

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/235945653>

# PHP Application Architecture

Conference Paper · February 2013

CITATIONS

0

READS

4,786

1 author:



[Martin Holzhauer](#)

Mayflower

3 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

# PHP Application Architecture

*And what we can do wrong*

# It'se Me

Martin Holzhauer

woodworker@twitter

woodworker@github

Working at

**ResearchGate**  
SCIENTIFIC NETWORK



Doing PHP Stuff since 2002 (PHP 4.1.2)

# What this Talk features

- basics behind application architecture
- what i've learned in the last years
- ideas you can apply to your work
- maybe good content

# What this Talk is NOT

- practical advice to change everything now
- the only truth
- a reason to stop think about architecture yourself
- full of fancy pictures and effects

# Architecture

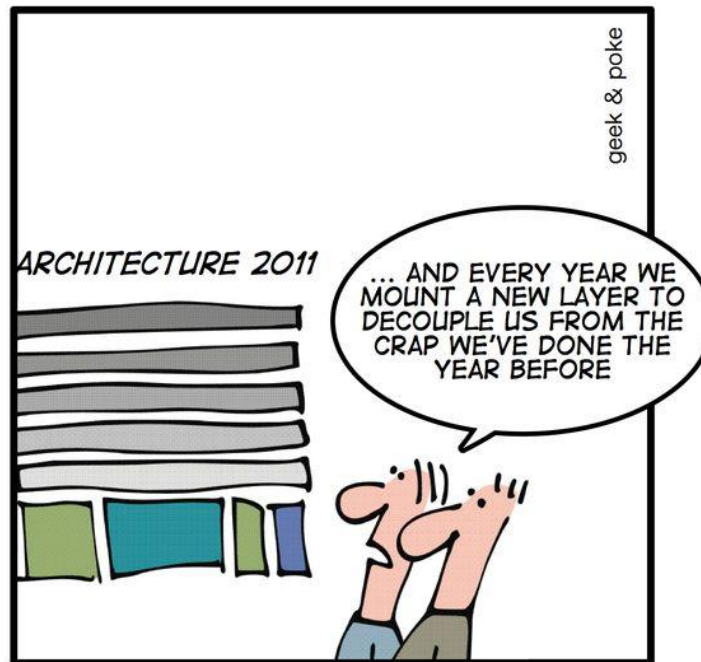
*The almighty Master Plan!?*

# We all do Architecture !?

*At least we hope so.*

# ***BEST PRACTICES IN APPLICATION ARCHITECTURE***

*TODAY: USE LAYERS TO DECOUPLE*



*ANNUAL RINGS*

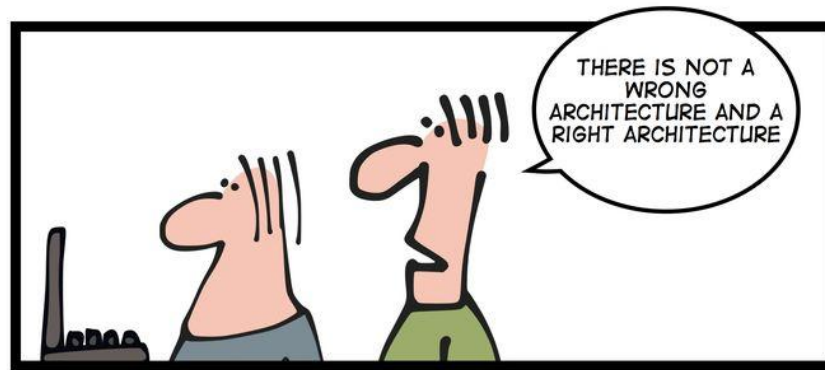


# Using Patterns is not having an Architecture

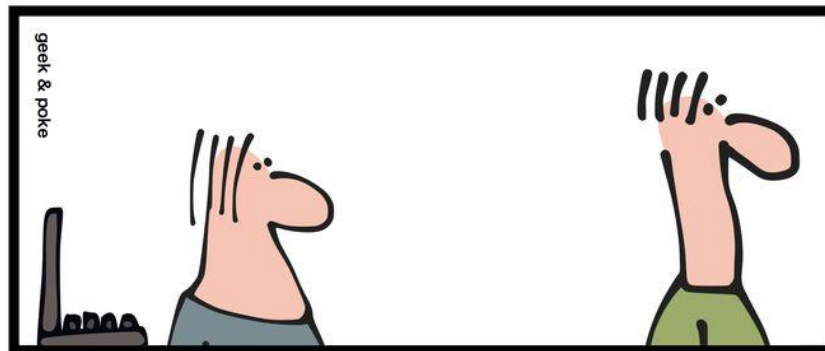
*"We do have an Architecture, our Application is MVC"*

# Using Patterns is not having an Architecture

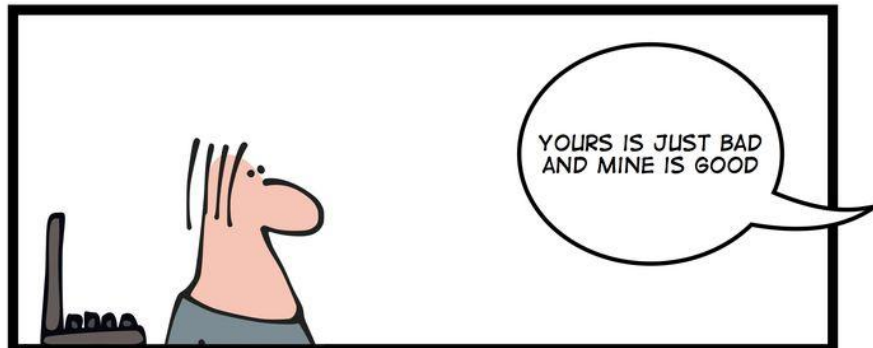
*Frameworks, Programming Languages or Patterns  
are only the tools to implement your Architecture*



IT ARCHITECTURE IS NOT ALWAYS SIMPLE



FORTUNATELY...



... MOST OF THE TIME IT IS

And what is Architecture?



