Dominare il codice ereditato

Tommaso Torti e Matteo Vaccari

Agile Day 2007, Bologna, 23 novembre

sourcesense

ReadMe

Per vedere funzionare l'applicazione:

* modifica /etc/hosts inserendo

  127.0.0.1 xxx.yyy.it
  10.0.1.2 xxx.zzz.it

* inserisci i seguenti plugin di Firefox:
  * Modifica gli header: https://addons.mozilla.org/en-US/firefox/addon/967
    vai su Tools -> Modify Headers e aggiungi:
    MSISDN = 393928390078
    PARTY-ID = 34353252
  * User agent switcher: https://addons.mozilla.org/en-US/firefox/addon/59
    salvare il file http://xxx.sourcesense.com/files/zero9/useragentswitcher.xml
    e importarlo

* esegui "script/create_databases.sh"
* esegui "script/start.sh"
* punta il browser a http://localhost:8080/progetto/p.do?page=Home

sourcesense

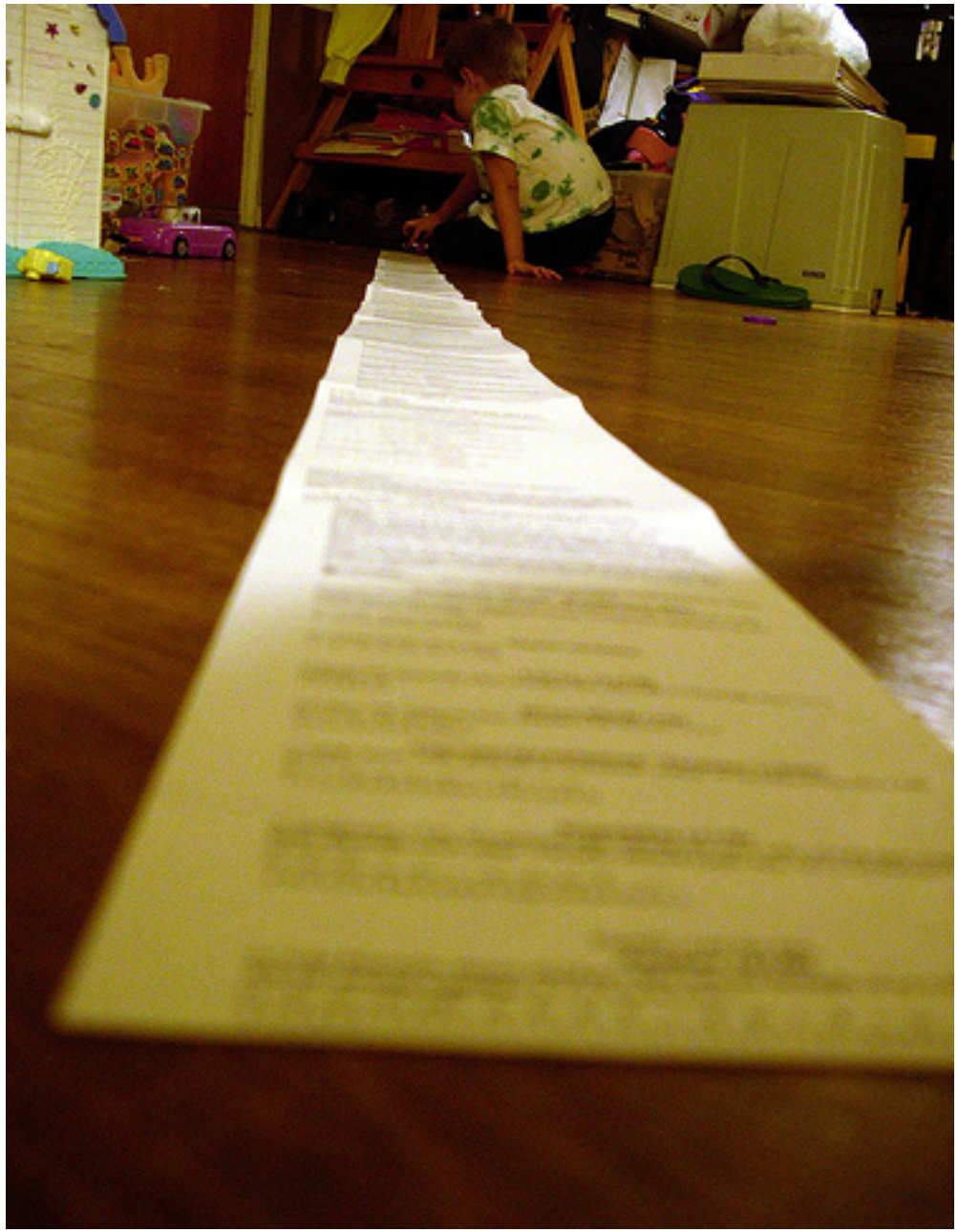
create_databases.sh

```
#!/bin/bash
if [ ! -d db ]; then
  echo "Questo script deve essere eseguito nella dir principale del progetto"
  exit 1
fi
echo 'Drop databases...'
mysqladmin -uroot --force drop db
mysqladmin -uroot --force drop db_test

echo 'Create databases...'
mysqladmin -uroot create db
mysqladmin -uroot create db_test
echo "grant all on db.* to db@localhost identified by 'db';" |  mysql -uroot
echo "grant all on db_test.* to db@localhost identified by 'db';" |  mysql -uroot

echo 'Build schema...'
cat db/db-schema.sql | mysql -udb db -pdb
cat db/db-schema.sql | mysql -udb db_test -pdb

echo 'Populate development...'
mysql -udb -pdb db < db/populate_db.sql

echo 'Done!'
```

