Automating Architecture Constraint Checks

International PHP Conference

Tobias Schlitt, Kore Nordmann

26th October 2016

Hi

Tobias Schlitt
@tobySen

Kore Nordmann
@koredn

@qafoo

@tideways

Qafoo
passion for software quality

Coding Style

Is This Enough?

Is this enough?

Common patterns exist inherently in every project.

Definition Of Done

Defining Patterns

- Coming out of a legacy code base
- Writing a new component which is supposed to stay
- Shared understanding of code structure

Defining a **vision** for a new software or refactorings


Common Design Rules

Common Design Rules

- No logic in Controller
- Always exceptions for error handling
- Use data objects instead of arrays
- Use Gateways / Repositories to load and save data
- Service dependencies must always be injected
- Hide externals behind facades
- Test all domain logic
- Do not abbreviate

Customer Scenario: External Modules

- Vendor for a software product based on PHP
- Product used / customized by development teams of
  - various backgrounds
  - various skill levels
- How can we . . .
  - . . . ensure coding concepts are applied like meant to be?
  - . . . enforce certain constraints on the architecture?

Customer Scenario: By Example

- Constructor typehints to abstractions
- Facades must not return hidden types
- Only factories may access the Dependency Injection Container
- The domain must not access the controller layer
- The front-end must not access the database but only call web services on the back-end


Hinders Creativity?

Benefits

- Design Patterns are hard to discover (spread accross files)
- Teaching and learning is purely a team effort right now
  - Pair Programming
  - Code Reviews
- New developers are faster


Stability Is A Benefit

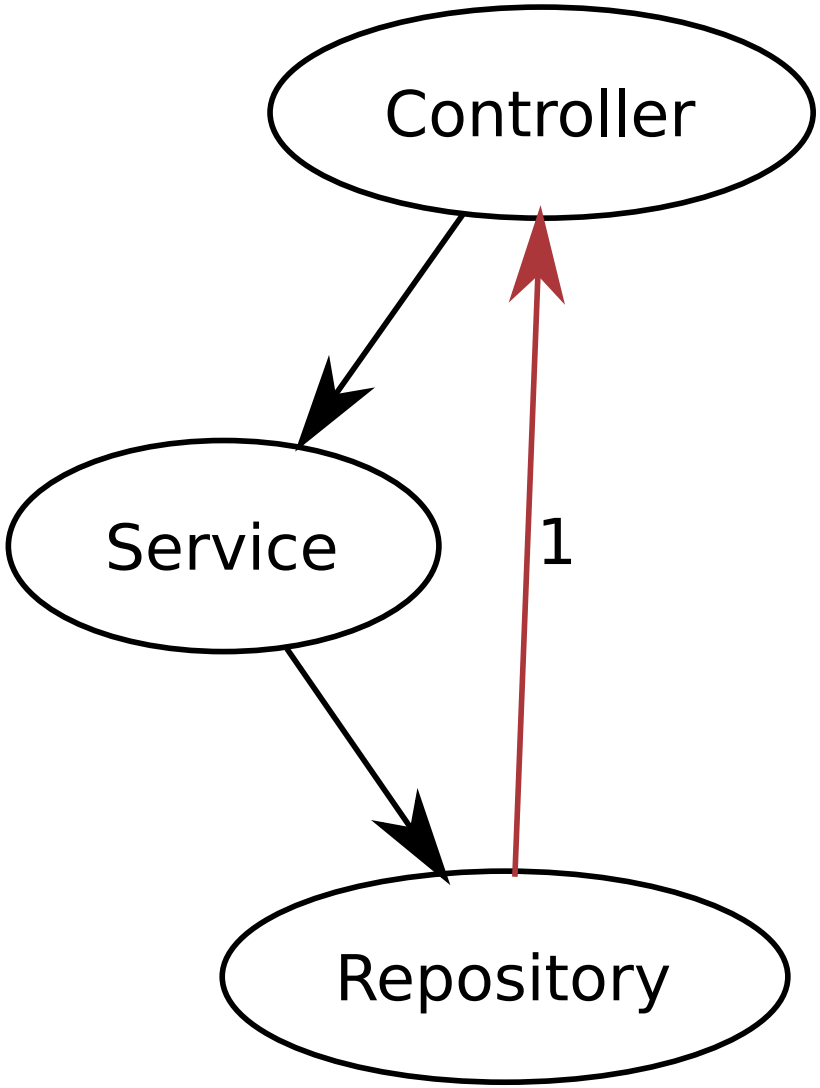
The PHP Tool Stack Gives us . . .

