



Testing 102

A Brief of whole pieces

Azzam S.A

Goals

- Why / What / How
- Best Practice

Why

- Confidence
- Financial
- Better Code

Didn't Everyone Know it?

Doing it wrong (probably?)

What?

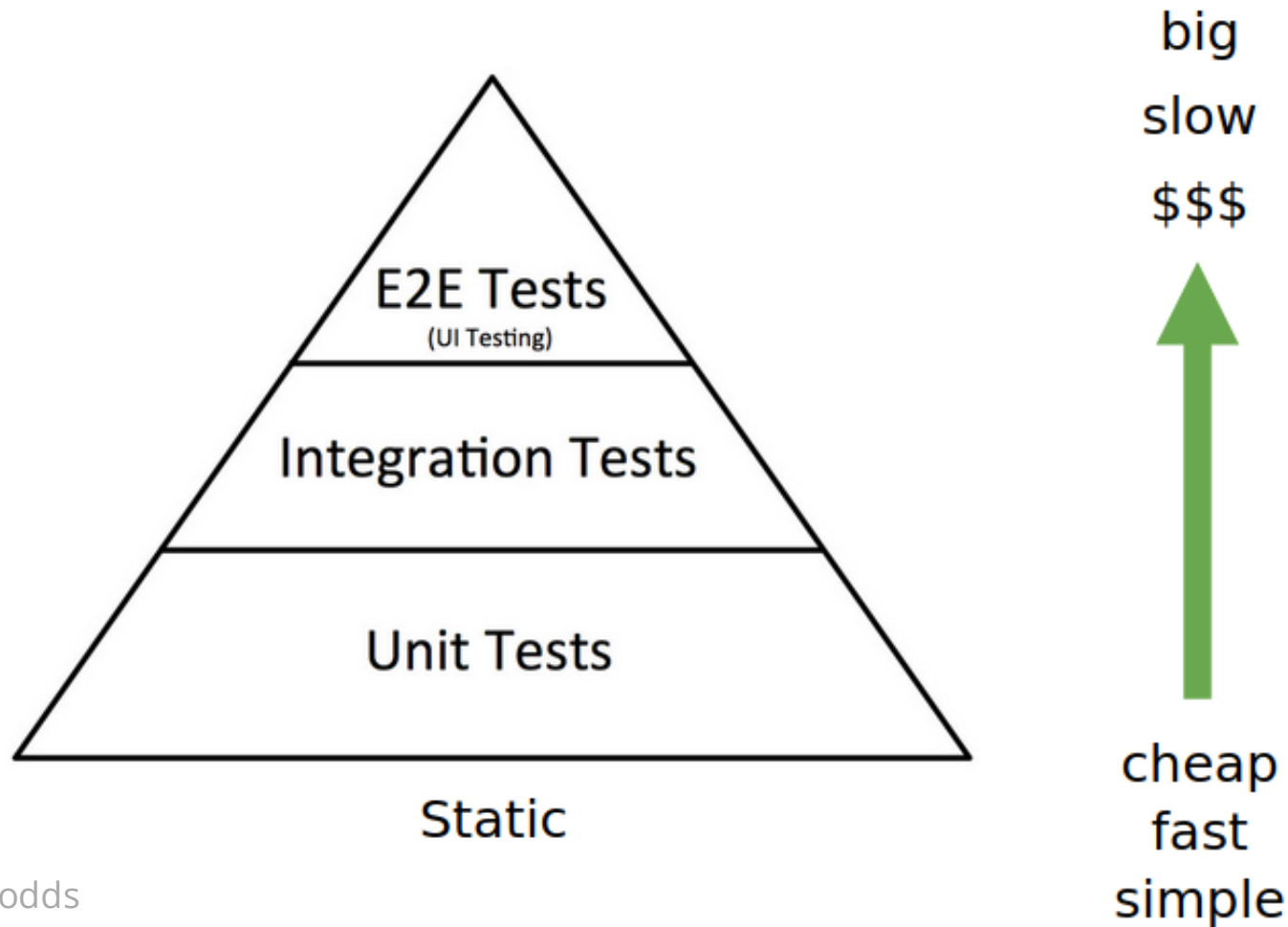
Testing

```
def inc(x):  
    return x + 1  
  
def test_answer():  
    assert inc(3) == 5
```

PYTHON

Automated vs Manual

Unit vs Integration vs E2e



@kentcdodds

Monkey Patch

```
from module import cat
```

PYTHON

```
def speak(self):  
    return "Hi, everybody!"
```

```
cat.speak = speak
```

Test Doubles

Stub

PYTHON

```
def fake_get_objects(self, resource, bucket, prefix):  
    return {"Contents": [{"Key": "foo.png"}], "CommonPrefixes": [{"Prefix": "a/b/"}]}  
  
def test_is_exists(self, monkeypatch):  
    monkeypatch.setattr("get_objects", self.fake_get_objects)  
  
    assert is_exists(resource, "bucket-a", "foo.png") is True  
    assert is_exists(resource, "bucket-a", "bar.png") is False
```

