

Tales from the eXtreme side

07.11.2007

Lars Jankowsky

CTO OXID eSales AG

About me:

PHP, C++, Developer, Teamleader since 1992

PHP since 1998

Many successful projects from 2 to 20 developers

Running right now three projects using eXtreme Programming



Communication

Communication

- Teams should`nt have more than 10 members
- It does not work good if team members are in different locations.
- Video conferencing over Skype is not a good idea if you have more than two attendees.

Office Layout

Office Layout

- You have to think about office layout.
- If project is going to be tough, let Team sit closer.
- Buy a huge whiteboard (or two)



Communication



Planning and running Iterations

Planning XP - Iteration

- Do not accept any estimations above 2 SP
- Send SP every evening to one person („Iteration Master“)

Stand Up Meeting

Stand Up Meeting

- Make sure that people stand.
- Some people can stand a long time!
- Do not talk about storypoints in Standup Meeting.
- Let them tell what they did yesterday and what they plan to do today - make sure that they do not use „task numbers“



Communication



Planning and running Iterations



Coding XP

Pair Programming

Pair Programming

- Use pair programming on complex tasks not more than 2 days in a row.
- Team performance decreases, quality goes up.

Refactoring

Refactoring

- Refactoring for the sake of refactoring is not good.
- Refactor only the main problems which causes problems and disturb when adding new functionality.
- Encourage the team to refactor while adding new code.
- Keep the architecture clean and free of hacks.

Bosses pay us money not
refactoring itself but
working code.



Communication



Planning and running Iterations



Coding XP



Testing XP

Big Ball of Mud !?

http://en.wikipedia.org/wiki/Big_ball_of_mud

"I almost finished the task. I implemented this feature, all what's left is to write tests."

Test Driven Development

- Use TDD everywhere!
- When TDD is used, the resulted code is of better quality, more modular, less coupled.
- Writing Tests after coding needs more time.
- Don't forget your javascript code.
 - <https://dev.thefrontside.net/crosscheck>
 - <http://www.jsunit.net>

„We have so much work todo let's skip the tests
until release...“

Testing XP

- Sticking to TDD can be hard when writing Selenium tests in the start of the project.
- Write Selenium Tests for new features before you implement them.
- One Developer write Code/Unit Tests whilst the other is writing Selenium acceptance tests

Testing protected and private methods ?

Testing XP

- Writing tests for public interface only makes refactoring easier.
- There are cases where you must test internal functions (e.g. API's)
- There is no „truth“ - you decide!

Testing XP

```
public function getProxy($superClassName)
{
    $proxyClassName = "{$superClassName}Proxy";

    if (!class_exists($proxyClassName)) {

        $class = <<<CLASS
            class $proxyClassName extends $superClassName
            {
                public function __call(\function, \$args)
                {
                    \$function = str_replace('protected_', '_', \$function);
                    return call_user_func_array(array(&\$this, \$function), \$args);
                }
            }
        CLASS;
        eval($class);
    }
    return new $proxyClassName();
}
```

„We simply skip the Beta test this time as you have so much automated tests.“

Continuous Integration

Continuous Integration

- Use cruise control, bamboo or something similar
- Team should see monitor always.
- Transparency is important - Developer need to see status and bugs.
- Define some punishment for breaking the Build.
- Check daily commits!

Continuous Integration

- Keep build time as short as possible.
 - Feedback should be instant.
 - The more you need to wait for test results the less benefit tests bring to you.
- If tests take to long ?
 - Tweak the tests.
 - Group the tests.
 - Cover more cases in unit tests, then you will need less cases in Selenium tests.

„If CC will succeed completely, I tell you , girl + boys , we will lay back and do no more commit " "

Continuous Testing ?

- Use tools to assure Quality.
 - PHPUnit PMD
 - CodeSniffer
 - Metrics (Cyclomatic complexity, NPath)
 - xDebug`s code coverage
 - ...

Code Coverage ?

- Only on a per function base.
- Coverage > 95% ?
 - It does not mean that you have good tests.

**Metrics can create a false
feeling of safety!**

Agile ?

THINK !

Questions ?

Contact information

Lars Jankowsky

CTO

OXID eSales AG

www.oxid-esales.com

E-Mail: lars.jankowsky@oxid-esales.com

Fon: +49 761 36889 0