- Check for appliance of coding style
  `PHP_CodeSniffer`
- Assert basic code metrics
  `PHPLOC`
- Alert on code copy & paste `PHP Copy/Paste Detector`
- Assert complexity & coupling metrics
  `PHP_Depend`
- Alert on common code design smells
  `PHP Mess Detector (PHPMD)`
- Assertions on dependency graph
  `Deptrac`

Deptrac

```yaml
paths:
  - ./src
exclude_files:
  - .*test.*
layers:
  - name: Controller
    collectors:
      - type: className
        regex: .*Controller.*
  - name: Repository
    collectors:
      - type: className
        regex: .*Repository.*
  - name: Service
    collectors:
      - type: className
        regex: .*Service.*
ruleset:
  Controller:
    - Service
  Service:
    - Repository
  Repository: ~
```

Deptrac



PHP Mess Detector

- Evaluates rules on code entities (e.g. classes, methods, . . .)
- Uses an Abstract Syntax Tree (AST)
- Pre-build rule sets to indentify
  - Possible bugs
  - Suboptimal code
  - Overcomplicated expressions
  - Unused parameters, methods, properties
- `https://phpmd.org/`
- Works on basis of PHP_Depend (`https://pdepend.org/`)

Custom PHPMD: DataObjects do not Contain Methods

```
1  class DataObjectContainsNoMethod extends \PHPMD\AbstractRule
2      implements \PHPMD\Rule\ClassAware
3  {
4      public function apply(\PHPMD\AbstractNode $node)
5      {
6      }
7  }
```

Qafoo
passion for software quality

Custom PHPMD: DataObjects do not Contain Methods

```
1  class DataObjectContainsNoMethod extends \PHPMD\AbstractRule
2      implements \PHPMD\Rule\ClassAware
3  {
4      public function apply(\PHPMD\AbstractNode $node)
5      {
6          /* @var $node \PHPMD\Node\ClassNode */
7          $type = $node;
8          do {
9              $type = $type->getParentClass();
10             if (isset($type) && $type->getName() === 'DataObject') {
11                 $this->check($node);
12             }
13         } while ($type);
14     }
15 }
```

Custom PHPMD: DataObjects do not Contain Methods

```
1   class DataObjectContainsNoMethod extends \PHPMD\AbstractRule
2       implements \PHPMD\Rule\ClassAware
3   {
4       public function apply(\PHPMD\AbstractNode $node)
5       {
6                   $this->check($node);
7       }
8       protected function check(\PHPMD\Node\ClassNode $node)
9       {
10          $methods = $node->getMethods();
11
12          foreach ($methods as $method) {
13              $this->addViolation($method, array($method->getName()));
14          }
15      }
16  }
```

Qafoo
passion for software quality

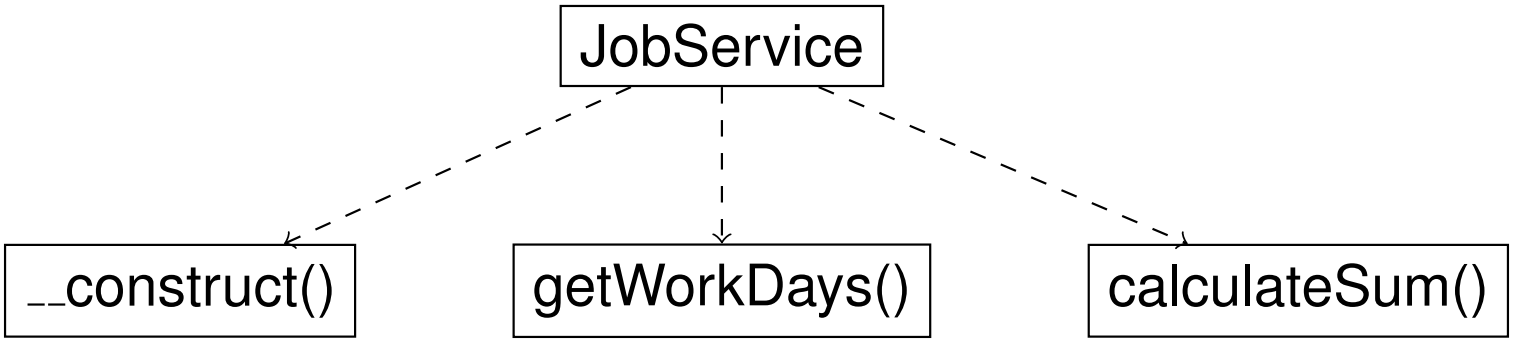
Custom PHPMD: DataObjects do not Contain Methods

```
1  <?xml version="1.0"?>
2  <ruleset name="Qafoo TimePlanner Rule Set"
3           xmlns="http://pmd.sf.net/ruleset/1.0.0"
4           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5           xsi:schemaLocation="http://pmd.sf.net/ruleset/1.0.0
6  ____________________http://pmd.sf.net/ruleset_xml_schema.xsd"
7           xsi:noNamespaceSchemaLocation="
8  ____________________http://pmd.sf.net/ruleset_xml_schema.xsd">
9  </ruleset>
```

