MCR 2017

FABIAN SCHMENGLER

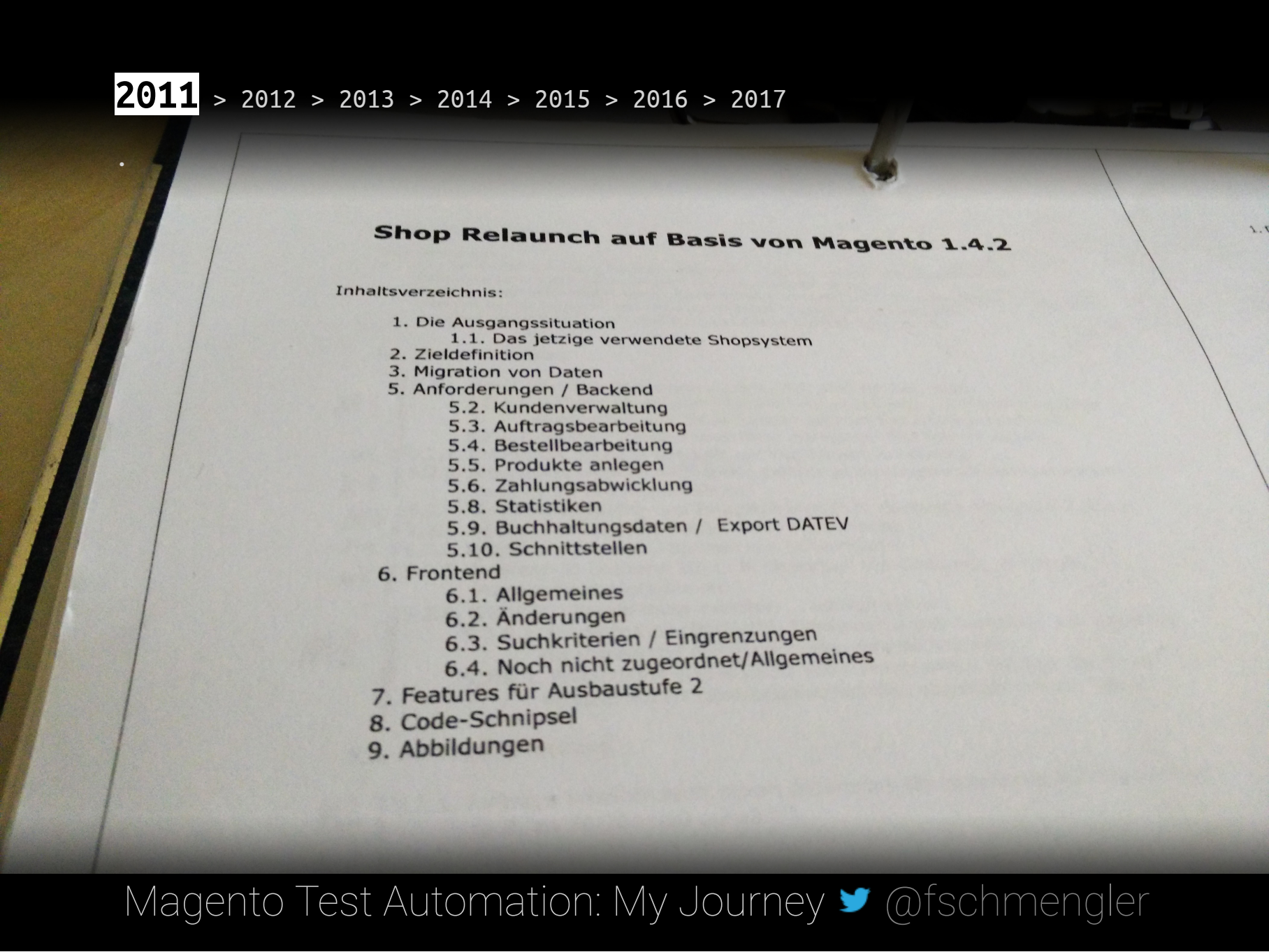
Magento Backend Developer at integer_net

@fschmengler

Magento test automation: my journey

UK.MAGETITANS.COM

#MageTitansMCR @MageTitans



Shop Relaunch auf Basis von Magento 1.4.2

Inhaltsverzeichnis:

