Code review - how static analysis saves the work

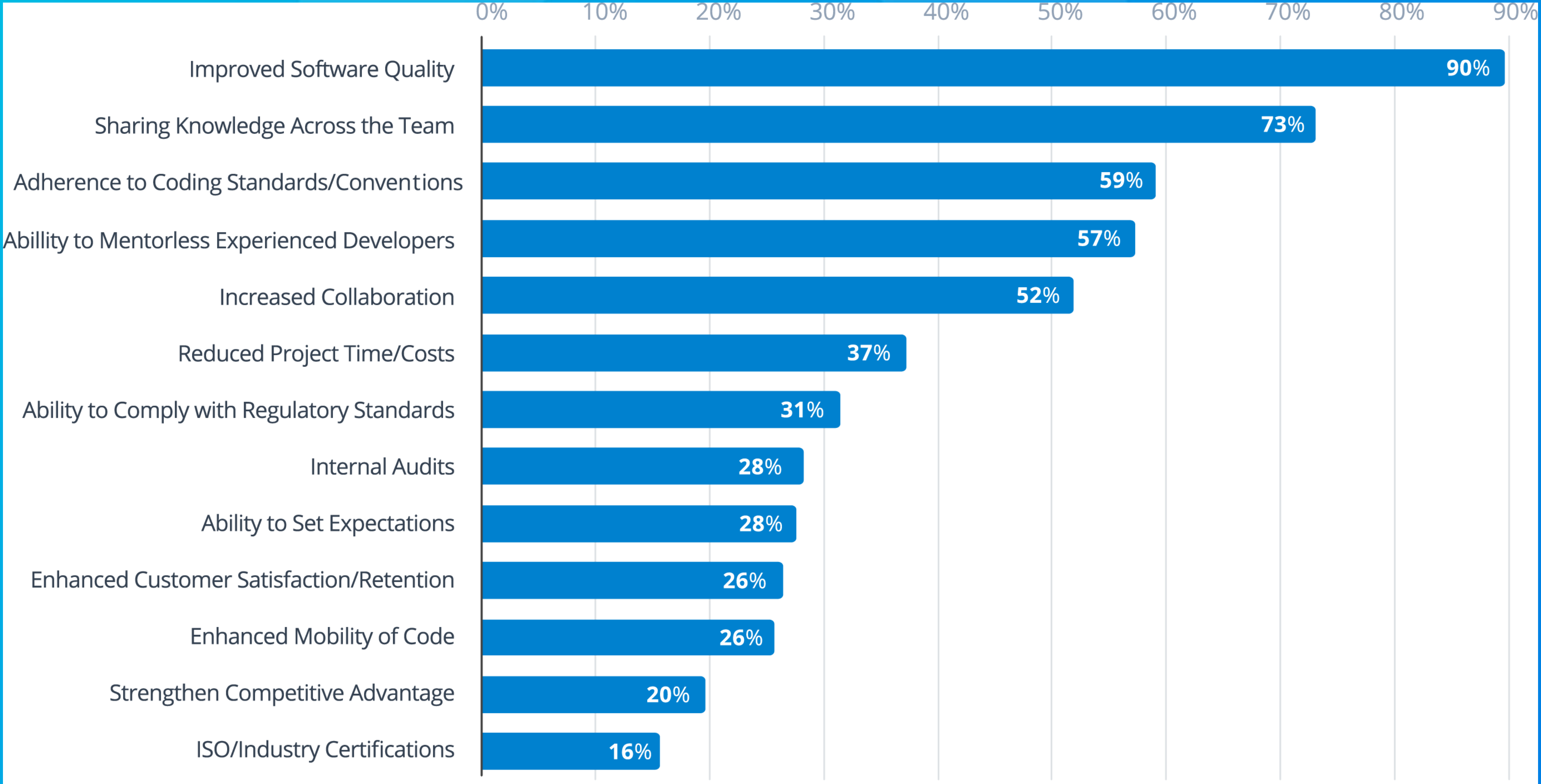
Mateusz Winnicki

www.mateuszwinnicki.pl
mateusz.winnicki@euvic.pl

Code review

Goals

- Defect-free, well-documented software
- Ensuring coding standards
- Knowledge sharing
- Teaching
- Improving quality