Architecture is the  
definition of your use cases

*If you want a short business punchline*

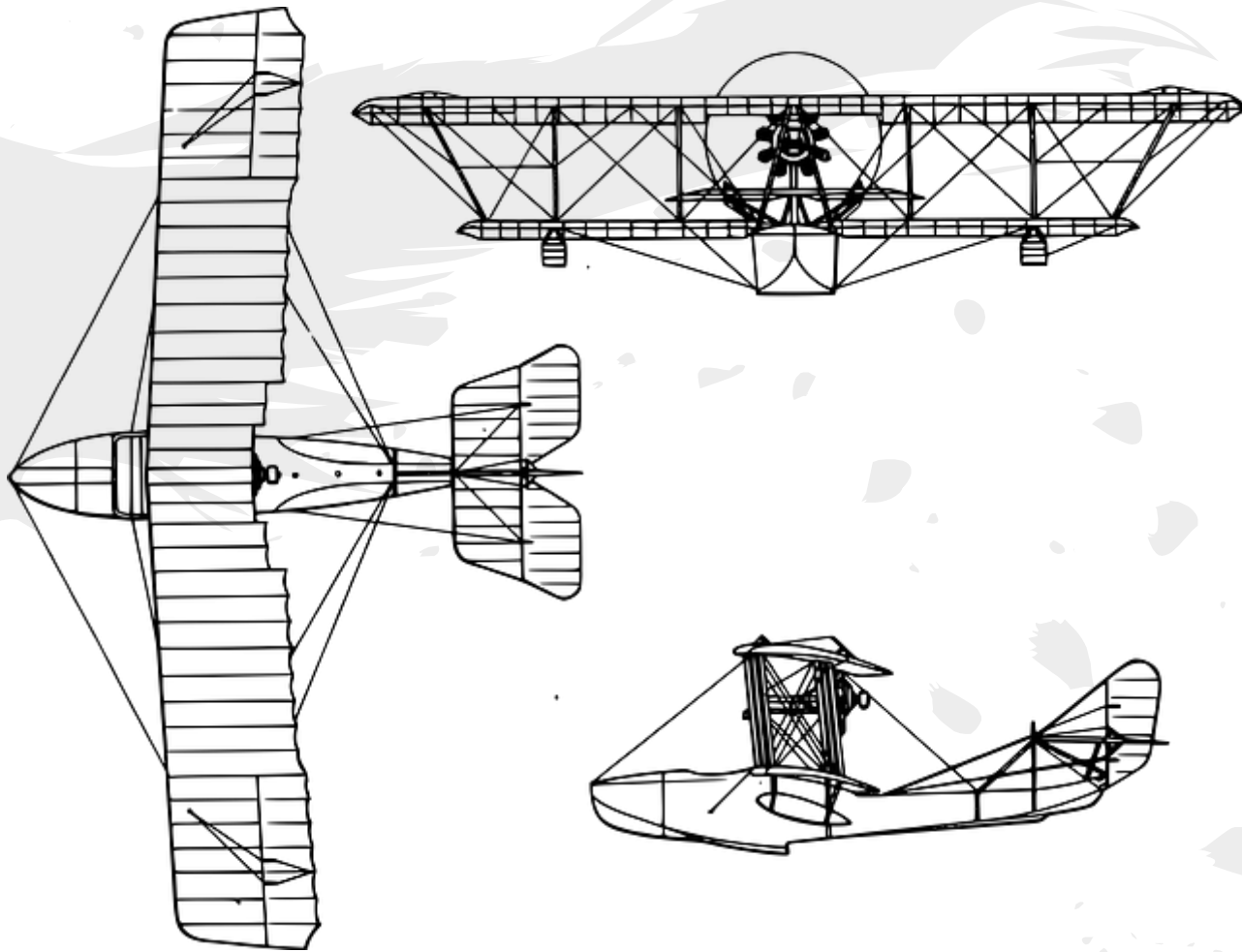
Architecture is the  
definition of your business