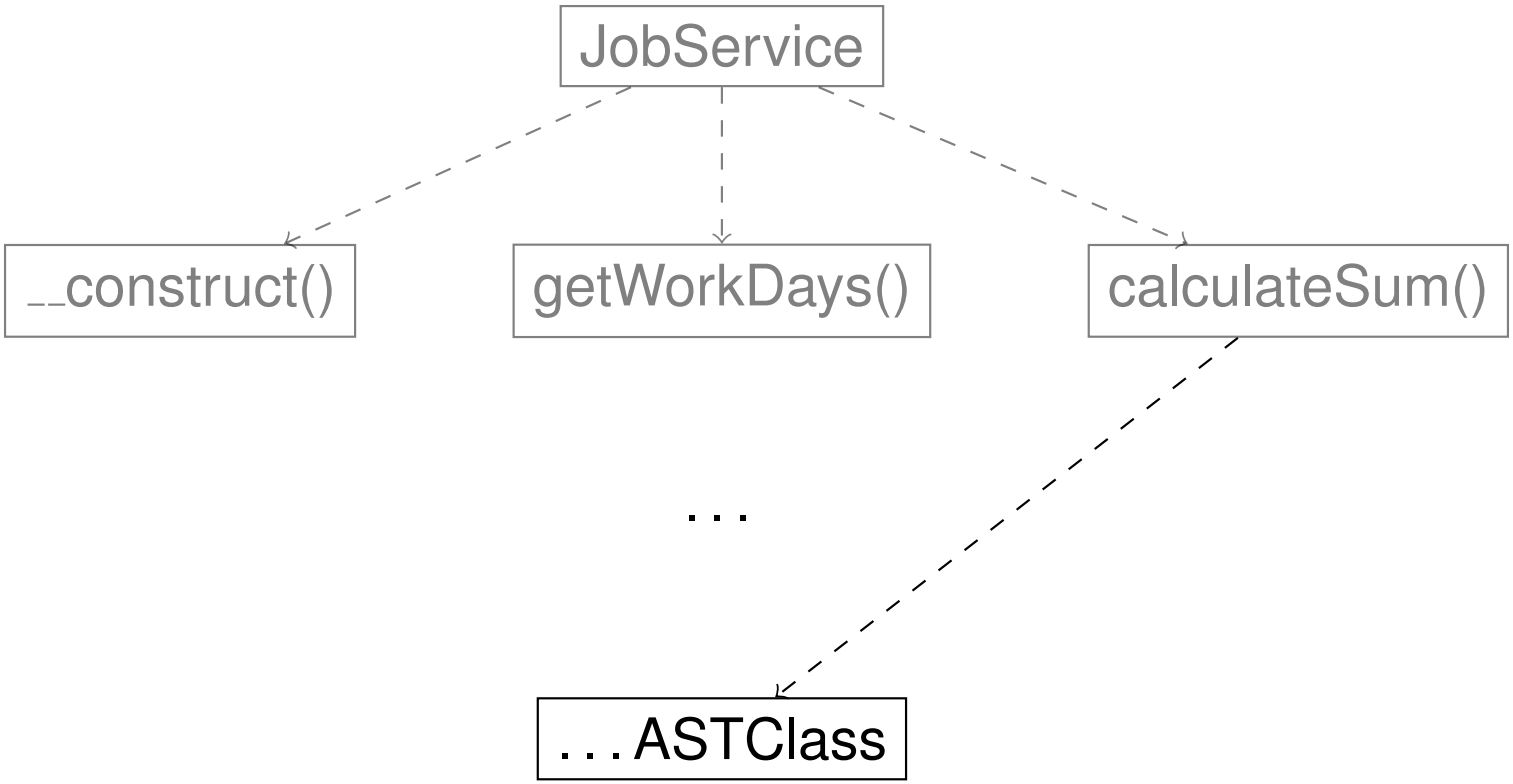
Custom PHPMD: DataObjects do not Contain Methods

```
<?xml version="1.0"?>
<ruleset name="Qafoo TimePlanner Rule Set"
    <description>
        Asserting example architecture guidelines in the Qafoo TimePlanner app.
    </description>
    <rule
            name="DataObjectContainsNoMethod"
            message="Data Objects are not meant to contain methods: '{0}'"
            class="Qafoo\ArchitectureRules\DataObjectContainsNoMethod"
    />
</ruleset>
```