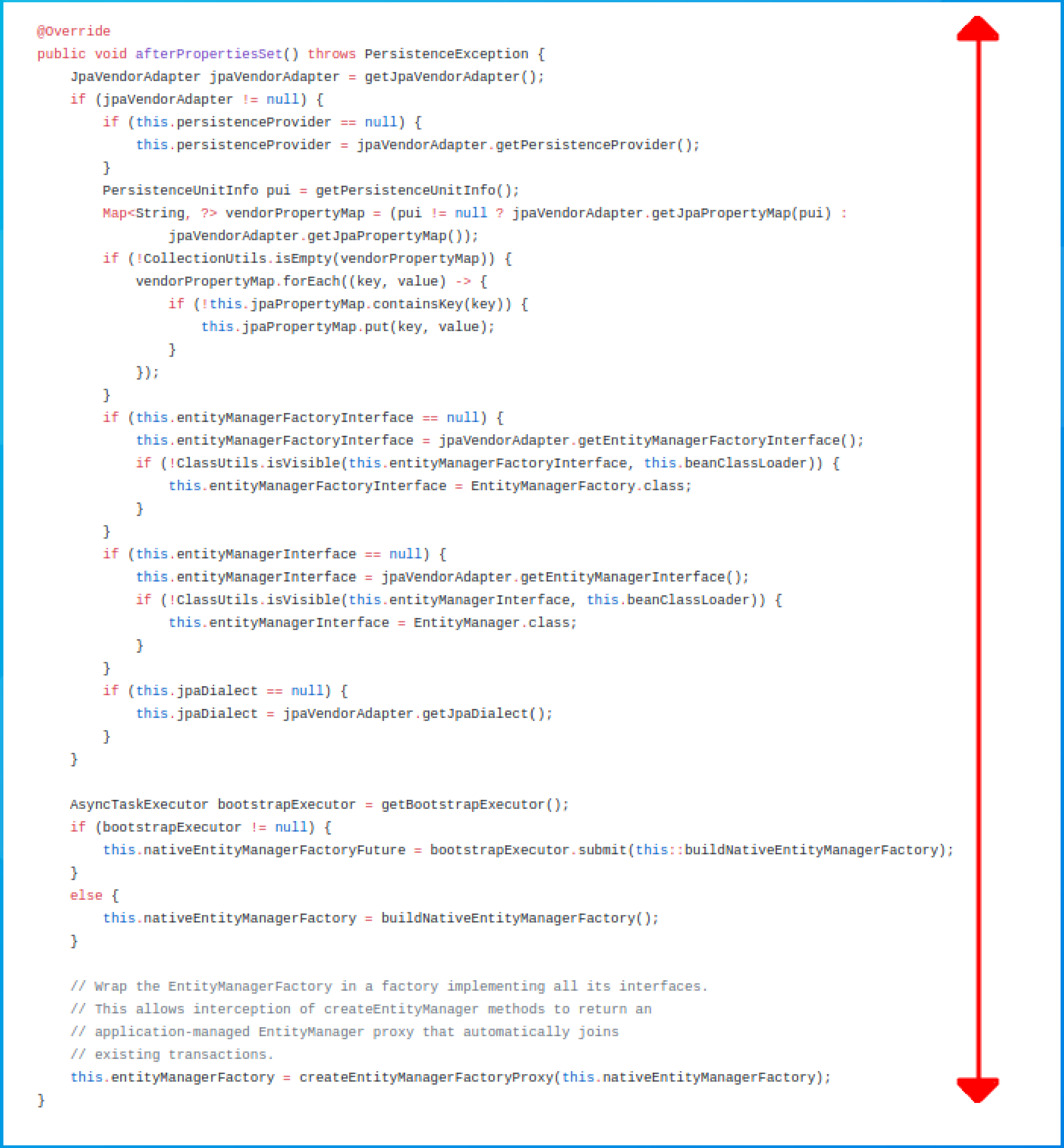
2018 State of Code Review report
https://smartbear.com

CR – some best practices

- 300 LOC at time
- 30 - 45 minutes
- use checklist
- blame the code, not an author
- annotate your code before review (for author)

But...

