

33 THINGS YOU WANT TO DO BETTER

TOM BUJOK

33 THINGS YOU WANT TO DO BETTER

ABOUT ME

TOM BUJOK

SBB, BERN, CH

@TOMBUJOK

WWW.REFICIO.ORG



no fluff {jobs}

Lean - Transparent - Effective

www.nofluffjobs.com

THE CODE WE PRODUCE IS BAD

SPAGHETTI CODE **JUNGLE**



JOHN - THE DEVELOPER









WELL-DEFINED STANDARDS

[TIME | BUSINESS]+ PRESSURE

WHAT ABOUT OUR SKILLS?



**PRACTICE MAKES
PERFECT**

**DEPLOYING OUR SKILLS
IS MUTUALLY EXCLUSIVE
FROM MASTERING THEM!**

"WE ARE WHAT WE REPEATEDLY DO; EXCELLENCE,
THEN, IS NOT AN ACT BUT A HABIT."

~ARISTOTLE

WHAT IS A **HABIT**?

Wikipedia:

"Habits are routines of behavior that are repeated regularly and tend to occur **subconsciously**."

UNDESIRABLE BEHAVIOR PATTERN

BAD HABITS - ELIMINATE ASAP

Bad Habits - Katherine Murdock “The Psychology of Habit”:

- Recognise bad habits and eliminate them ASAP
- The older you get the more difficult it is to remove a bad habit
- Each repetition leaves its mark!

Turning bad habits into good ones - Dr. Michael Roussell, PhD.:

- You can't erase a habit, you can only overwrite one.
- Insert the new habits into the current habit loops

LET'S RECAP

LOMBOK


```
public class Person {  
    private final Integer age;  
    private final String name;  
    private final String surname;  
}
```

```
@ToString  
@EqualsAndHashCode  
@RequiredArgsConstructor  
@Getter @Setter
```

```
public class Person {  
  
    private final Integer age;  
    private final String name;  
    private final String surname;  
  
}
```

@Data

public class Person {

private final Integer age;

private final String name;

private final String surname;

}

```
@Builder
```

```
public class Person {
```

```
    @NonNull private final Integer age;
```

```
    @NonNull private final String name;
```

```
    @NonNull private final String surname;
```

```
}
```

```
Person.builder()
```

```
    .name("James").surname("Bond").age(33)
```

```
    .build();
```

@Log4j

```
public class CleanupLombokExample {
```

```
    public void read(String input, String output) throws {  
        @Cleanup InputStream in = new FileInputStream(input);  
        @Cleanup OutputStream out = new FileOutputStream(output);  
        byte[] b = new byte[1024];  
        while (true) {  
            int r = in.read(b);  
            if (r == -1) break;  
            out.write(b, 0, r);  
        }  
        log.info("Finished!");  
    }
```

```
}
```

@Log4j

```
public class CleanupLombokExample {
```

```
    public void read(String input, String output) throws {  
        @Cleanup InputStream in = new FileInputStream(input);  
        @Cleanup OutputStream out = new FileOutputStream(output);  
        byte[] b = new byte[1024];  
        while (true) {  
            int r = in.read(b);  
            if (r == -1) break;  
            out.write(b, 0, r);  
        }  
        log.info("Finished!");  
    }  
}
```

GUAVA

@Test

public void exampleOptionalPresent() {

```
    Optional<Integer> possible = Optional.of(5);  
    possible.isPresent(); // returns true  
    possible.get(); // returns 5
```

}

@Test(expected = IllegalStateException.class)

public void exampleOptionalEmpty() {

```
    Optional<Integer> empty = Optional.absent();  
    empty.isPresent(); // returns false  
    empty.get(); // throws IllegalStateException
```