start.sh

```
#!/bin/bash
if [ -z "${CMT_DEVELOPMENT_UPLOAD}" ] ; then
 echo "Deve essere settata la variabile di ambiente CMT_DEVELOPMENT_UPLOAD";
 exit 1;
fi

ABS_PATH=$(cd $(dirname $0); cd ..; pwd)
CATALINA_HOME="$ABS_PATH/tomcat-5.5.25"

rm -rf $CATALINA_HOME/logs/*
rm -rf $CATALINA_HOME/webapps/progetto*

ant clean
ant deploy

ln -s /tmp $CATALINA_HOME/webapps/progetto/dynamicImages/upload

$CATALINA_HOME/bin/catalina.sh jpda start

tail -f $CATALINA_HOME/logs/catalina.out
```

Log



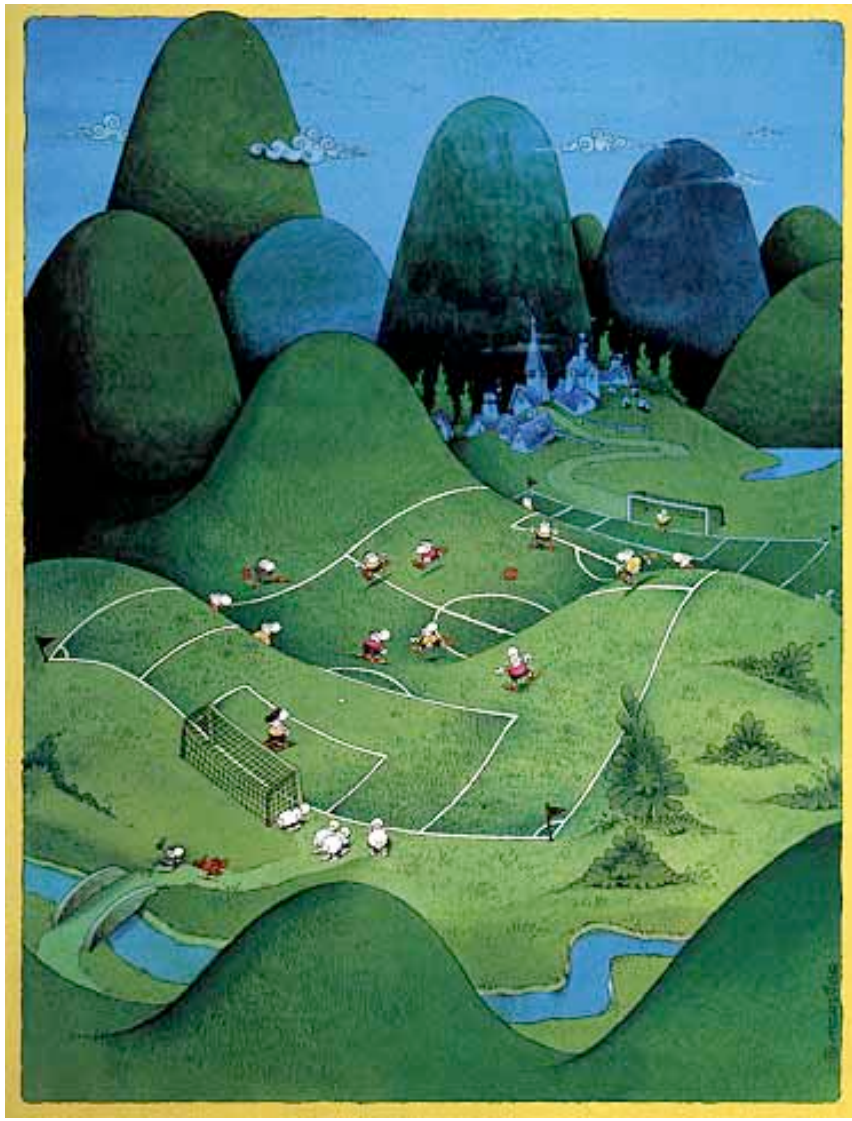
sourcesense

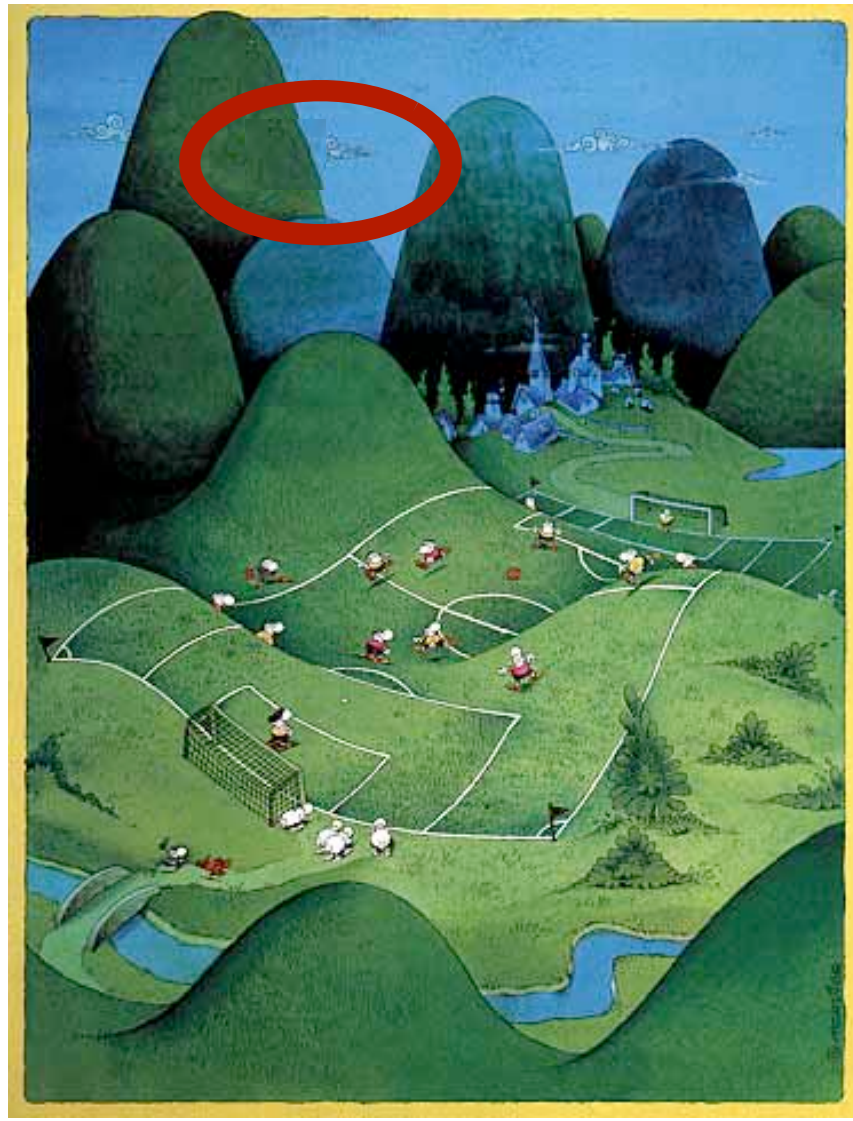
Log

```
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=
   org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=
   %5p %c{1}:%L - %m%n
log4j.rootLogger=INFO, stdout
```