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

5.6. Zahlungsabwicklung
5.8. Statistiken
5.9. Buchhaltungsdaten / Export DATEV
5.10. Schnittstellen

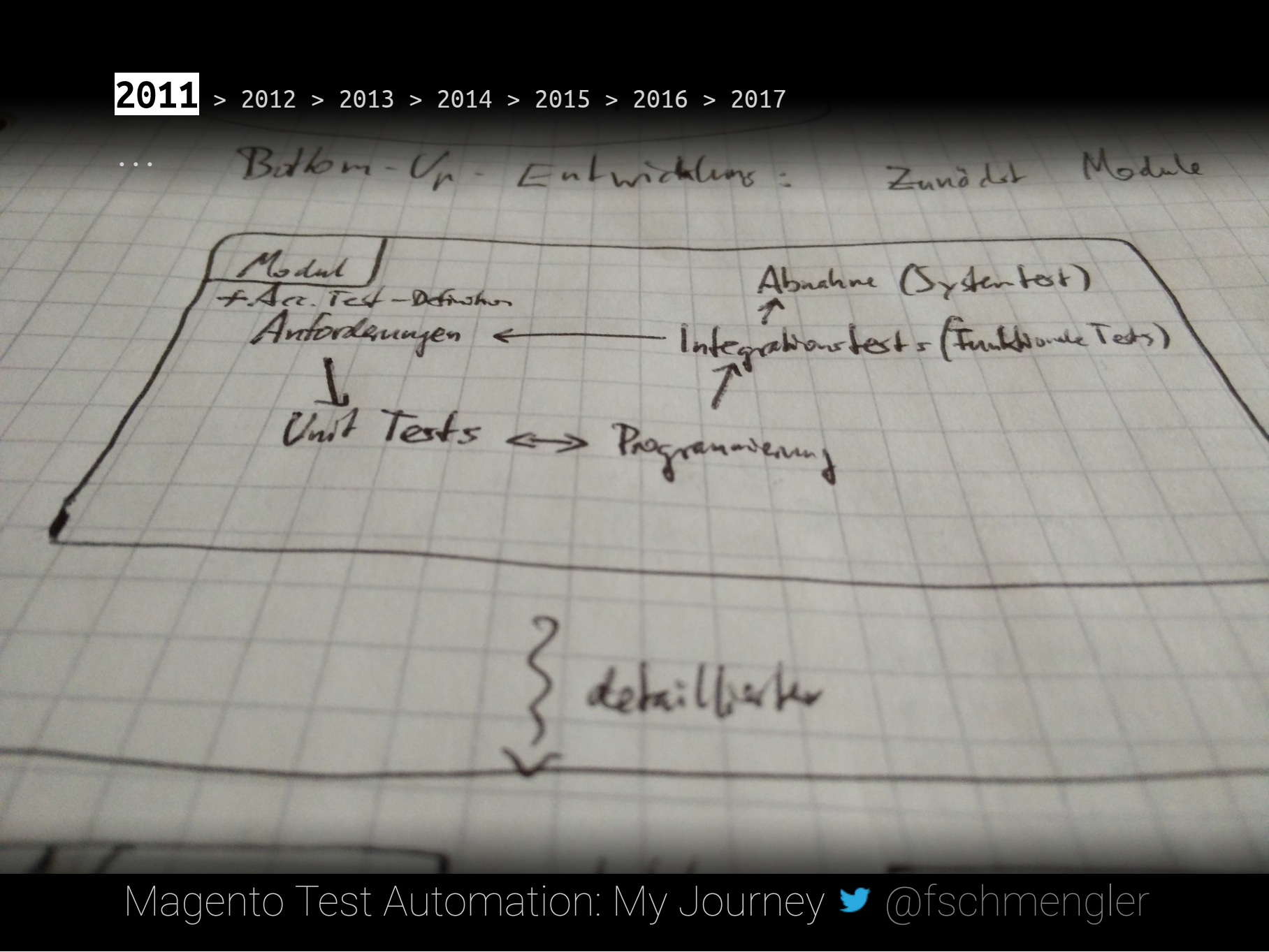
6. Frontend
   6.1. Allgemeines
   6.2. Änderungen
   6.3. Suchkriterien / Eingrenzungen
   6.4. Noch nicht zugeordnet/Allgemeines