}


```
@Test
public void validationExample() {

    Preconditions.checkNotNull(initialized);

    Preconditions.checkNotNull("IDLE".equals(status), "!IDLE");

    Preconditions.checkNotNull(start, end, data.length);

    String currentUser = Preconditions.checkNotNull(userId);

}
```

```
@Test
public void exampleMapDeclaration() {

    Map<String, String> mapJ6 = new HashMap<String, String>();

    Map<String, String> mapJ7 = new HashMap<>();

    Map<String, String> mapGuava = Maps.newHashMap();

}
```

```
@Test
public void exampleCollections() {

    // Map<String, List<Long>>
    Multimap<String, Long> ml = ArrayListMultimap.create();

    // Map<String, Set<Long>>
    Multimap<String, Long> ms = HashMultimap.create();

    // Map<String, Map<String, Long>>
    Table<String, String, Long> tab = HashBasedTable.create();

}
```

LAMBDAJ

@Test

```
public void filteringComplexShowcase() {
```

```
    Person me = new Person("Mario", "Fusco", 35);
```

```
    Person luca = new Person("Luca", "Marrocco", 29);
```

```
    Person biagio = new Person("Biagio", "Beatrice", 39);
```

```
    List<Person> list = asList(me, luca, biagio);
```

```
        filter(having(
            on(Person.class).getAge(),
            greaterThan(30)), list);
```

```
}
```

@Test

public void closureSimpleShowcase() {

```
Closure println = closure(); {  
    of(System.out).println(var(String.class));  
}
```

```
println.apply("one");  
println.each("one", "two", "three");
```

}

LOGGING

LOG4J

```
try {  
    throw new RuntimeException("Dev oops");  
} catch (Exception ex) {  
    log.error(ex);  
}
```

```
try {  
    throw new RuntimeException("Dev oops");  
} catch (Exception ex) {  
    log.error("Caught you bastard!", ex);  
}
```


SLF4J

```
try {  
    throw new RuntimeException("Dev oops");  
} catch (Exception ex) {  
    // does not compile  
    // log.error(ex);  
}
```

```
String id = "1231-4935-2314";  
try {  
    throw new RuntimeException("Dev oops");  
} catch (Exception ex) {  
    log.error("Caught you bastard! {}", id, ex);  
}
```