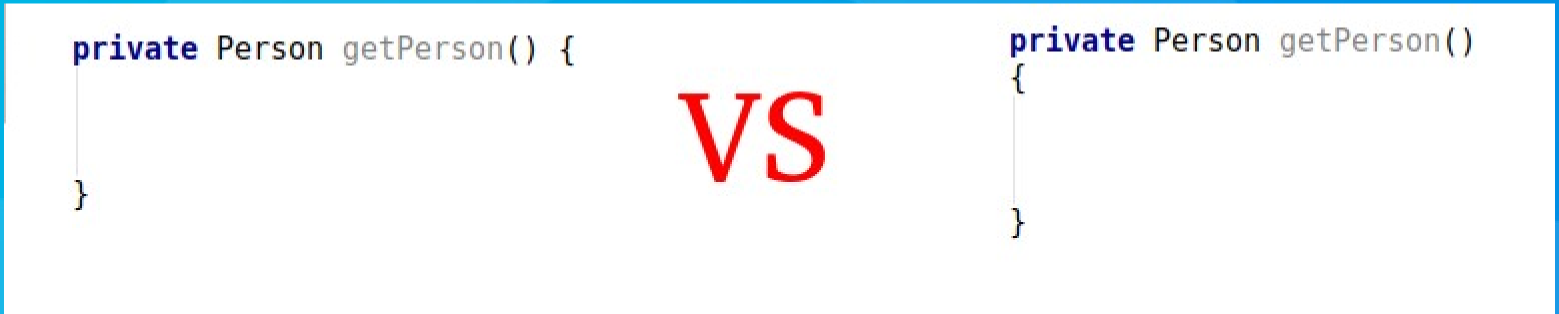
...how many times we were arguing about irrelevant stuff?



```
@Override
public void afterPropertiesSet() throws PersistenceException {
    JpaVendorAdapter jpaVendorAdapter = getJpaVendorAdapter();
    if (jpaVendorAdapter != null) {
        if (this.persistenceProvider == null) {
            this.persistenceProvider = jpaVendorAdapter.getPersistenceProvider();
        }
        PersistenceUnitInfo pui = getPersistenceUnitInfo();
        Map<String, ?> vendorPropertyMap = (pui != null ? jpaVendorAdapter.getJpaPropertyMap(pui) :
                jpaVendorAdapter.getJpaPropertyMap());
        if (!CollectionUtils.isEmpty(vendorPropertyMap)) {
            vendorPropertyMap.forEach((key, value) -> {
                if (!this.jpaPropertyMap.containsKey(key)) {
                    this.jpaPropertyMap.put(key, value);
                }
            });
        }
        if (this.entityManagerFactoryInterface == null) {
            this.entityManagerFactoryInterface = jpaVendorAdapter.getEntityManagerFactoryInterface();
            if (!ClassUtils.isVisible(this.entityManagerFactoryInterface, this.beanClassLoader)) {
                this.entityManagerFactoryInterface = EntityManagerFactory.class;
            }
        }
        if (this.entityManagerInterface == null) {
            this.entityManagerInterface = jpaVendorAdapter.getEntityManagerInterface();
            if (!ClassUtils.isVisible(this.entityManagerInterface, this.beanClassLoader)) {
                this.entityManagerInterface = EntityManager.class;
            }
        }
        if (this.jpaDialect == null) {
            this.jpaDialect = jpaVendorAdapter.getJpaDialect();
        }
    }

    AsyncTaskExecutor bootstrapExecutor = getBootstrapExecutor();
    if (bootstrapExecutor != null) {
        this.nativeEntityManagerFactoryFuture = bootstrapExecutor.submit(this::buildNativeEntityManagerFactory);
    }
    else {
        this.nativeEntityManagerFactory = buildNativeEntityManagerFactory();
    }

    // Wrap the EntityManagerFactory in a factory implementing all its interfaces.
    // This allows interception of createEntityManager methods to return an
    // application-managed EntityManager proxy that automatically joins
    // existing transactions.
    this.entityManagerFactory = createEntityManagerFactoryProxy(this.nativeEntityManagerFactory);
}
```