*or at least what we developer think is the definition*

# Programming is Architecture

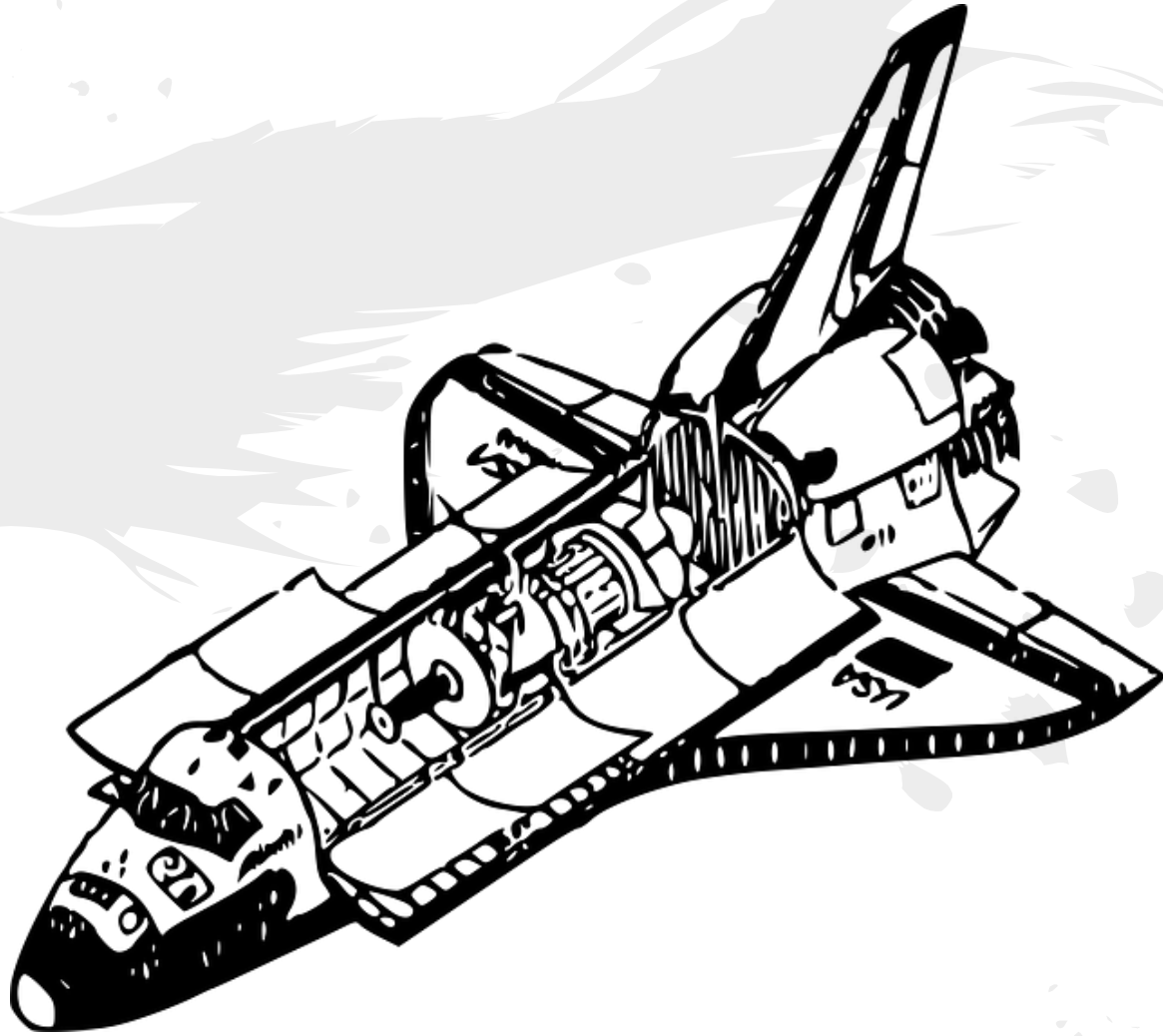
*When you think about your functions, classes, objects  
or methods you are doing Architectural decisions.*