7. Features für Ausbaustufe 2
8. Code-Schnipsel
9. Abbildungen

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

..

Magento Test Automation: My Journey @fschmengler

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017



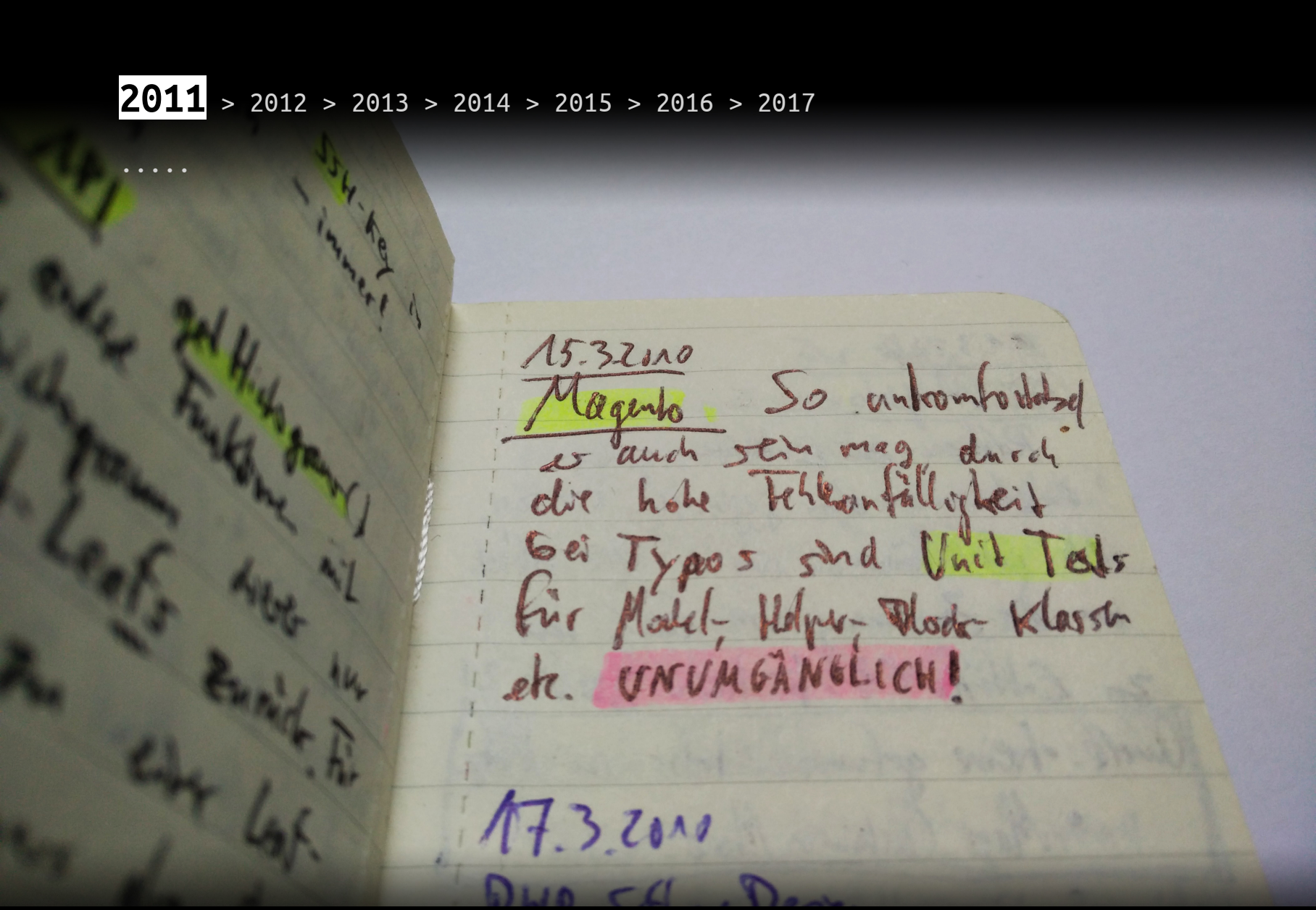
2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

....