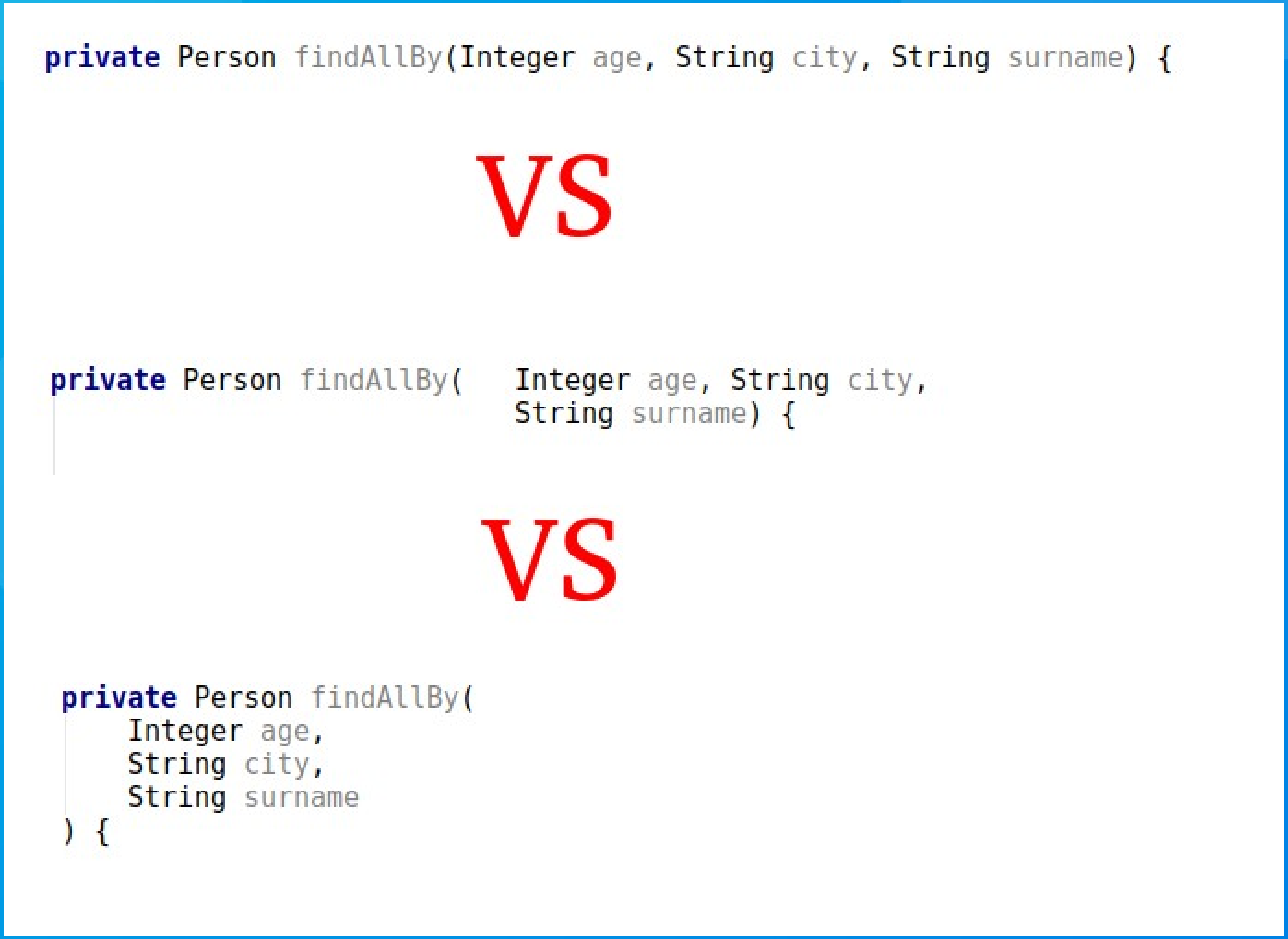
```java
private Person getPerson() {

}
```

VS

```java
private Person getPerson()
{

}
```



```java
private Person findAllBy(Integer age, String city, String surname) {
```