Test Doubles

Mock

PYTHON

```
def buckets(resource):
    """Print all bucket with specified attribute."""
    all_buckets = bucket_lib.buckets(resource)
    for bucket in all_buckets:
        click.secho(f"{bucket.creation_date:%Y-%m-%d %H:%M:%S} {bucket.name}")

def fake_buckets(resource):
    bucket = mock.Mock()
    bucket.name = "bucket-one"
    bucket.creation_date = datetime(2019, 9, 24, 1, 1, 0, 0)
    return bucket

def test_ls(monkeypatch, resource):
    monkeypatch.setattr(obs.libs.bucket, "buckets", fake_buckets)
    result = runner.invoke(cli, ["storage", "ls"])
    assert result.output == (f"2019-09-24 01:01:00 bucket-one\n")
```

How

Unit

PYTHON

```
def gen_random_name(prefix):  
    """Take random UUID and append specified prefix."""  
    return f"{prefix}-{str(uuid.uuid4())[:13]}"  
  
def test_random(self, monkeypatch):  
    monkeypatch.setattr(uuid, "uuid4", lambda: "71e43e94-c10d")  
    assert bucket.gen_random_name("awesome") == f"awesome-71e43e94-c10d"
```

Integration

PYTHON

```
def test_mb(monkeypatch, plain_auth):
    monkeypatch.setattr(obs.libs.auth, "strtobool", lambda http: True)
    monkeypatch.setattr(requests, "put", fake_mb)
    monkeypatch.setattr(obs.libs.utils, "check_plain", lambda response: None)

    runner = CliRunner()
    result = runner.invoke(cli, ["storage", "mb", "bucket-one", "--random"])

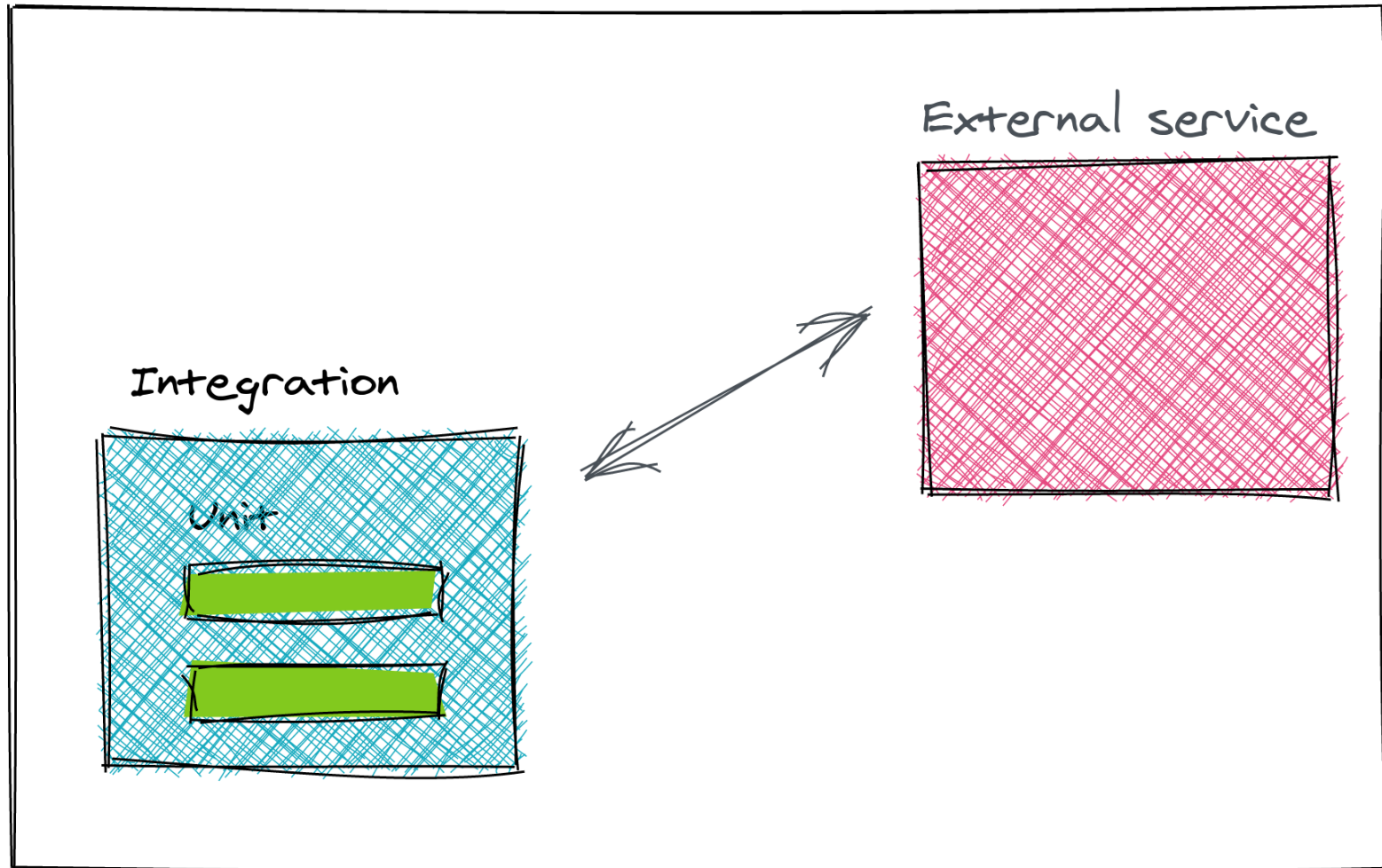
    assert result.output == f'Bucket "bucket-one" created successfully\n'
```