- SimpleTest? PHPUnit?
- Ibuildings MageTest? EcomDev PHPUnit?

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....



Magento Test Automation: My Journey @fschmengler

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....

foo_bar?
Foo_Bar?
fooBar?
foobar?
bar?
foo/bar?

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....

TEST ALL THE THINGS!

```
public function testObserver()
{
    $this->assertEventObserverDefined(
        'adminhtml', 'core_block_abstract_prepare_layout_before',
        SGH_OrderPrintPopups_Model_Observer_Block::MODEL,
        'onCoreBlockAbstractPrepareLayoutBefore', 'orderprintpopups'
    );
}
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....

TEST ALL THE THINGS!

```
/**
 * @param Mage_Core_Block_Abstract $block
 * @return Varien_Event_Observer
 */
protected function _mockEventObserver($block, $otherArgs = array())
{
    $args = array_merge(array('block' => $block), $otherArgs);
    $observer = new Varien_Event_Observer();
    $observer->setEvent(new Varien_Event($args));
    return $observer;
}
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

....

TEST ALL THE THINGS!

```
$this->assertLayoutHandleLoaded(
    'orderprintpopups_warensendung_index',
    'module layout handle loaded'
);
$this->assertLayoutBlockRendered(
    SGH_OrderPrintPopups_Block_Popup_Warensendung::ALIAS,
    'popup block rendered'
);
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....

TEST ALL THE THINGS!

```
$route = 'orderprintpopups/warensendung/barcodestatic';
$this->assertResponseBodyQueryRegex(
    '#sgh_warensendung',
    '#<img [^>]*src="[^"]+' . $route . '#'
);
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....

TEST ALL THE THINGS!

```
/**
 * getter tests
 */
test("getter", function() {
        equal(this.block.getName(), "foo", "getName()");
        equal(this.block.getContent(), "lorem ipsum falleri fallera", "getContent()");
        equal(this.block.getPlaceholder(), "{foo}", "getPlaceholder()");
});
```

.....

What lies behind the code

- Hours of debugging
- Hidden depenencies to global state
- Bugs of the testing frameworks

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

```
...........S
```

A story of skipped tests

```
$this->markTestSkipped('fatal error???');
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

```
...........S
```

A story of skipped tests

```
$this->markTestSkipped('fatal error???');

$this->markTestSkipped('conflict with fixture and existing db');
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

```
............S
```

A story of skipped tests

```
$this->markTestSkipped('fatal error???');

$this->markTestSkipped('conflict with fixture and existing db');

$this->markTestSkipped('"out of stock", although fixture seems ok'); //FIXME
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

```
...........S
```

A story of skipped tests

```
$this->markTestSkipped('fatal error???');

$this->markTestSkipped('conflict with fixture and existing db');

$this->markTestSkipped('"out of stock", although fixture seems ok'); //FIXME

$this->markTestSkipped(
    'The new comparator classes in PHPUnit 3.6 cannot handle recursive comparision with
    $maxDepth parameter YET.'
);
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

```
...........S
```

A story of skipped tests

```
$this->markTestSkipped('fatal error???');

$this->markTestSkipped('conflict with fixture and existing db');

$this->markTestSkipped('"out of stock", although fixture seems ok'); //FIXME

$this->markTestSkipped(
    'The new comparator classes in PHPUnit 3.6 cannot handle recursive comparision with
    $maxDepth parameter YET.'
);

$this->markTestSkipped(
    'expectation + fixture changed. reason for missing prefix unknown '.
    'but only occurs in test environment!'
);
```

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

```
............SS
```

A story of skipped tests

```
$this->markTestSkipped(
    'Das klappt schon wieder alles nicht mit Adminlogin+Dispatch'
);
```

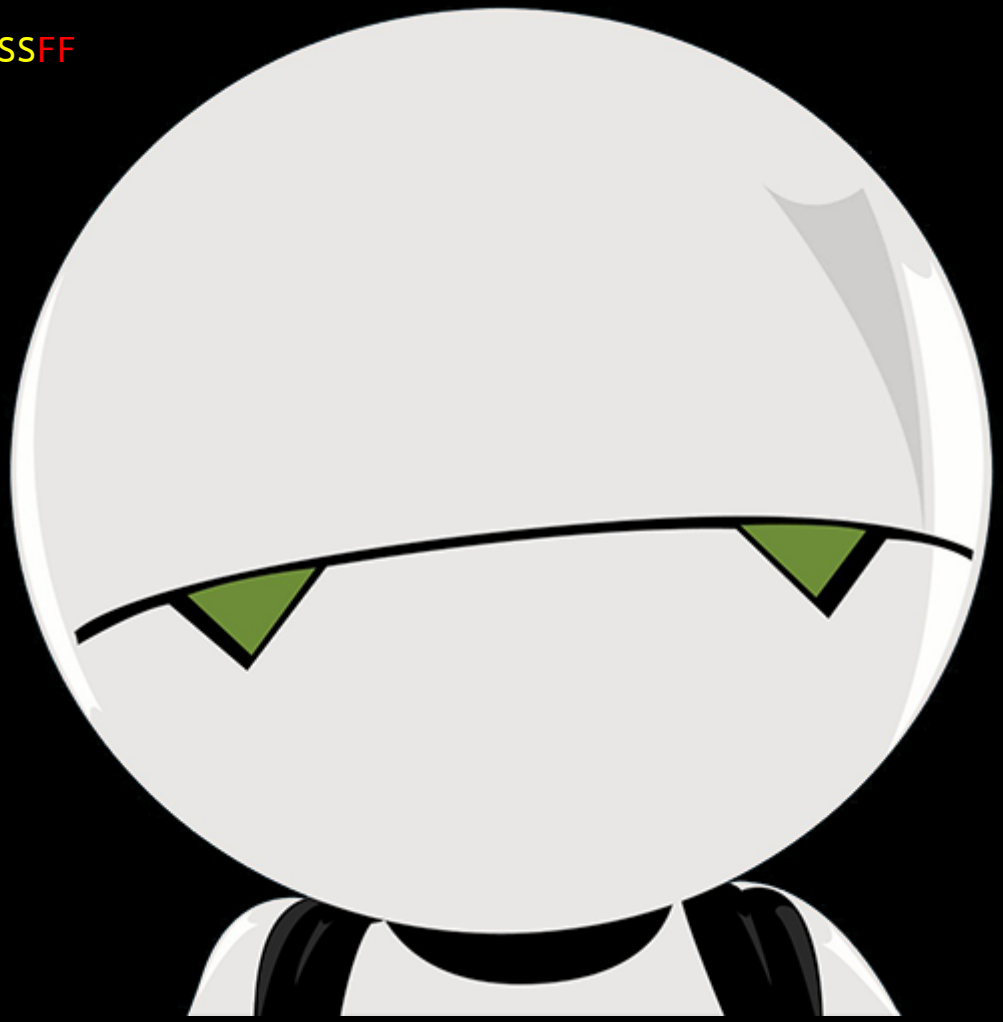
2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....SSF