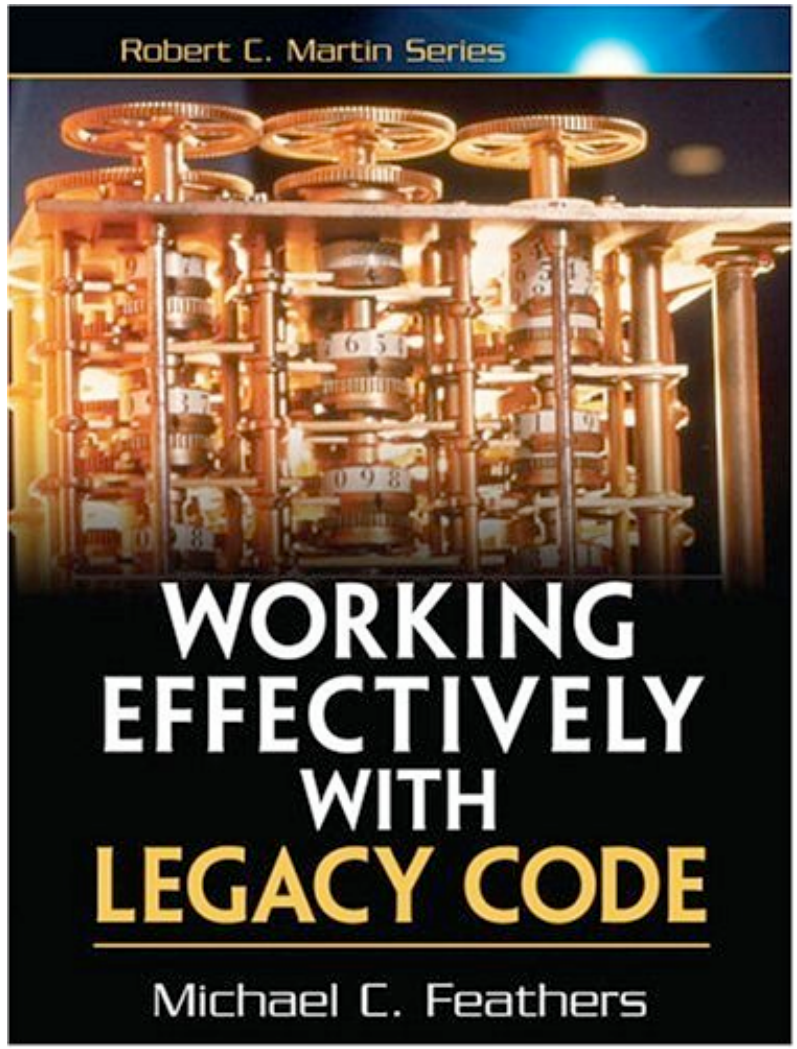
Refactoring servlet

Spot the differences





sourcesense


Robert C. Martin Series
WORKING EFFECTIVELY WITH LEGACY CODE
Michael C. Feathers

Il dilemma

- Non posso rifattorizzare senza test
- Ma il codice è così ingarbugliato, che
- Non posso testare senza rifattorizzare

Il dilemma

- Non posso cambiare il codice senza testare
- Non posso testare senza cambiare il codice

L’algoritmo

- Trova il punto da modificare
- Rompi le dipendenze
- Scrivi i test
- Modifica; rifattorizza

Rompere le dipendenze

- Controfigure
- Cuciture
- Oggetti umili

Controfigure

- Come testare una servlet?
- Senza usare un web server?

```
public class ImageServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response) {

    // ...

    }

}
```

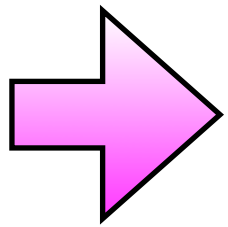
- HttpServletRequest: 54 metodi!

```java
// Generata automaticamente da Eclipse
public class EmptyServletRequest implements HttpServletRequest {
    public String getAuthType() {
        return null;
    }
    // ... altri 53 metodi vuoti
}

// Amorevolmente scritta a mano
public class FakeRequest extends EmptyServletRequest {
    public Map properties = new HashMap();

    public Object getAttribute(String arg0) {
        return properties.get(arg0);
    }
}
```

```
@Test public void testImagingServlet() {
    FakeRequest request = new FakeRequest();
    request.properties.set("foo", "bar");

    ImagingServlet servlet = new ImagingServlet();
    servlet.init(defaultServletConfig());
    servlet.service(request, response);

    assertMatch("/Hello world/", response.getBody());
}
```

