These Are Not Thests You Are Looking For

International PHP Conference – Spring Edition

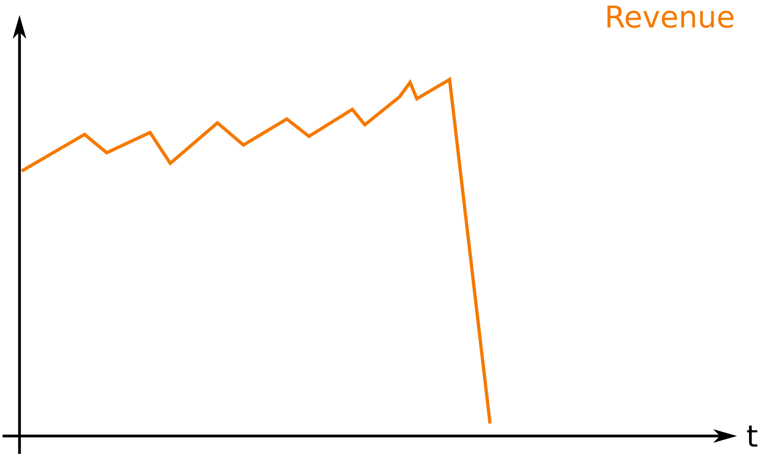
Tobias Schlitt (@tobySen) & Kore Nordmann (@koredn)

3rd June 2014

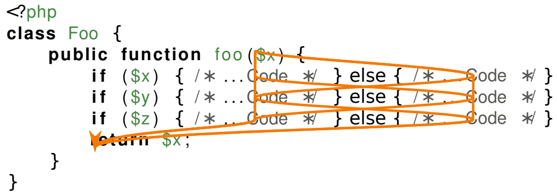
Software Quality

Why care?

Revenue goes down. . .



Complexity



```
<?php
class Foo {
    public function foo($x) {
        if ($x) { /* ...Code */ } else { /* ...Code */ }
        if ($y) { /* ...Code */ } else { /* ...Code */ }
        if ($z) { /* ...Code */ } else { /* ...Code */ }
        return $x;
    }
}
```

Sensible Testing

Cover every line of code

- Does not mind side effects
- Does not cover different pre-conditions

Sensible Testing

Cover every execution path

- **You should write at least $nPath tests for every method!**
- Does not mind different parameter values

Sensible Testing

Cover every execution path with sensible parameters

- Common integer boundaries: $-2^{63}, -2^{31}, -1, 0, 1, 2^{31}, 2^{63}$
- **You should write at least** $\$nPath * \$parameterCount * \$boundaries$ **tests per method!**

Qafoo
passion for software quality

sqrt()

$$\sqrt{}$$

Qafoo
passion for software quality

E_TOO_MANY_TESTS

WTF?

Integer Overflow

We refactored projects with a NPath complexity $> 2^{64}$ in controllers

This means more then

18,446,744,073,709,551,616 execution paths!

- Development obviously was stalled…
  - Nobody understands possible side effects any more
  - This is impossible to test

Wrap-Up

- We do not require ultimate stability
  - We do PHP for development speed (adaption to changes)
  - We can deploy our full stack in a couple of minutes
  - Refactor before complexity explodes
- What we actually should do:
  - Estimate business impact of code
  - Write sensible integration tests

Business Impact

- Which code has ...
  - direct impact on revenue*?
  - indirect impact on revenue*?
  - no impact on revenue*?

* you might have different business goals then just revenue

Business Impact

How can a developer know?

- Familiarize yourself with the business goals
- Ask for business metrics
- Measure and watch important business metrics
- Product owner annotates business impact in user stories

Qafoo
passion for software quality

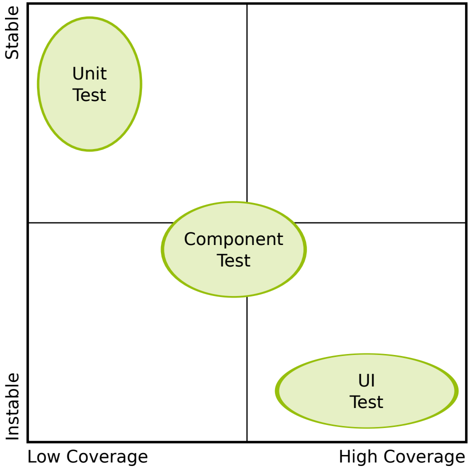
Impact Of Code

- There are metrics which show the impact of code on the system:
  - Afferent Coupling (C_A) / Efferent Coupling (C_E)
  - Code-Rank / Reverse Code-Rank
- A really “unimportant” component still might break about everything

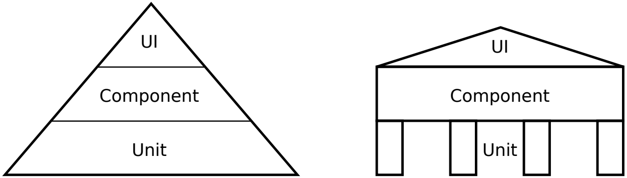
Testing The Full Stack

- Making sure the important stories work
  - Does not ensure that everything works
  - . . . but the most important bits will work!
- Large tests do not really help debugging.

Trade-Offs


Stable
Instable
Unit Test
Component Test
UI Test
Low Coverage
High Coverage

The Test Pyramid



Effective Component Tests

- Mock at component borders
  - … throw out the database
  - … ignore the SOAP endpoint
- Requires:
  - Sane and simple APIs (Facades)
  - Dependency Inversion (Injection)

Test Driven Development

- Use TDD as a design principle
  - Unit Tests always converge to Integration Tests
- You can TDD using using Unit-, Component- & UI-Tests

Summary

- Unit Tests with full coverage are a great learning tool
- Write testable code, focus on testing important bits
- Test everything which broke once
- Make sure the important business cases always work

We are

passion for software quality

Helping people to create high quality web applications.

`http://qafoo.com`

- Trainings, Workshops and Consulting

Qafoo

passion for software quality

THANK YOU

Rent a quality expert

qafoo.com