Magento Test Automation: My Journey @fschmengler

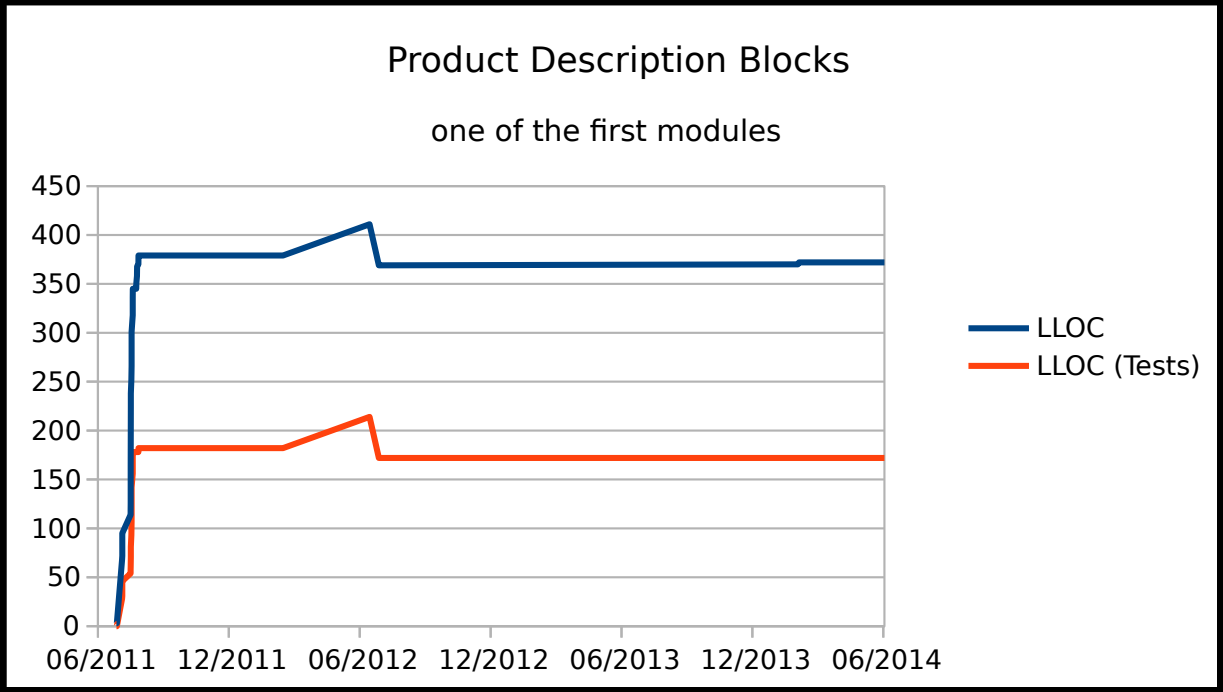
2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....SSFF



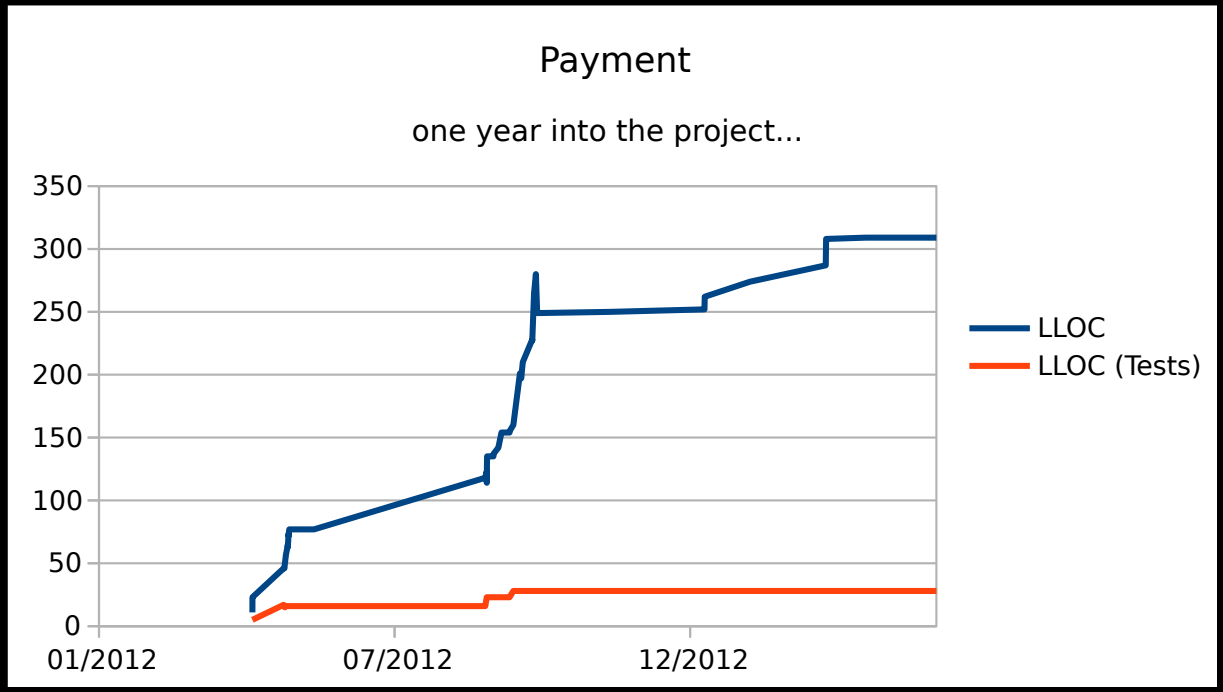
2011 > 2012 > 2013 > 2014 > 2015 > 2016 > 2017

