

# Code Inspection

## Software Engineering 2 - Project

Alberto Maria Metelli   Riccardo Mologni

Politecnico di Milano  
M. Sc. in Computer Science and Engineering

Code Inspection Presentation, 7th January 2016

# Outline

- 1 Introduction
  - Our code inspection
- 2 Our assignment
  - Class and methods
  - Functional role
- 3 Issues
  - Some example of issues
- 4 Tables and charts
  - Issues per category
  - Issues per method/class

# Outline

- 1 Introduction
  - Our code inspection
- 2 Our assignment
  - Class and methods
  - Functional role
- 3 Issues
  - Some example of issues
- 4 Tables and charts
  - Issues per category
  - Issues per method/class

# Our code inspection

- Manual inspection
  - Checklist
  - Other problems
- Automatic code review
  - SonarQube
  - PMD
  - FindBugs

Different methods and tools also to compare the different sensibility towards code style.

# Outline

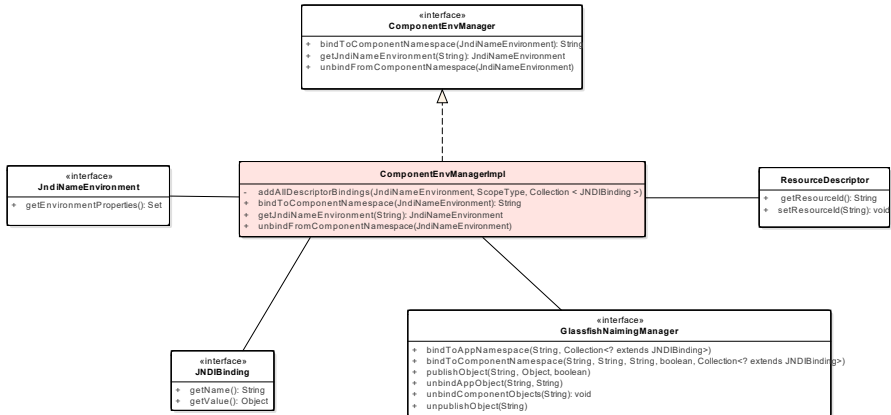
- 1 Introduction
  - Our code inspection
- 2 Our assignment
  - Class and methods
  - Functional role
- 3 Issues
  - Some example of issues
- 4 Tables and charts
  - Issues per category
  - Issues per method/class

# Class and methods

```
public class ComponentEnvManagerImpl implements  
ComponentEnvManager
```

- `public JndiNameEnvironment getJndiNameEnvironment(String componentId)`
- `public String bindToComponentNamespace(JndiNameEnvironment env) throws NamingException`
- `private void addAllDescriptorBindings  
addAllDescriptorBindings(JndiNameEnvironment env , ScopeType  
scope , Collection < JNDIBinding > jndiBindings)`
- `public void unbindFromComponentNamespace(JndiNameEnvironment  
env) throws NamingException`

# Class and methods



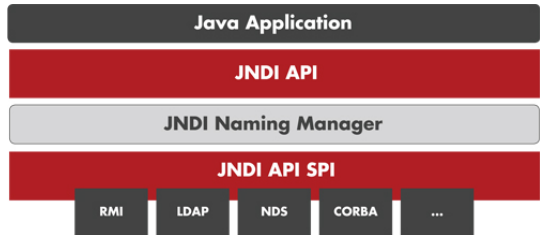
# Outline

- 1 Introduction
  - Our code inspection
- 2 Our assignment
  - Class and methods
  - Functional role
- 3 Issues
  - Some example of issues
- 4 Tables and charts
  - Issues per category
  - Issues per method/class



# Understanding functional role...

- Few comments
- Missing JavaDoc



## Briefly about JNDI environments

- Each resource object is identified by a *JNDI name*, bound on the *naming server*.
- Web components must have access to a *JNDI naming environment*.
- The *application component's naming environment* allows customization of the application component's business logic during deployment or assembly without need of changing code.

# Functional role of methods

- `getJndiNameEnvironment` returns the JNDI environment associated to the component whose name is passed as parameter
- `bindToComponentNamespace` publishes in the naming server the name of the JNDI environment passed as parameter
- `addAllDescriptorBindings` converts the resource descriptors associated to the JNDI environment to JNDI bindings
- `unbindFromComponentNamespace` reverse function with respect to `bindToComponentNamespace`

# Outline

- 1 Introduction
  - Our code inspection
- 2 Our assignment
  - Class and methods
  - Functional role
- 3 **Issues**
  - Some example of issues
- 4 Tables and charts
  - Issues per category
  - Issues per method/class

# Example 1

<b>#Issue</b>	<b>1</b>
<i>Class/Method</i>	ComponentEnvManagerImpl/getJndiNameEnvironment
<b>#Line</b>	161
<i>Code fragment</i>	RefCountJndiNameEnvironment rj
<i>Issue category</i>	Naming conventions
<i>Issue ref</i>	1
<i>Motivation</i>	Method variable name rj non meaningful.
<i>Comment</i>	Variable name should refer to the object referenced.