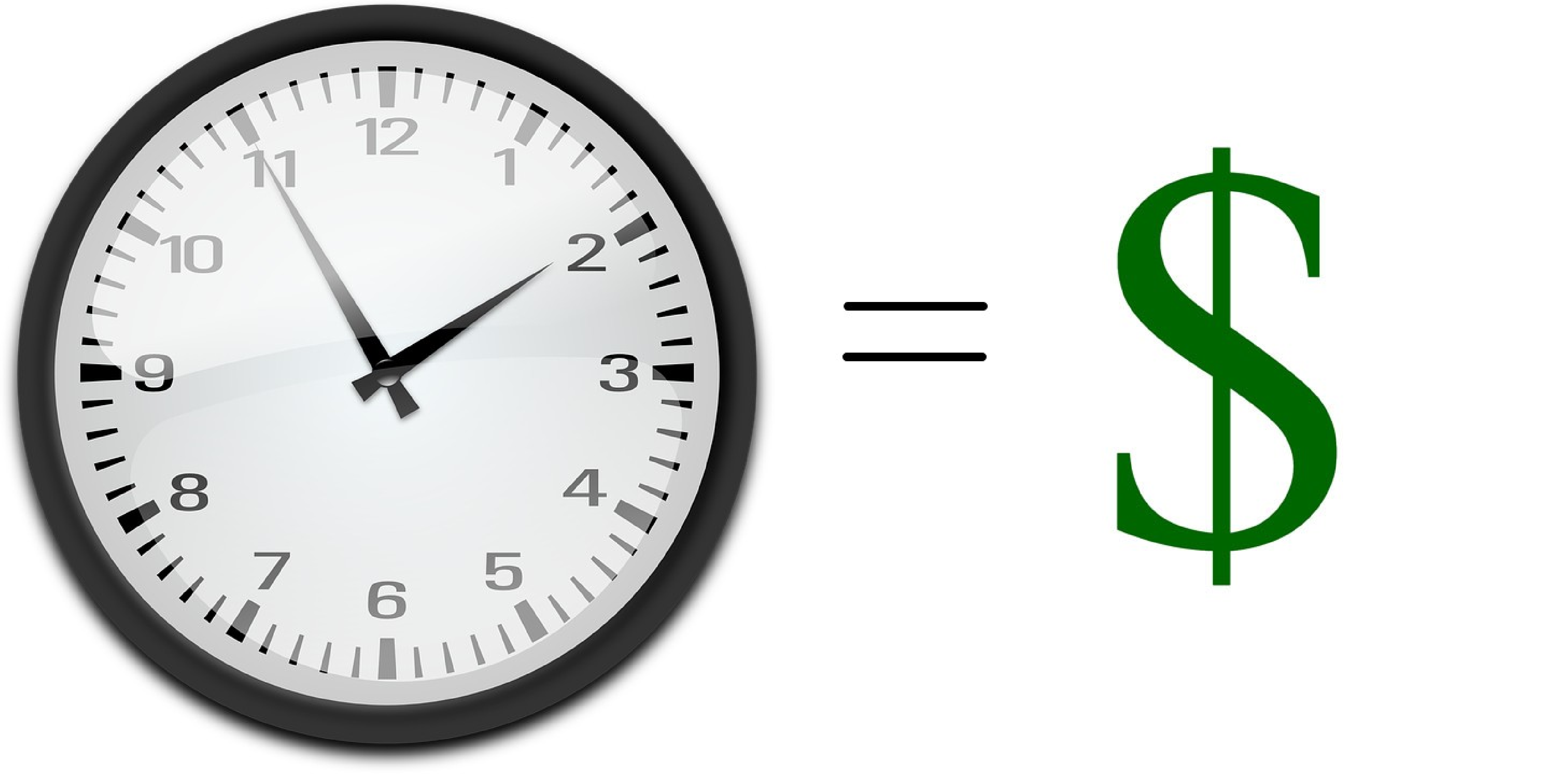
VS

```java
private Person findAllBy(   Integer age, String city,
                            String surname) {
```

VS

```java
private Person findAllBy(
    Integer age,
    String city,
    String surname
) {
```

WHERE ARE TESTS?!


12
1
2
3
4
5
6
7
8
9
10
11
=
$

Let’s automate irrelevant!

Static analysis

Code formatting

Broken rule?