.....SSFF.



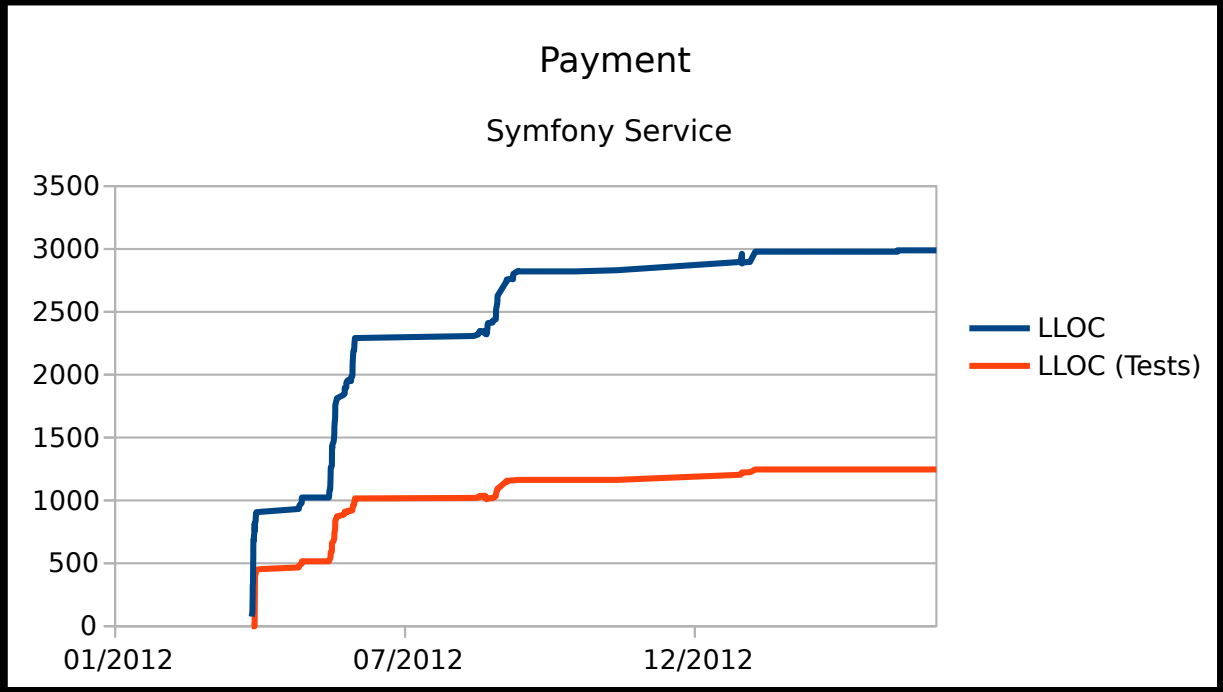
2011 > **2012** > 2013 > 2014 > 2015 > 2016 > 2017

.....SSFF.S



2011 > **2012** > 2013 > 2014 > 2015 > 2016 > 2017

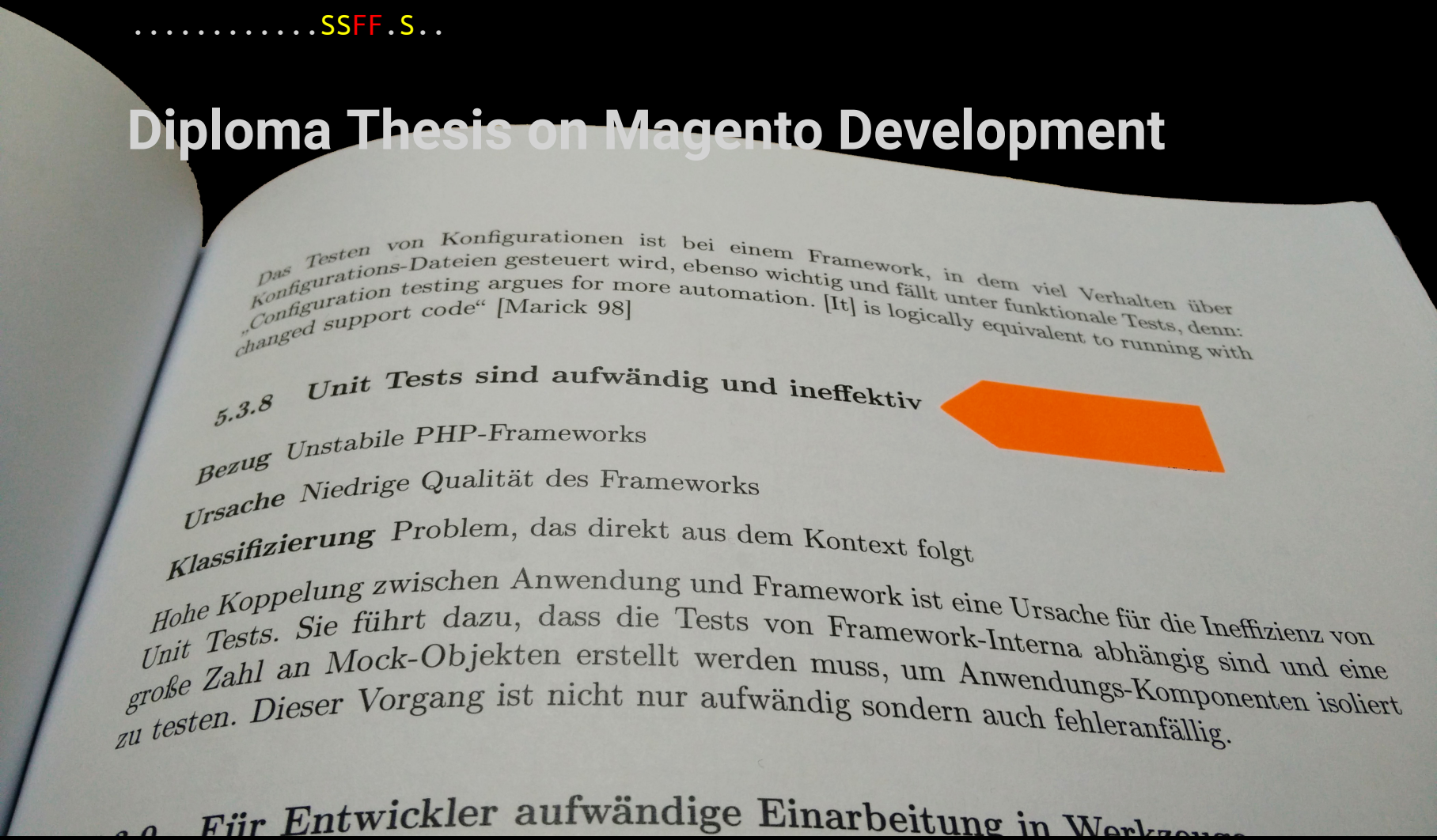
.....SSFF.S.