## Example 2

<b>#Issue</b>	<b>39</b>
<i>Class/Method</i>	ComponentEnvManagerImpl
<b>#Line</b>	106
<i>Code fragment</i>	GlassfishNamingManager namingManager
<i>Issue category</i>	Initialization and Declarations
<i>Issue ref</i>	28
<i>Motivation</i>	Friendly instance variable not used by same package classes, should be private.
<i>Comment</i>	Keeping the lowest visibility as possible is preferred for information hiding.

## Example 3 - usage of continue

```
for (ResourceDescriptor descriptor : allDescriptors) {  
    if (!dependencyAppliesToScope(descriptor, scope)) {  
        continue;  
    }  
  
    if (descriptor.getResourceType().equals(DSD)) {  
        if (descriptor.isDeployed()) {  
            continue;  
        }  
    }  
  
    [...]  
}
```

## Example 3 - usage of continue

```
for (ResourceDescriptor descriptor : allDescriptors) {  
    if (dependencyAppliesToScope(descriptor, scope) &&  
        !(descriptor.getResourceType().equals(DSD)  
            &&  
            descriptor.isDeployed())) {  
        [...]  
    }  
}
```



## Summarized data

- Number of issues: 64
- Number of issues for KSLOC: 215,1
- No serious bugs, mainly *stylistic* issues and missing comments
  - Most of them are in category *File Organization* and *Comments*

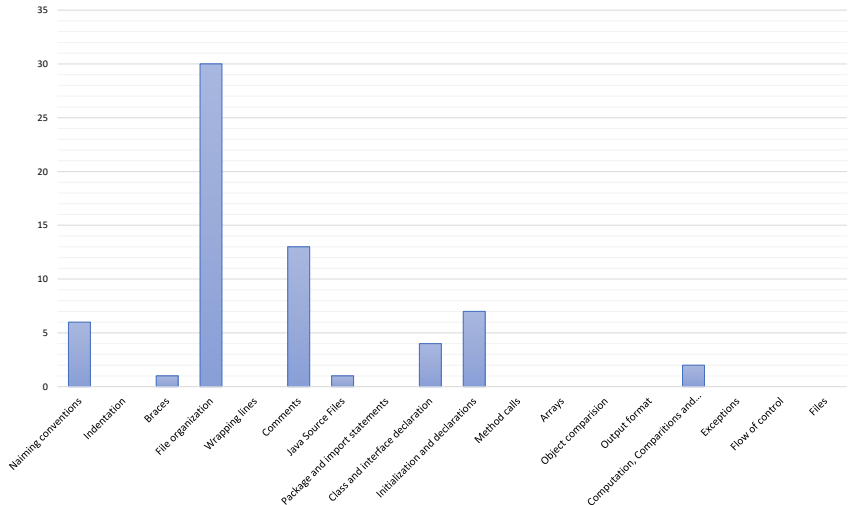
# Outline

- 1 Introduction
  - Our code inspection
- 2 Our assignment
  - Class and methods
  - Functional role
- 3 Issues
  - Some example of issues
- 4 Tables and charts
  - Issues per category
  - Issues per method/class

# Issue per category - Table

Issue category	Naming conventions	Braces	File organization	Comments	Java Source Files	Class and interface declaration	Initialization and declarations	Computation, Comparisons and Assignments
<i>Total per category</i>	6	1	30	13	1	4	7	2
<i>Total per category (%)</i>	9,38	1,6	46,9	20,3	1,6	6,3	10,9	3,1

## Issue per category - Chart



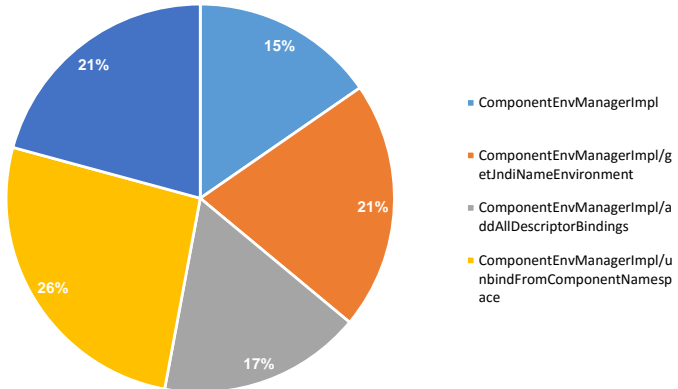
# Outline

- 1 Introduction
  - Our code inspection
- 2 Our assignment
  - Class and methods
  - Functional role
- 3 Issues
  - Some example of issues
- 4 Tables and charts
  - Issues per category
  - Issues per method/class

## Issue per method/class - Table

<b>Class/Method</b>	<b>SLOC</b>	<i>Total for class/method</i>	<i>Total per class/method per KSLOC</i>
ComponentEnvManagerImpl	121	20	165,3
ComponentEnvManagerImpl/ getJndiNameEnvironment	9	2	222,2
ComponentEnvManagerImpl/ addAllDescriptorBindings	55	10	181,8
ComponentEnvManagerImpl/ unbindFromComponentNamespace	46	13	282,6
ComponentEnvManagerImpl/ bindToComponentNamespace	85	19	223,5
<i>Total</i>	316	64	215,1

## Issue per method/class - Chart



# References

- Software Engineering 2 course slides
- Glassfish reference <http://glassfish.pompe1.me/>
- Oracle documentation



# Questions