End to End

PYTHON

```
def test_mb(monkeypatch, plain_auth):  
  
    runner = CliRunner()  
    result = runner.invoke(cli, ["storage", "mb", "bucket-one", "--random"])  
  
    assert result.output == f'Bucket "bucket-one" created successfully\n'
```

End to end



Where to Start

- Most upset part
- unit → integration
- End to End
- CI

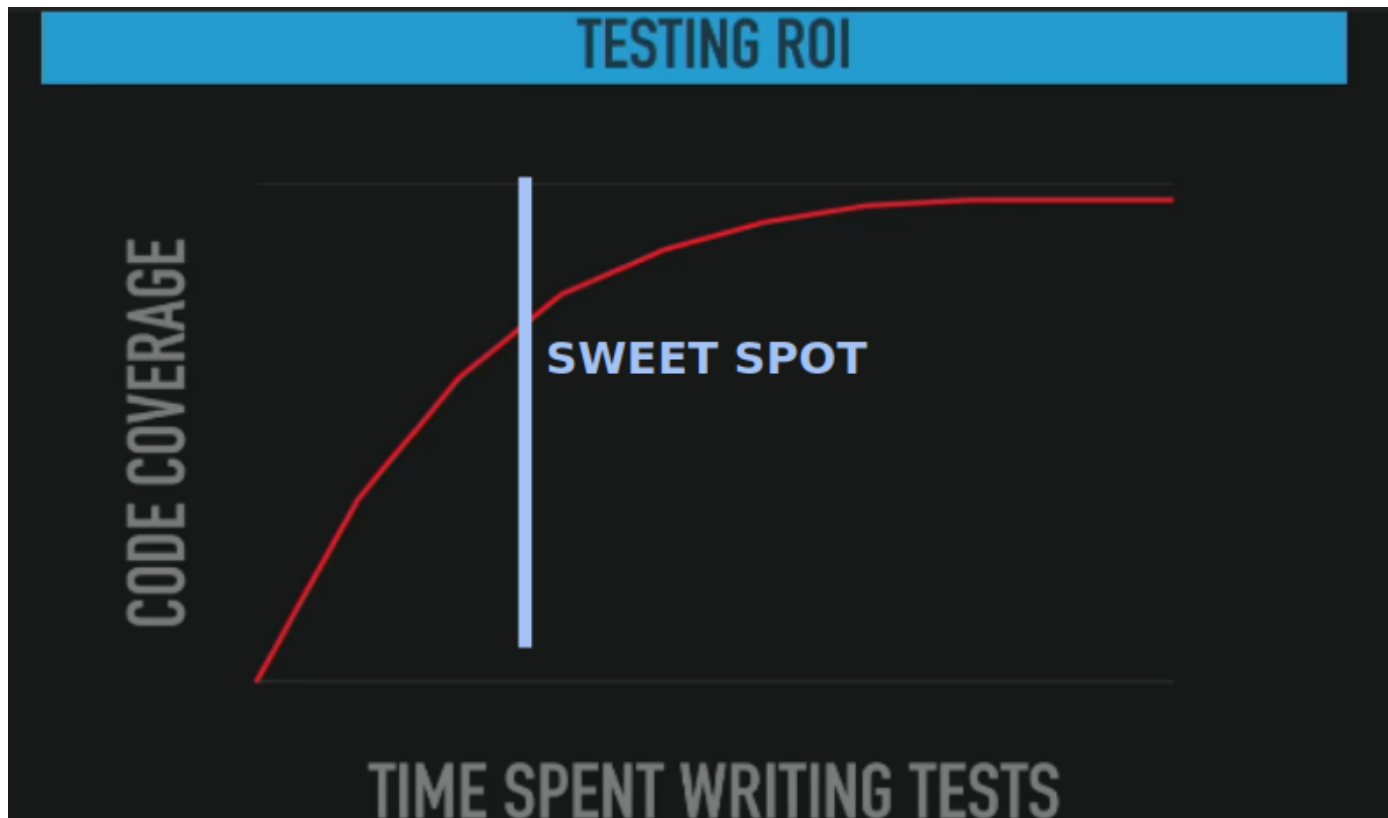
Tips

Avoid Implementation Details



giphy

100% code coverage?



Aaron Abramov

100% code coverage?

- Testing **any** code
- Doing Implementation detail
- Chaos, when refactoring

Mostly Integrations

```
assert is_umbrella_opens() is True
```

PYTHON

```
# Tests: Passed!
```

Mostly Integrations



giphy

Avoid Test Repetation



Thank You!

twitter @azzamsyawqi

www azzamsa.com



Credits

- The Testing Garden — Kent C. Dodds
- Establishing testing patterns with software design principles
— Aaron Abramov