2011 > **2012** > 2013 > 2014 > 2015 > 2016 > 2017

.....SSFF.S..

Diploma Thesis on Magento Development



Das Testen von Konfigurationen ist bei einem Framework, in dem viel Verhalten über Konfigurations-Dateien gesteuert wird, ebenso wichtig und fällt unter funktionale Tests, denn: „Configuration testing argues for more automation. [It] is logically equivalent to running with changed support code“ [Marick 98]

5.3.8 Unit Tests sind aufwändig und ineffektiv

Bezug *Unstabile PHP-Frameworks*

Ursache *Niedrige Qualität des Frameworks*

Klassifizierung *Problem, das direkt aus dem Kontext folgt*

Hohe Koppelung zwischen Anwendung und Framework ist eine Ursache für die Ineffizienz von Unit Tests. Sie führt dazu, dass die Tests von Framework-Interna abhängig sind und eine große Zahl an Mock-Objekten erstellt werden muss, um Anwendungs-Komponenten isoliert zu testen. Dieser Vorgang ist nicht nur aufwändig sondern auch fehleranfällig.

Für Entwickler aufwändige Einarbeitung in Werkzeuge

2011 > **2012** > 2013 > 2014 > 2015 > 2016 > 2017

```
............SSFF.S...
```

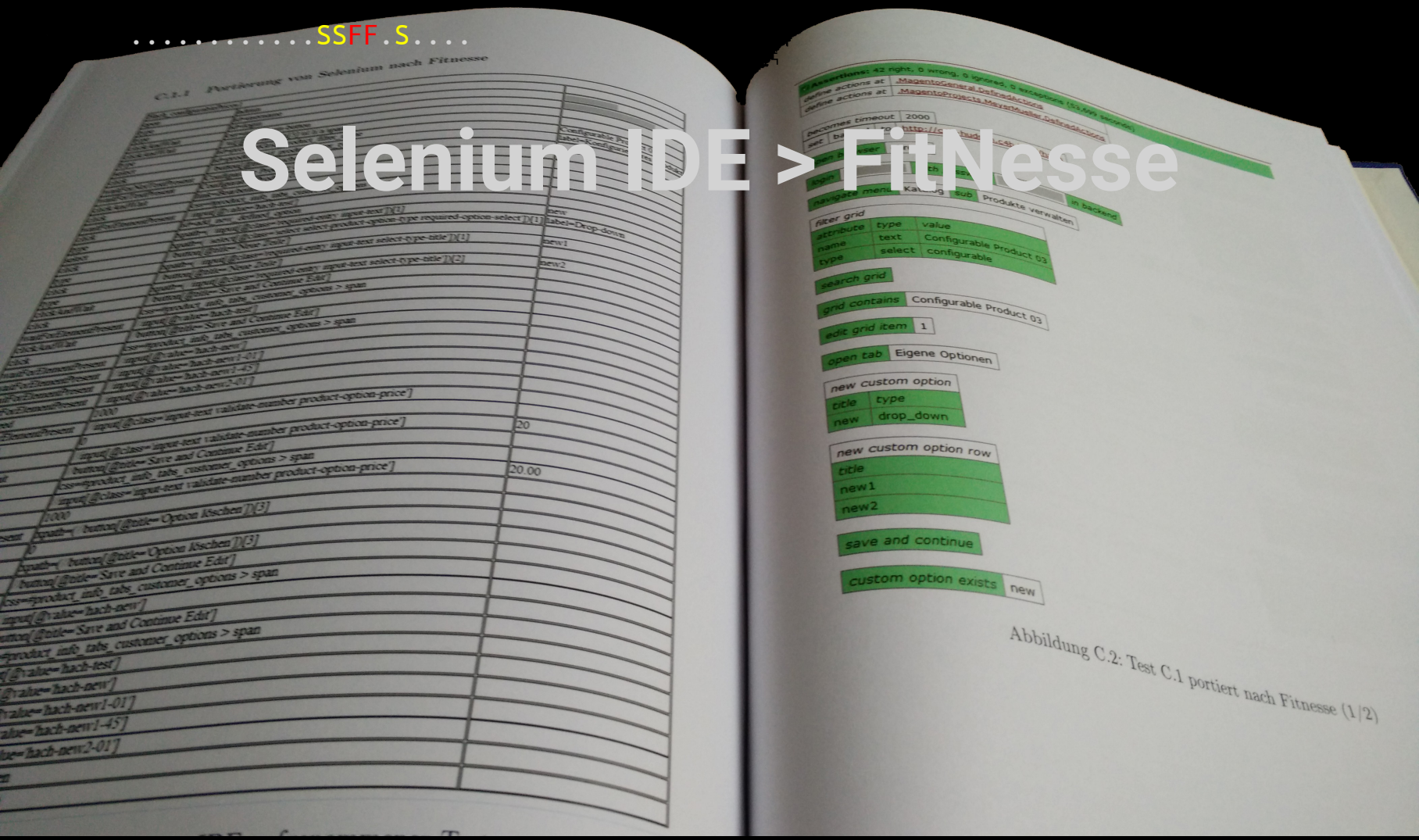
Research & Experiments

- Built-in tests
- Magento TAF for Smoke Test Suite
- Abstraction of automated acceptance tests
  FitNesse + ZiBreve

2011 > 2012 > **2013** > 2014 > 2015 > 2016 > 2017

.....SSFF.S....

Selenium IDE > FitNesse



2011 > 2012 > **2013** > 2014 > 2015 > 2016 > 2017

.....SSFF.S.....

Agency project with Selenium

Automated Acceptance Tests

- Selenium IDE
- Tests for every feature
- Partially written by QA

2011 > 2012 > **2013** > 2014 > 2015 > 2016 > 2017

.....SSFF.S.....S

2011 > 2012 > 2013 > **2014** > 2015 > 2016 > 2017

```
............SSFF.S.....S.
```

Agency project with EcomDev_PHPUnit

Automated Unit/Integration Tests

- Tests mandatory, coverage monitored
- Sophisticated setup
- Still typical problems

2011 > 2012 > 2013 > **2014** > 2015 > 2016 > 2017

.....SSFF.S.....S.F

DEADLINE IS COMING

2011 > 2012 > 2013 > **2014** > 2015 > 2016 > 2017

.....SSFF.S.....S.FF

M1 Integration Tests: Typical Problems

- Tests altering global state (singletons, registry)

2011 > 2012 > 2013 > **2014** > 2015 > 2016 > 2017

.....SSFF.S.....S.FF

M1 Integration Tests: Typical Problems

- Tests altering global state (singletons, registry)
- Tests depending on existing data, others delete it and use their own

2011 > 2012 > 2013 > **2014** > 2015 > 2016 > 2017

.....SSFF.S.....S.FF

M1 Integration Tests: Typical Problems

- Tests altering global state (singletons, registry)
- Tests depending on existing data, others delete it and use their own
- Tests not well isolated, not running in DB transaction

2011 > 2012 > 2013 > **2014** > 2015 > 2016 > 2017

.....SSFF.S.....S.FF

M1 Integration Tests: Typical Problems

- Tests altering global state (singletons, registry)
- Tests depending on existing data, others delete it and use their own
- Tests not well isolated, not running in DB transaction
  - Complicated fixture setup

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF.

THIS WEIRD UNIT TESTING TRICK BLEW MY MIND!

- **Decouple** business logic from Magento
- Use plain PHPUnit

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

```
............SSFF.S.....S.FF..
```

Teilweise lässt sich TDD dennoch anwenden, nämlich für Code-Einheiten, die in sich geschlossen sind und wenig mit dem Framework interagieren. Diese sind einfacher zu Unit-testen (siehe A.2.3 S. 190) und das Design ist weniger vom Framework abhängig.

Magento Test Automation: My Journey @fschmengler

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF...

Bad news

- we still need higher level tests

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF...