# Find your use cases





# Define your use cases



# Architecture and Use cases

When you look at your project:

- do you see the framework?
- do you see mvc?
- do you see web?
- do you see database?

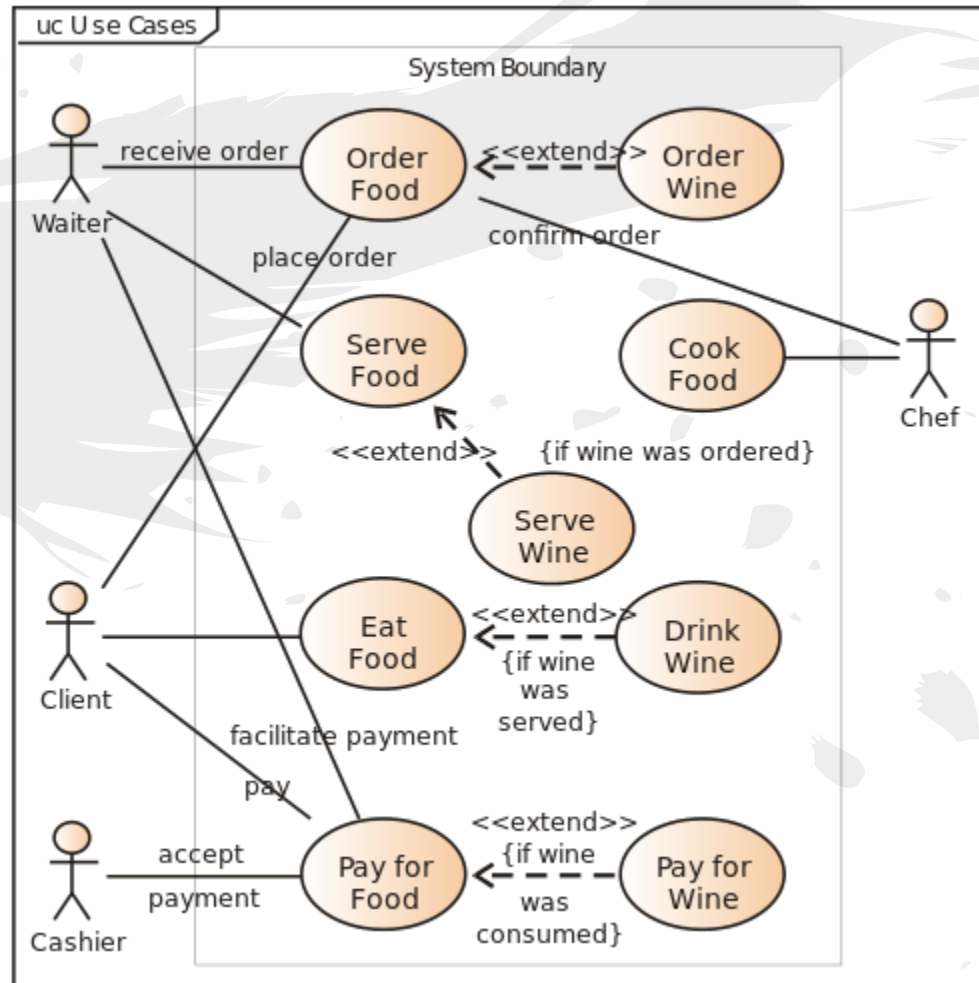


# Architecture and Use cases

**or**

**do you see the use cases**

# Real Life vs. Programming

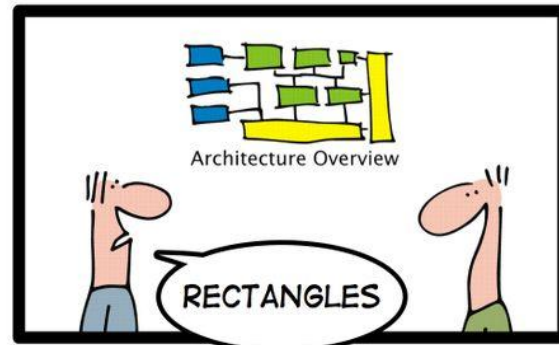
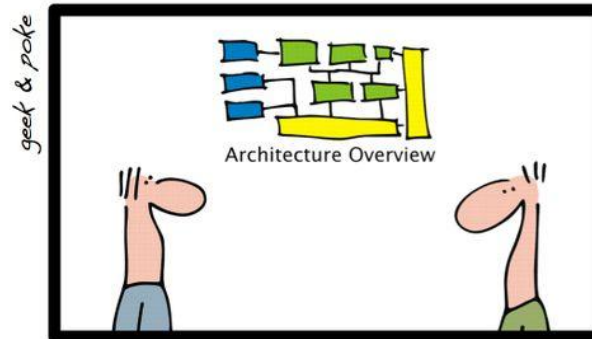
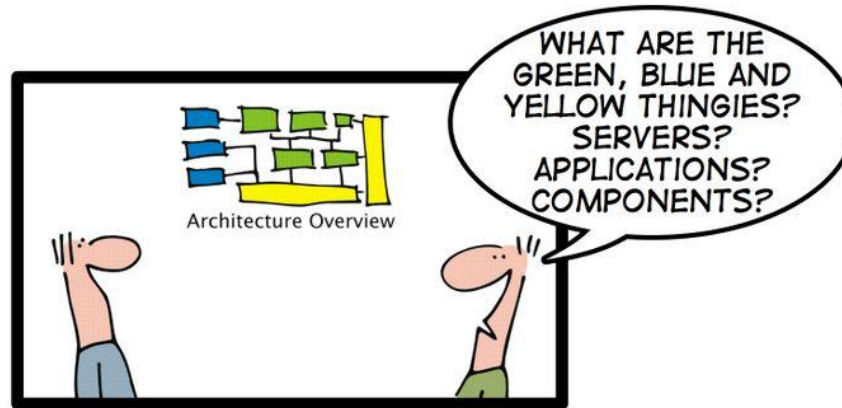


# Use cases

you could do UML Case diagrams

but don't be silly ٩(•̀•́)۶

## ENTERPRISE ARCHITECTURE MADE EASY




PART 1: DON'T MESS WITH THE  
GORY DETAILS

# Use cases

My opinion is that  
"User adds a Place"  
on an index card is good enough

Your use cases  
will disclose your API

The image features a horizontal split. The top half is a solid dark grey color. The bottom half is white, with a jagged, torn-paper-like border separating it from the grey section. Scattered across the white area are several small, irregular grey shapes, resembling ink splatters or debris from the torn paper.