BAD DEFAULTS

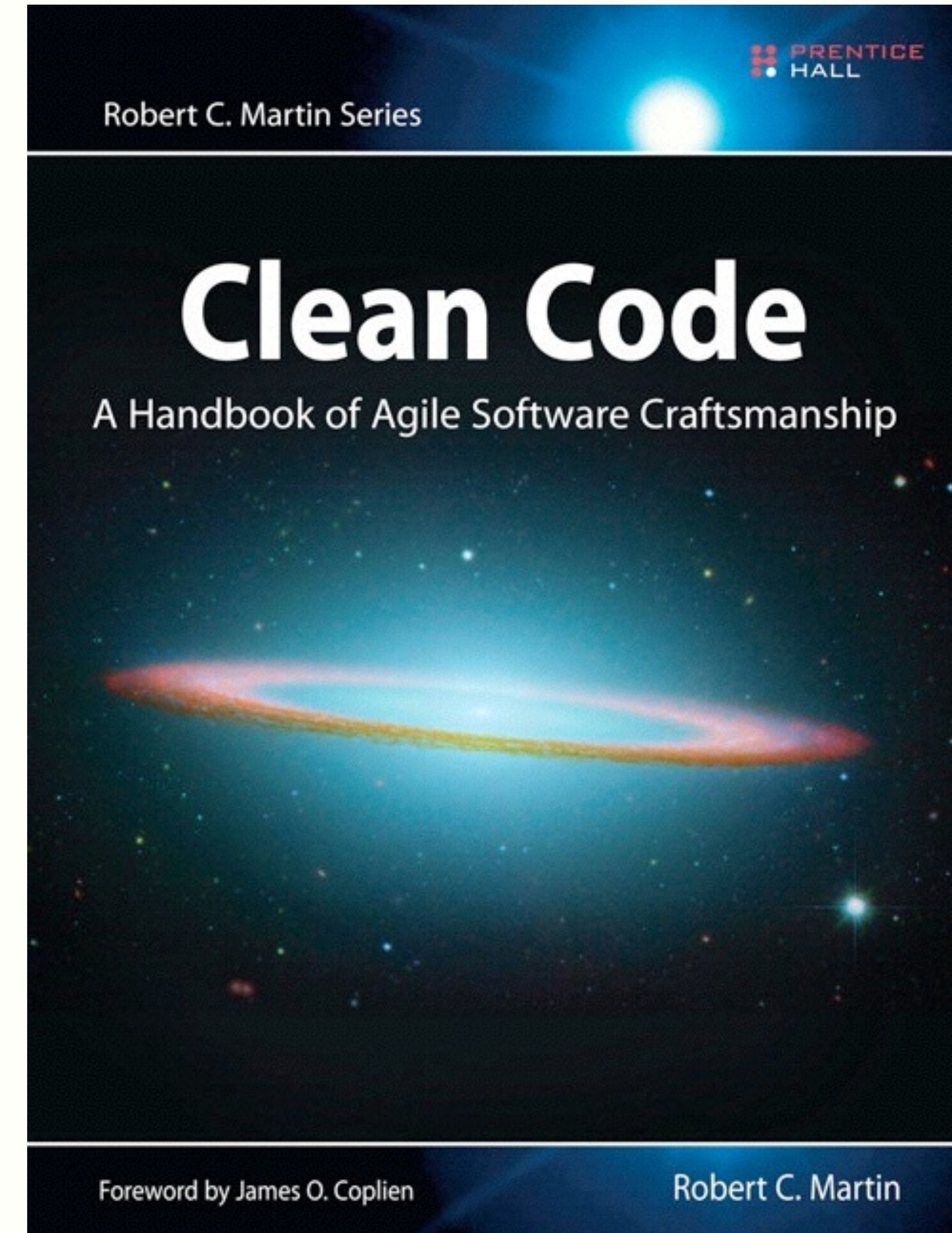
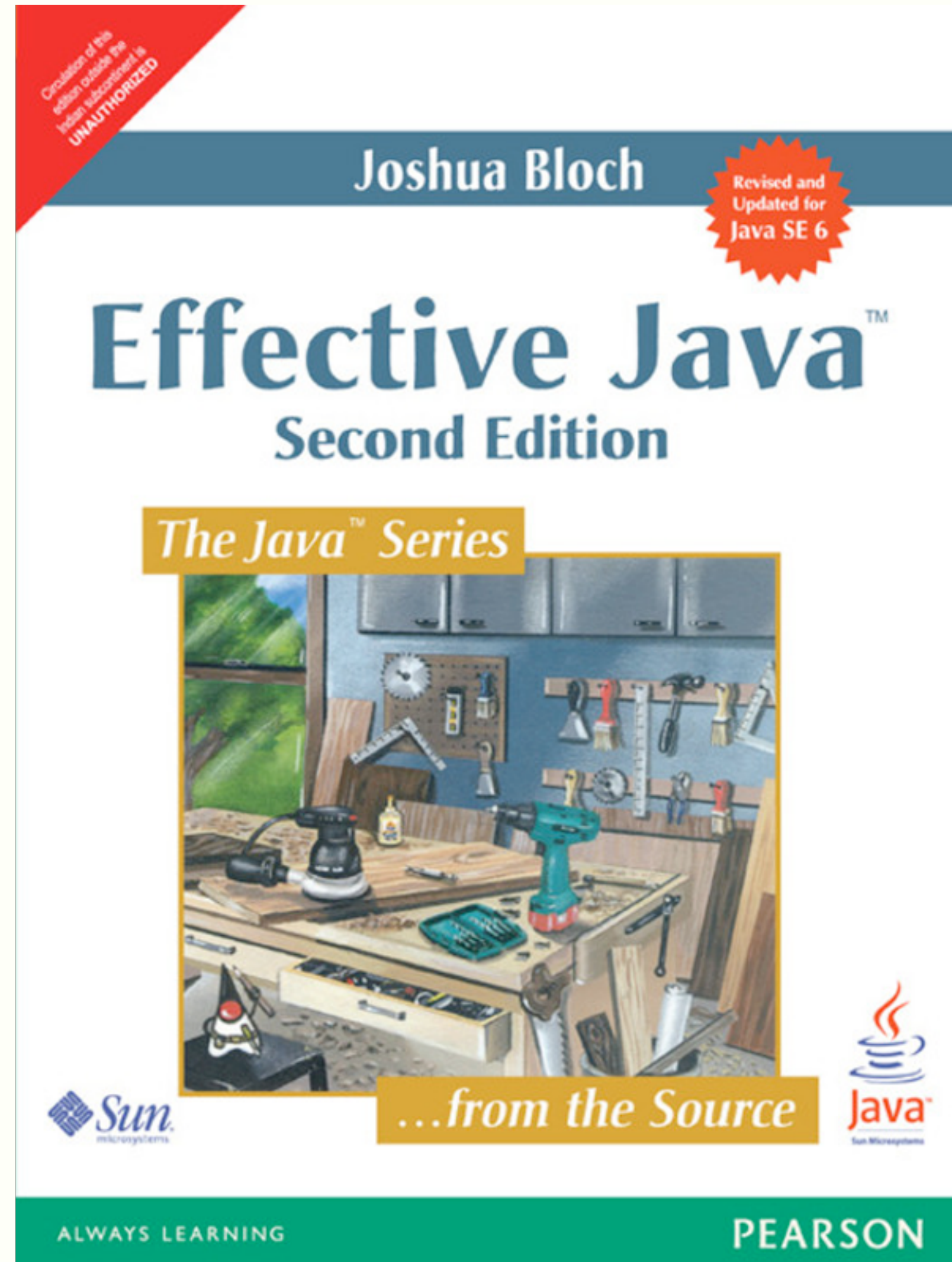
```
@Test
public void autoGeneratedCatchExample_BAD() {
    try {
        stockService.buy("AAPL", 10);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