Bad news

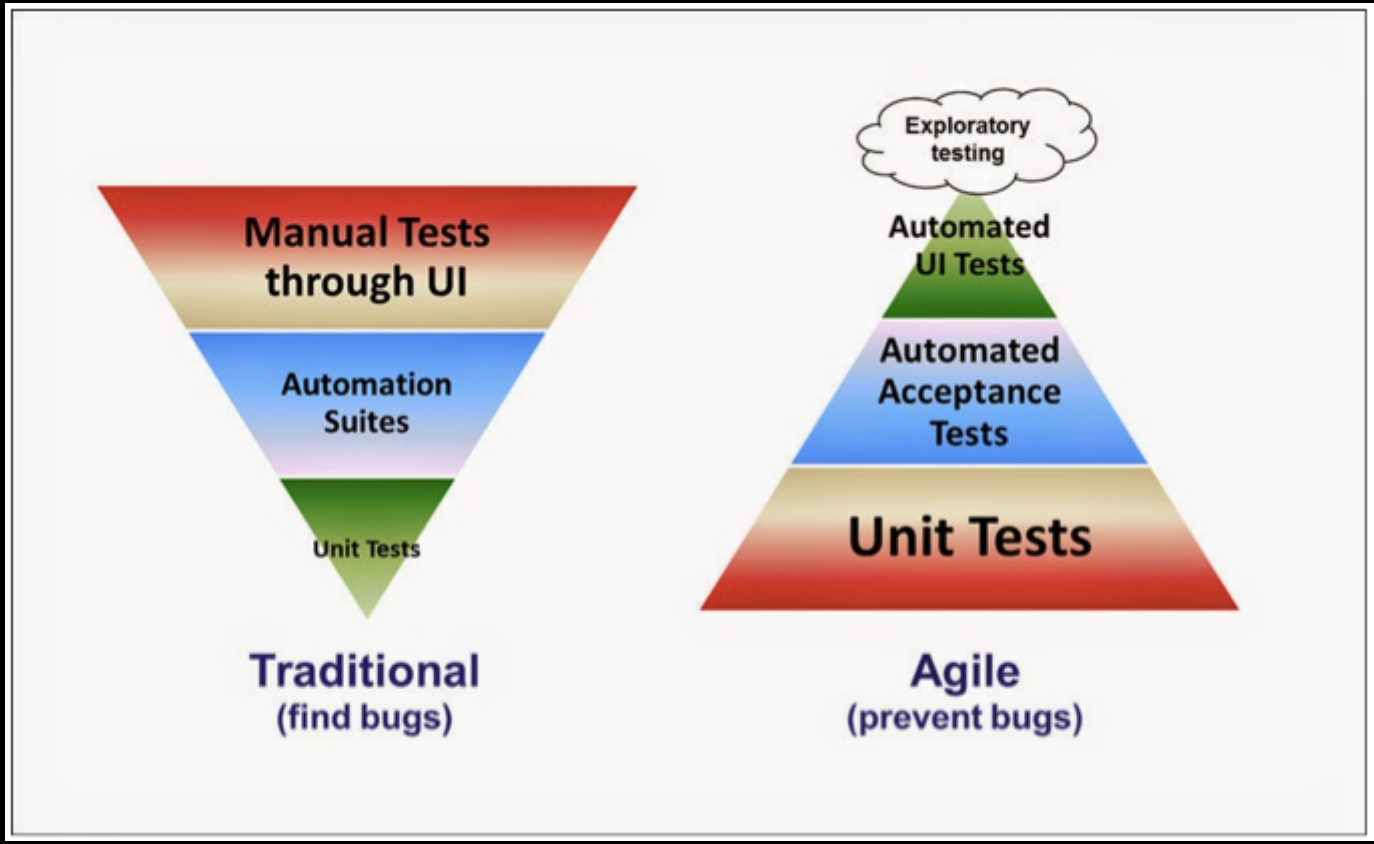
- we still need higher level tests

Good news

- less of them

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF....



2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF.....

Experiments

- Acceptance and integration test framework 

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF.....

Experiments

- Acceptance and integration test framework 

- Acceptance test framework MAGIUM

> "THE SELENIUM-BASED TESTING FRAMEWORK FOR PEOPLE WHO HATE TESTING"

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF.....

Experiments

- Acceptance and integration test framework 

- Acceptance test framework 

  > "THE SELENIUM-BASED TESTING FRAMEWORK FOR PEOPLE WHO HATE TESTING"

- Test framework Codeception 

2011 > 2012 > 2013 > 2014 > **2015** > 2016 > 2017

.....SSFF.S.....S.FF.....

A new era:

Magento 2

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

.....SSFF.S.....S.FF.......

Magento 2 Extension

- Reused refactored lib from Magento 1 extension
- Integration tests for Magento 2 specific code

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

```
............SSFF.S.....S.FF........
```

Magento 2 Projects

- Good integration test framework

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

```
............SSFF.S.....S.FF........
```

Magento 2 Projects

- Good integration test framework
- Easier to start with

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

```
............SSFF.S.....S.FF........
```

Magento 2 Projects

- Good integration test framework
- Easier to start with
- More reliable

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

```
............SSFF.S.....S.FF........S
```

Magento 2 Projects

- Still frustrating sometimes

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

```
............SSFF.S.....S.FF........S
```

Magento 2 Projects

- Still frustrating sometimes
- Failures after minor version update

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

```
............SSFF.S.....S.FF........S
```

Magento 2 Projects

- Still frustrating sometimes
- Failures after minor version update
- Test runs not completely isolated

2011 > 2012 > 2013 > 2014 > 2015 > **2016** > 2017

.....SSFF.S.....S.FF........S

Magento 2 Projects

- Still frustrating sometimes
- Failures after minor version update
- Test runs not completely isolated
- **Decoupled** approach still works best

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

```
............SSFF.S.....S.FF........S.
```

TDD Katas: Back to the roots

- Regular practice
- Small tasks, no real project
- **Goal:** Make TDD my *default habit*
- https://www.schmengler-se.de/katas

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

```
............SSFF.S.....S.FF........S..
```

How I would start today / Recommendations

1. For a **quick win**, create smoke test suite with Codeception

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

```
............SSFF.S.....S.FF........S..
```

How I would start today / Recommendations

1. For a **quick win**, create smoke test suite with Codeception
2. Katas / small self contained projects to **learn TDD**

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

```
............SSFF.S.....S.FF........S..
```

How I would start today / Recommendations

1. For a **quick win**, create smoke test suite with Codeception
2. Katas / small self contained projects to **learn TDD**
3. Start to build (small) parts of your code **independent** from Magento - "TDD" those parts

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

.....SSFF.S.....S.FF........S..

How I would start today / Recommendations

1. For a **quick win**, create smoke test suite with Codeception
2. Katas / small self contained projects to **learn TDD**
3. Start to build (small) parts of your code **independent** from Magento - "TDD" those parts
4. Try to add **simple integration tests** (in M1 with Xtest)
    - but skip if it's getting complicated

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

```
............SSFF.S.....S.FF........S...
```

Mistakes I won't make again

- Try to sell tests as separate part of project quote or estimation

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

```
............SSFF.S.....S.FF........S...
```

Mistakes I won't make again

- Try to sell tests as separate part of project quote or estimation
- Try to teach "unit testing" and "testing Magento" at once

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

```
............SSFF.S.....S.FF........S..I
```

Coming soon:

Test Driven Magento by Example

http://tddwizard.com/

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

.....SSFF.S.....S.FF........S..I

Time: 25 minutes

OK (39 slides, 0 cat pictures)

2011 > 2012 > 2013 > 2014 > 2015 > 2016 > **2017**

.....SSFF.S.....S.FF........S..I

Time: 25 minutes

OK (39 slides, 0 cat pictures)

Thank you!

Questions?

Magento Test Automation: My Journey @fschmengler

Links

- `http://xtest-mage.com/`
- `https://magiumlib.com/`
- `http://codeception.com/`

Image sources

- Marvin: The Hitchhiker's Guide to the Galaxy (2005 movie)
- Angry German: https://www.youtube.com/watch?v=-_xUIDRxdmc
- Black metal cat: https://twitter.com/evilbmcats
- Agile testing pyramid: http://www.agilecoachjournal.com