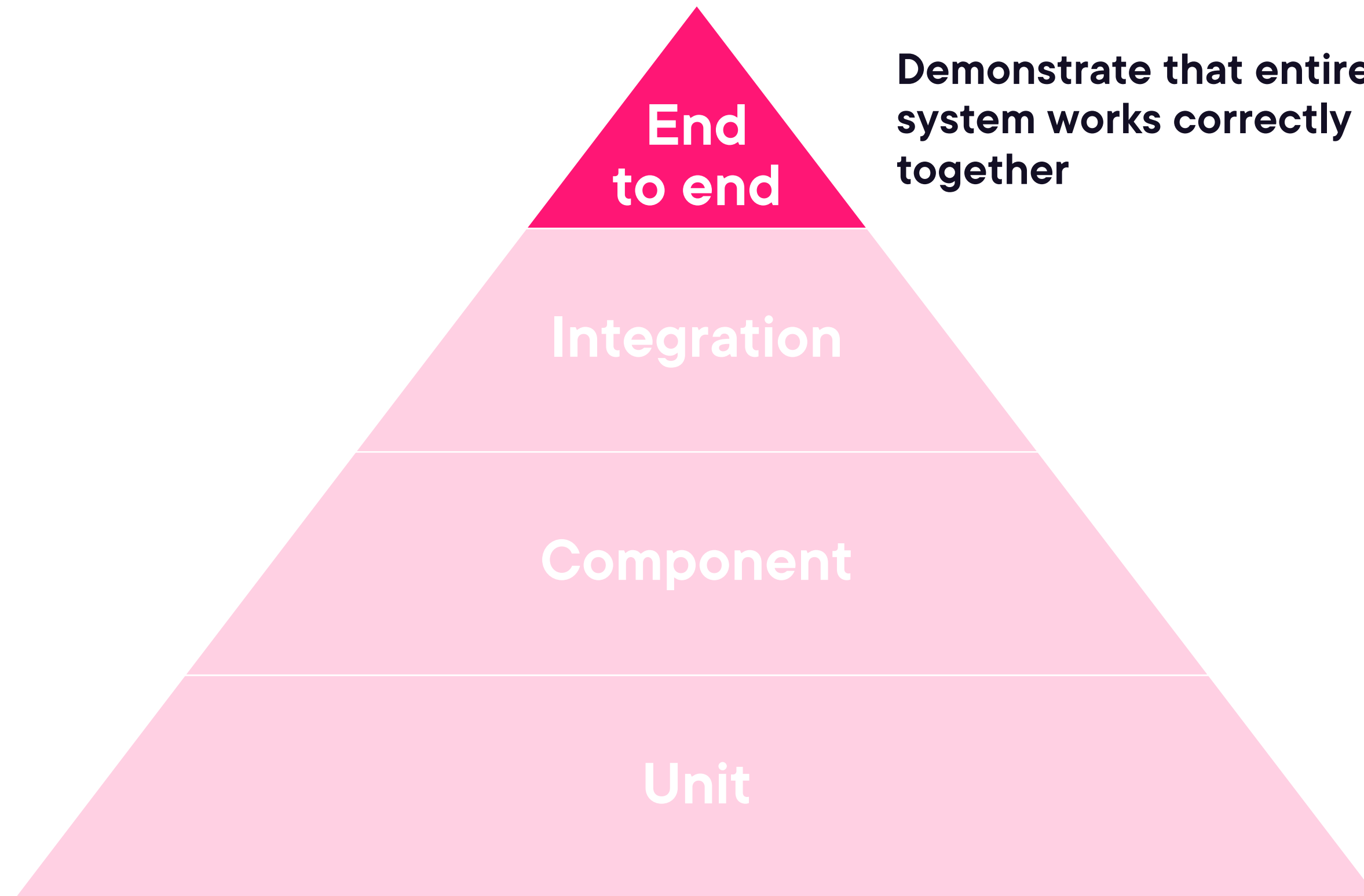
The Testing Pyramid



The Testing Pyramid



The Testing Pyramid



Testing API

```
type T
func (c *T) Cleanup(f func())
func (t *T) Deadline() (deadline time.Time, ok bool)
func (c *T) Error(args ...any)
func (c *T) Errorf(format string, args ...any)
func (c *T) Fail()
func (c *T) FailNow()
func (c *T) Failed() bool
func (c *T) Fatal(args ...any)
func (c *T) Fatalf(format string, args ...any)
func (c *T) Helper()
func (c *T) Log(args ...any)
func (c *T) Logf(format string, args ...any)
func (c *T) Name() string
func (t *T) Parallel()
func (t *T) Run(name string, f func(t *T)) bool
func (t *T) Setenv(key, value string)
func (c *T) Skip(args ...any)
func (c *T) SkipNow()
func (c *T) Skipf(format string, args ...any)
func (c *T) Skipped() bool
func (c *T) TempDir() string
```

pkg.go.dev/testing

Communicate with test runner

- Report failures
- Logging
- Configure how test is executed
- Interact with environment

No assertions!



Reporting Test Failures

Non-immediate failures

t.Fail()

t.Error(...interface{})

t.Errorf(string, ...interface{})

vs

Immediate failures

t.FailNow()

t.Fatal(...interface{})

t.Fatalf(string, ...interface{})



Why are Assertions Missing?

```
// reduce learning curve
func TestThing(t *testing.T) {
    l, r := 2, 4
    expect := 6

    got := l + r

    if got != expect {
        // report failure
    }
}
```

Checking for test failures uses same constructs as production code.

If you can write production code, you can write tests.



Why are Assertions Missing?

```
// focus on common concerns
func TestThing(t *testing.T) {
    l, r := 2, 4
    expect := 6

    got := l + r

    assert.Equal(got, expect)    // assertion style
    Expect(expect).To(Equal(got)) // expect style
}
```

**Both styles are equally valid and
subject to team preference**





show testing package

- focus on T object

create unit test

- failures
- logs





create integration test (?)





Running tests in parallel



White versus Black Box Testing

White Box Tests

Test has access to all tested code

**Tests in same package as
production code**

vs

Black Box Tests

**Test can only interact via public
API**

Tests in separate package





White box test

Black box test





Skipping tests
filtering tests



Summary



The Testing Pyramid

The Testing API

Black and White Box Testing

Targeting Test Runs