```
@Test
public void autoGeneratedCatchExample_GOOD() {
    try {
        stockService.buy("AAPL", 10);
    } catch (Exception ex) {
        throw new RuntimeException(ex.getMessage(), ex);
    }
}
```

```
@Test
public void insaneExceptionHandling() {
    try {
        stockService.buy("AAPL", 10);
    } catch (Exception ex) {
        // ignored as it will never happen
    }
}
```

```
@Test
public void insaneExceptionHandlingTrick() {
    try {
        stockService.buy("AAPL", 10);
    } catch (Exception ex) {
        // ignored as it will never happen
        System.exit(-1); // <PEACE!>
    }
}
```

```
public final class QueryUtil {  
    public static String extractTableName(String query) {  
        // (...)   
        return tableName;  
    }  
  
    public static String removeParameters(String query) {  
        // (...)   
        return queryNoParams;  
    }  
  
    public static String formatQuery(String query) {  
        // (...)   
        return queryFormatted;  
    }  
}
```

JOHN - THE DEVELOPER







UNITILS

```
public class UnitilsIOExample extends UnitilsJUnit4 {  
  
    @FileContent(value = "/input.txt", encoding = "UTF-8")  
    private String data;  
  
    @TempFile("output.txt")  
    private File output;  
  
    @Test  
    public void exampleFileContent() {  
        assertEquals("Jazoon 2013 rocks!", data);  
    }  
  
}
```

```
@Test
public void exampleAssertion() {
    User user1 = new User(1, "John", "Doe");
    User user2 = new User(1, "John", "Doe");

    assertFalse(user1.equals(user2));
    assertReflectionEquals(user1, user2);
}
```

```
@Test
public void exampleListAssertion() {
    List<Integer> l = asList(2, 1);

    assertReflectionEquals(asList(1, 2), l, LENIENT_ORDER);
}
```

```
private Service service = new Service();
```

```
@InjectInto(target = "service", property = "dao")  
private Dao dao = new Dao();
```

```
@InjectIntoStatic(target = Dao.class, property = "INSTANCE")  
private Dao singletonOverridingDao = new Dao();
```