# Use case driven development

- There is **NO** "one Interface fits all"
- Specific methods for specific use cases
- Do not try to abstract you interface to much
- As always, you need to find good compromises

# Implement use cases, not technologies

*decouple your system from the underlying technologies*

# Implement use cases and hide technologies

- Do not use ORM/ODM Entities in your public API
- Do not use Framework specific Interfaces in your public API
- Do not any external, third party stuff in YOUR public API

# Implement use cases and hide technologies

- `selectFromDb($field, $table)` Bad

- `getById($id)` Good

# Implement use cases and hide technologies

- `search('text', Search::Type_Keyword)` OKisch
- `searchByKeyword('text')` Better

# Implement use cases

*Your models and interfaces represent your business  
not your database*

# The cool thing about that

You can switch your framework/tools

- as in we are porting our app-frontend to Symfony2
- we are moving to \$coolNoSQLDB
- we decided to hate zend\_db
- we have written a optimized service for \$usecase in erlang

# The cool thing about that

You can simply write another UI

- we need a mobile site
- we want an Android/iPhone app
- we want an API



A good architecture will let  
you grow

*Or shrink or change*

# An Example

*Or better a view on a small part of your business*

# Architecture and growth

## An Example (Search)

Iteration 1

Usage:

```
$placeService->search('Berlin')
```

Implementation:

LIKE query

# Architecture and growth

## An Example (Search)

Iteration 1

Usage:

```
$placeService->searchByCity('Berlin')
```

Implementation:

LIKE query

# Architecture and growth

## An Example (Search)

Iteration 2

Usage:

```
$placeService->searchByCity('Berlin')
```

Implementation:

Fulltext index

# Architecture and growth

## An Example (Search)

Iteration 3

Usage:

```
$placeService->searchByCity('Berlin')
```

Implementation:

Solr

# Architecture and growth

## An Example (Search)

Iteration 4

Usage:

```
$placeService->searchByCity('Berlin')
```

Implementation:

Solr + Database

# Architecture and growth

## An Example (Search)

Iteration 5

Usage:

```
$placeService->searchByCity('Berlin')
```

Implementation:

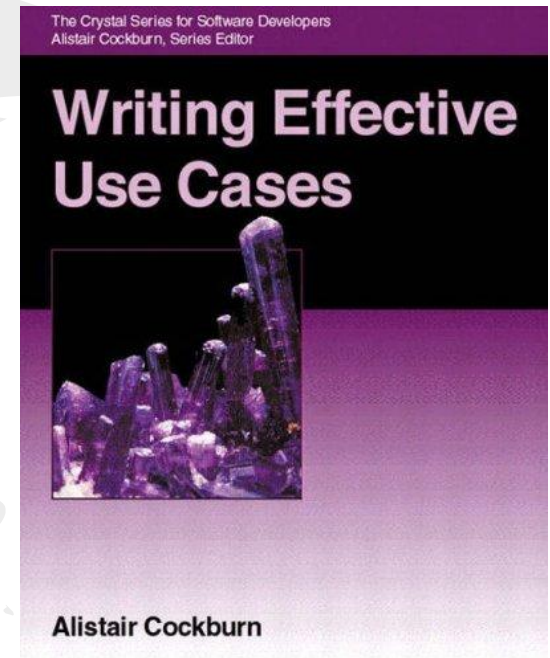
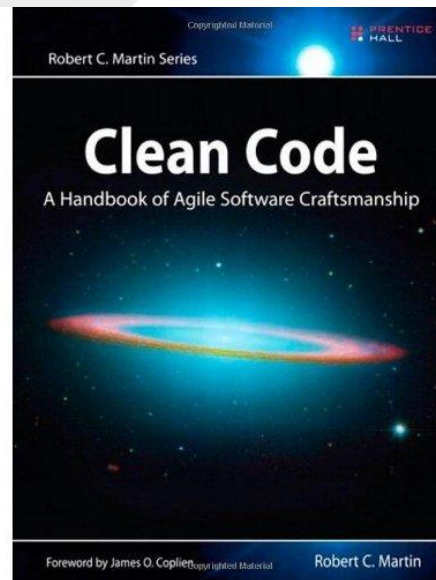
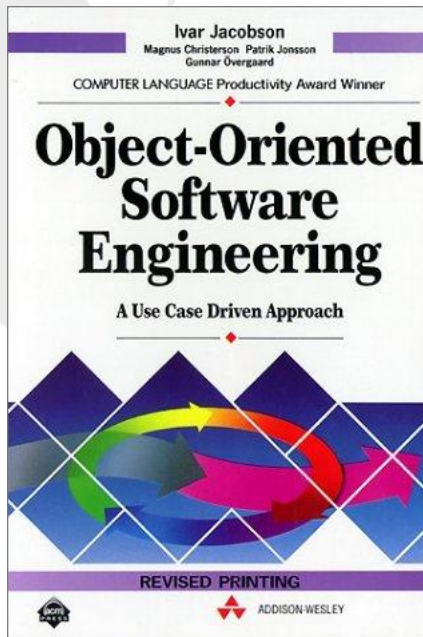
Solr + Memcache + DB



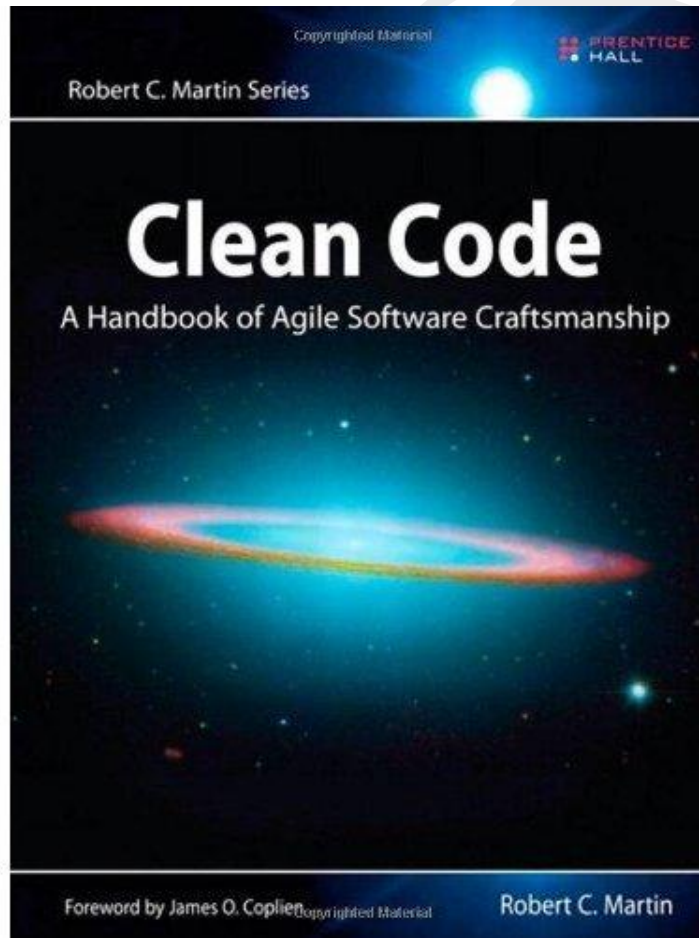
# My Point

*You care about the use cases  
You do not give a f\*\*k about implementation*

# More Informations



# General Informations



# Questions?

*Maybe i will have answers  
or can confuse you a bit more ;D*

# KTHXBYE

*@woodworker*  
*[martin@holzhauer.eu](mailto:martin@holzhauer.eu)*