Code style analysis (eg. Checkstyle)

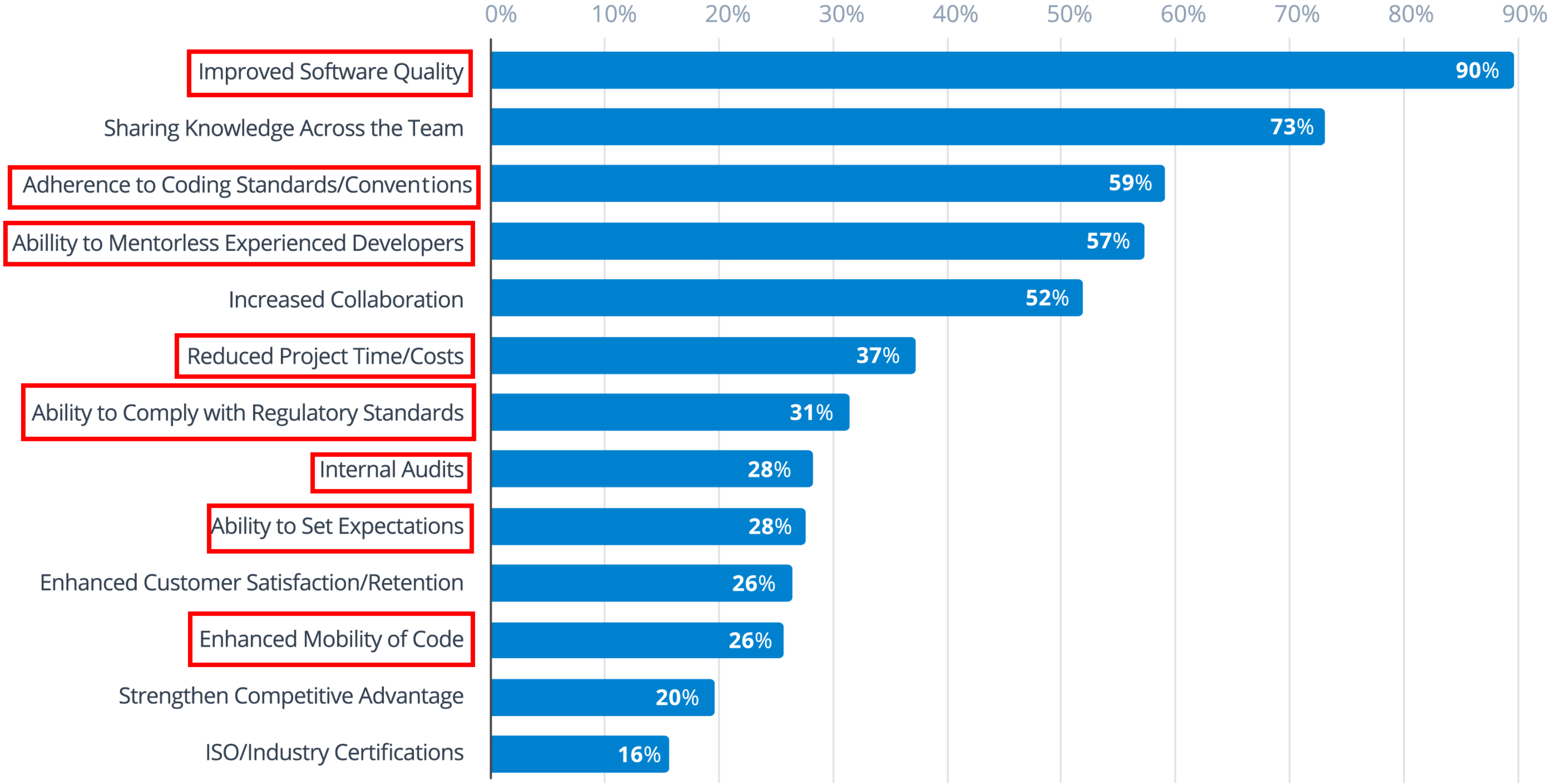
Broken rule?

Potential problems analysis (eg. Findbugs)

Broken rule?

Test coverage analysis (eg. JaCoCO)

Broken rule?


0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
Improved Software Quality
90%
Sharing Knowledge Across the Team
73%
Adherence to Coding Standards/Conventions
59%
Abillity to Mentorless Experienced Developers
57%
Increased Collaboration
52%
Reduced Project Time/Costs
37%
Ability to Comply with Regulatory Standards
31%
Internal Audits
28%
Ability to Set Expectations
28%
Enhanced Customer Satisfaction/Retention
26%
Enhanced Mobility of Code
26%
Strengthen Competitive Advantage
20%
ISO/Industry Certifications
16%

Why so strict with these rules?

Who will set the rules?

What about stupid rules?

What can I gain?

- more time to focus on domain problems
- more time to focus on real code complexity problems
- predictability of design
- ensuring the standards
- **learning**

Q&A