```
@Test  
public void testSingletonOverrideInjection() {  
    assertSame(singletonOverridingDao, Dao.INSTANCE);  
}
```

```
}
```

JUNITPARAMS


```
@RunWith(JUnitParamsRunner.class)
public class HardcodedParamsExample {

    @Test
    @Parameters({
        "17, false",
        "22, true"
    })
    public void personIsAdult(int age, boolean correct) {
        assertThat(new Person(age).isAdult(), is(correct));
    }
}
```

```
public class PersonProvider {  
    public static Object[] provideAdults() {  
        return $(  
            $(new Person(25), true),  
            $(new Person(32), true)  
        );  
    }  
  
    public static Object[] provideTeens() {  
        return $(  
            $(new Person(12), false),  
            $(new Person(17), false)  
        );  
    }  
}
```

AWAITILITY


```
@Test
public void testWithAtomicPseudoClosure() {
    await()
        .atMost(10, SECONDS)
        .untilCall(
            to(userRepository).size(), equalTo(3));
}
```

```
private AtomicInteger atomic = new AtomicInteger(0);
```

```
@Test
public void testWithAtomicNumber() {
    await()
        .untilAtomic(atomic, equalTo(1));
}
```

```
}
```

BYTEMAN


```
@RunWith(BMUnitRunner.class)
public class FalseInjectionTest {

    @Test(expected = RuntimeException.class)
    @BMRule(name = "Processor IOException",
            targetClass = "FalseInjectionTest$StockService",
            targetMethod = "getQuote",
            action = "throw new RuntimeException()")
    public void checkIfFaultInjected() {
        new StockService().getQuote("AAPL");
    }
}
```

SPOCK


```

class SpockSpecExample extends Specification {

    @Unroll
    def "should detect blank for [#input]"() {

        expect:
        StringUtils.isBlank(input) == result

        where:
        input | result
        null  | true
        ""    | true
        " "   | true
        "\t\n" | true
        "."    | false

    }

}

```

```
def "should send messages to all subscribers"() {
```

```
  setup:
```

```
    Subscriber john = Mock(), jim = Mock()  
    String msg = "33rd Degree rocks!"
```

```
  when:
```

```
    publisher.send(msg)
```

```
  then:
```

```
    1 * john.receive(msg)  
    1 * jim.receive(msg)
```

```
}
```

```
}
```


GROOVY

XML PARSING

```
<?xml version="1.0" encoding="UTF-8"?>
<stocks>
  <stock>
    <symbol>Citibank</symbol>
    <price>100</price>
    <quantity>1000</quantity>
  </stock>
  <stock>
    <symbol>UBS</symbol>
    <price>90</price>
    <quantity>2000</quantity>
  </stock>
</stocks>
```


JAVA

```
DocumentBuilderFactory dbFactory =
    DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(stocks);
doc.getDocumentElement().normalize();
NodeList nodes = doc.getElementsByTagName("stock");
for (int i = 0; i < nodes.getLength(); i++) {
    Node node = nodes.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element el = (Element) node;
        NodeList list = el.getElementsByTagName("symbol");
        NodeList children = list.item(0).getChildNodes();
        System.out.println(children(0).getNodeValue());
    }
}
```

GROOVY

```
def doc = new XmlSlurper().parse(stocks)
doc.stock.each { println it.symbol.text() }
```


GRAPE

```
@Grapes ( [  
    @Grab(group='commons-lang',  
          module='commons-lang',  
          version='2.4')  
])  
  
def abbreviate(String s) {  
    StringUtils.abbreviate( s, 10 )  
}  
  
def strings = ['Hello', 'Groovy']  
strings.each { String aString ->  
    println "$aString: ${abbreviate(aString)}"  
}
```

GRADLE

MAVEN

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.demo</groupId>
  <artifactId>demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.mockito</groupId>
      <artifactId>mockito-all</artifactId>
      <version>1.8.4</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-resources-plugin</artifactId>
        <version>2.6</version>
        <executions>
          <execution>
            <id>copy-resources</id>
            <phase>validate</phase>
            <goals>
              <goal>copy-resources</goal>
            </goals>
            <configuration>
              <outputDirectory>${basedir}/target/resources/copy</outputDirectory>
              <resources>
                <resource>
                  <directory>src/test/resources</directory>
                  <filtering>true</filtering>
                </resource>
              </resources>
            </configuration>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</project>
```