Extract method

```
public void veryLongMethod() {
    boring code
    boring code
    boring code

    interesting code
    interesting code
    interesting code

    boring code
    boring code
    boring code
}
```

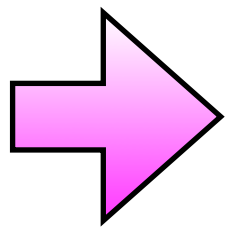


```
public void interestingCode() {
    interesting code
    interesting code
    interesting code
}

public void veryLongMethod() {

    boring code
    boring code
    boring code

    interestingCode();

    boring code
    boring code
    boring code
}
```

```
@Test public void myTest() {
    ClassUnderTest object = new ClassUnderTest();
    object.interestingCode();
    assert ...
}
```

Sfrutta le cuciture

```
class HardToTest {

    public void veryLongMethod() {
        askDifficultQuestions();
        interestingCode();
        contactFarAwayServer();
    }

}
```



```
class Tamed extends HardToTest {
    public Answer askDifficultQuestions() {
        return easyToTestValue;
    }
    public void contactFarAwayServer() {
        // be quiet
    }
}

@Test public void myTest() {
    Tamed object = new Tamed();
    object.interestingCode();
    assert ...
}
```

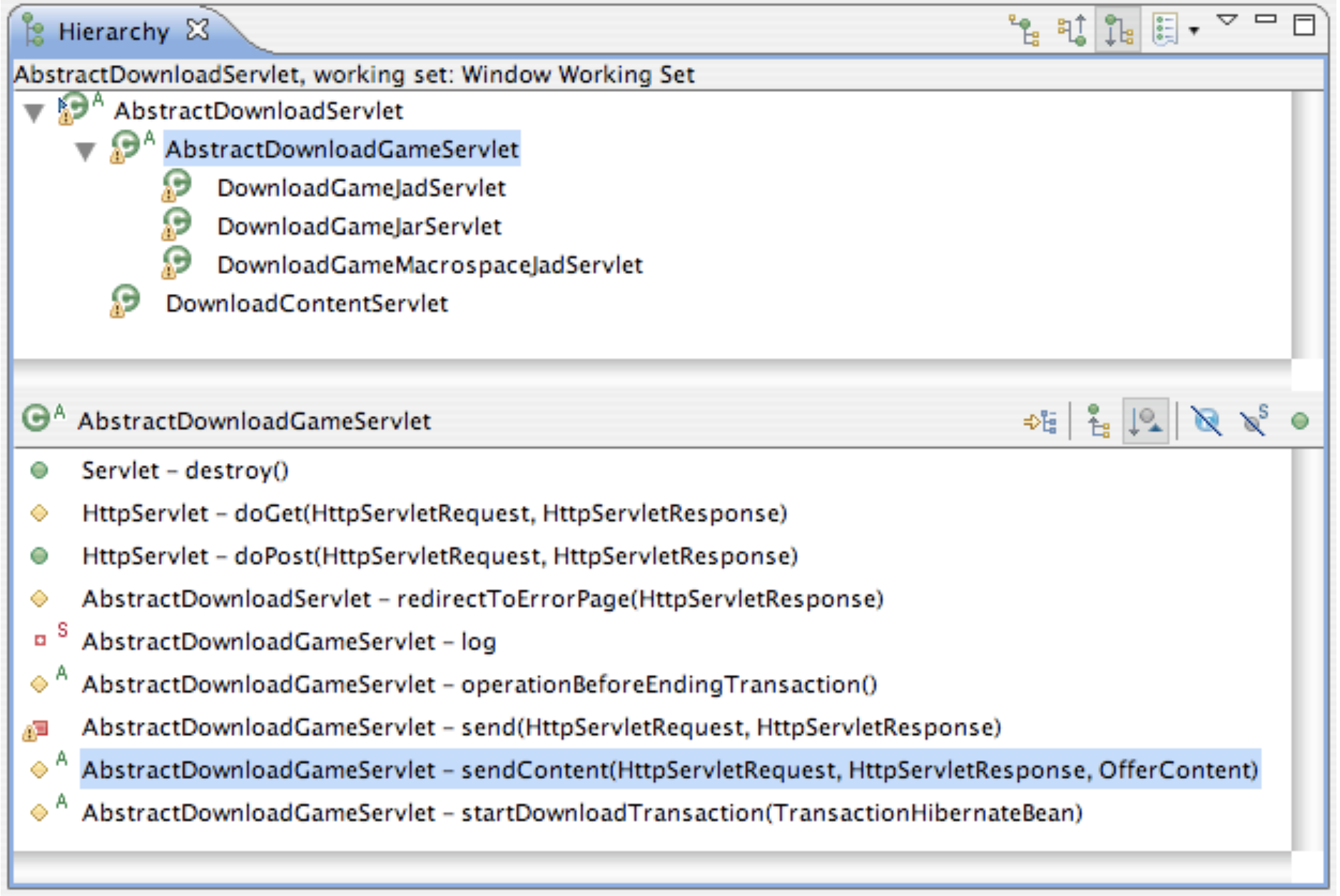
Test di integrazione

```
// i test devono essere LEGGIBILI!

public class AdministrationPanelTest extends HttpBaseTestCase {

    @Test
    public void shouldHaveAdministrationPanel() throws Exception {
        get("/app?category=" + categoryWallpapers.getId());
        HTMLElement adminPanelElement = response.getElementWithID("administrationPanel");
        assertNotNull("non c'e' il pannello di amministrazione", adminPanelElement);
    }
}
```

```
public abstract class HttpBaseTestCase extends BaseTestCase {

    protected ServletUnitClient client;
    protected WebResponse response;
    protected ServletRunner sr;

    @BeforeClass
    public static void useTestDatabase() {
        CmtConfig.reset("test");
    }

    @Before
    public void setUpServletRunner() throws Exception {
        sr = new ServletRunner(new FileInputStream("conf/development/web.xml"), "");
        sr.registerServlet("*.js", NullServlet.class.getName());
        client = sr.newClient();
    }

    protected void get(String url) throws Exception {
        response = client.getResponse("http://localhost" + url);
    }
}
```