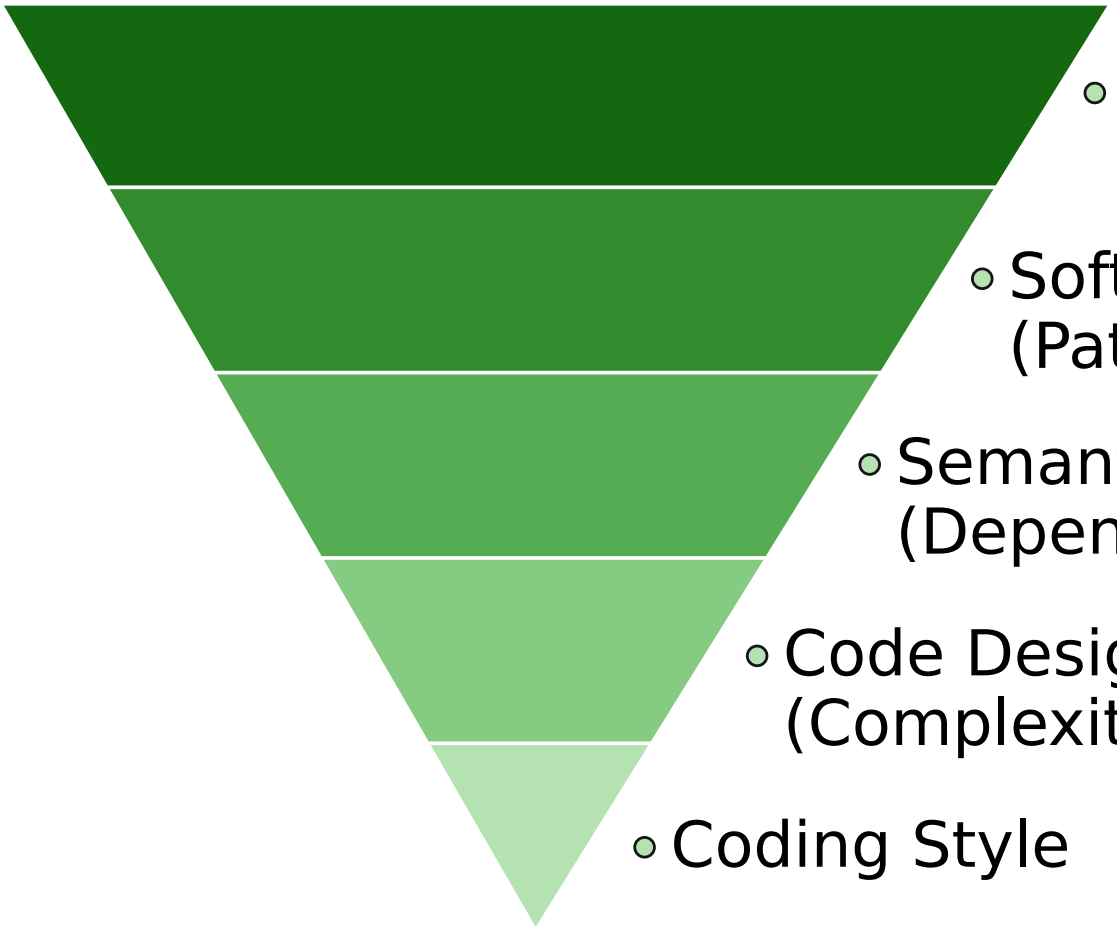
High Level AST (PHPMD)



Low Level AST (PDepend)



From Micro To Macro



- Architecture Documentation
- Software Design (Patterns)
- Semantic Checks (Dependencies)
- Code Design Checks (Complexity, Test Coverage)
- Coding Style

Summary

- Agree upon common patterns
  - These are team and project specific
- Assist code reviews by automated violation discovery
- Assert on external code quality beyond Coding Style
- Speed up your development

THANK YOU

Rent a quality expert
qafoo.com