GRADLE

```
apply plugin: 'java'

group='org.demo'
artifactId='demo'
version='0.1-SNAPSHOT'

repositories {
    mavenCentral()
}

dependencies {
    testCompile 'junit:junit:4.8.1'
    testCompile 'org.mockito:mockito-all:1.8.4'
}

task copyTestResources(type: Copy, dependsOn: 'assemble') {
    from 'src/test/resources'
    into 'src/test/resources/copy'
    include('**/*.xml', '**/*.properties')
}
```

TOOLS

GIT

```
git log  
git status  
git diff
```

```
git add <file>  
git checkout -- <file>  
git reset HEAD <file>
```

```
git branch  
git branch <branch_name>  
git branch -d <branch_name>  
git checkout <branch_name>
```

```
git commit -m "message"  
git commit --amend  
git merge <branch>
```

```
git stash list  
git stash push  
git stash pop
```

```
git push origin master  
git pull  
git reset HEAD  
git reset --hard
```


BASH

```
find -regex .*\\.groovy -exec grep -Hn "import spock." {} \;
```

```
find -regex .*\\.groovy -exec sed -i "s/@mrbean/@tom/g" {} \;
```

```
ls -Al | awk '{ print $6 " " $10 }' | sort -r
```

BABUN

```
ue62407@K10323:~  
ue62407::K10323 { ~ }-> ls -al  
total 31  
drwxr-xr-x+ 1 ue62407 Domain Users    0 Sep 11 16:16 .  
drwxrwxrwt+ 1 ue62407 Domain Users    0 Sep 11 16:16 ..  
-rwxr-xr-x  1 ue62407 Domain Users   356 Sep 11 16:16 .babunrc  
-rw-----  1 ue62407 Domain Users    21 Sep 17 10:07 .bash_history  
-rwxr-xr-x  1 ue62407 Domain Users  1496 Sep 11 07:04 .bash_profile  
-rwxr-xr-x  1 ue62407 Domain Users  6778 Sep 11 07:04 .bashrc  
-rwxr-xr-x  1 ue62407 Domain Users  1916 Sep 11 07:04 .inputrc  
-rwxr-xr-x  1 ue62407 Domain Users    674 Sep 11 07:04 .minttyrc  
-rwxr-xr-x  1 ue62407 Domain Users  1012 Sep 11 07:04 .nanorc  
drwxr-xr-x+ 1 ue62407 Domain Users    0 Sep 11 16:16 .vim  
-rwxr-xr-x  1 ue62407 Domain Users    53 Sep 11 07:04 .vimrc  
ue62407::K10323 { ~ }-> bark install vim  
Working directory is /setup  
Mirror is http://mirror.mcs.anl.gov/pub/cygwin/  
--2013-10-21 15:26:29-- http://mirror.mcs.anl.gov/pub/cygwin//x86/setup.bz2  
Resolving webproxy.sbb.ch (webproxy.sbb.ch)... 172.28.15.151, 172.28.15.152  
Connecting to webproxy.sbb.ch (webproxy.sbb.ch)|172.28.15.151|:80... connected.  
Proxy request sent, awaiting response...  
ue62407::K10323 { ~ }-> ^C  
ue62407::K10323 { ~ }->
```

Released on 13th of May 2014

<http://babun.github.io>

IDE

⌘ + ⌥ + B

CTRL + ALT + B

CODE REVIEWS

WRAP-UP



Any jackass can kick down a barn, but it takes a
good carpenter to build one.

(Sam Rayburn)

izquotes.com



ANY QUESTIONS?

THANK YOU

TOM BUJOK

@TOMBUJOK

WWW.REFICIO.ORG