http://httpunit.sourceforge.net/

Refactoring servlet

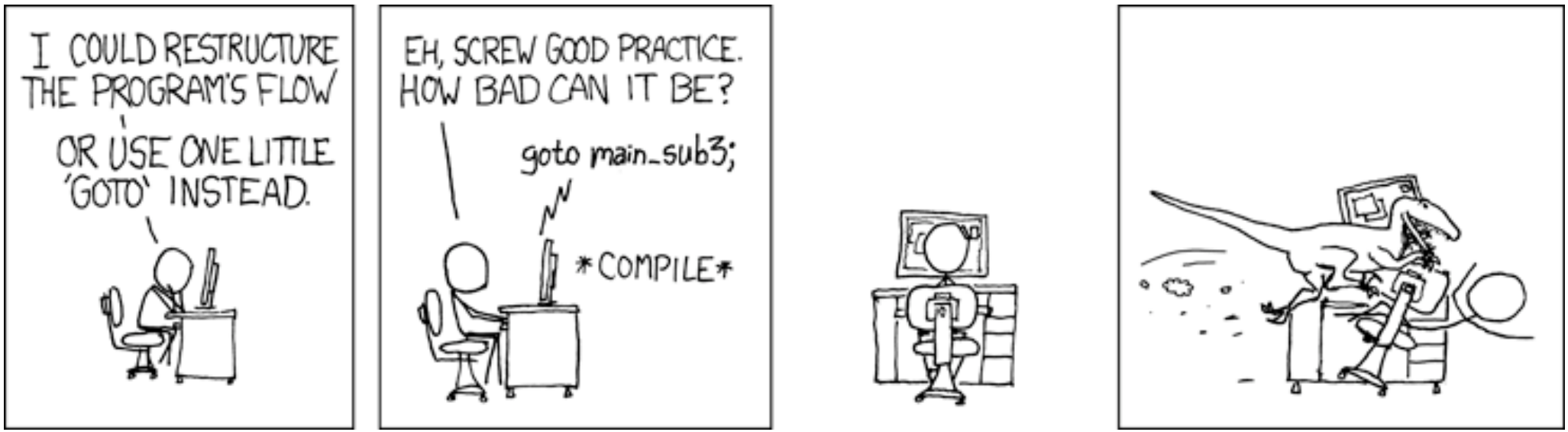


sourcesense

Sensazioni

sourcesense

Tentazioni



xkcd.com

sourcesense

Test jsp

```
@Test
public void testHomePage() throws Exception {
    FakePageContext pc = new FakePageContext();
    setRequestAttribute("DEVICECAPABILITIES", pc.capabilities);

    get("/home.jsp");

    output().shouldContain("NO-CACHE");
}
```

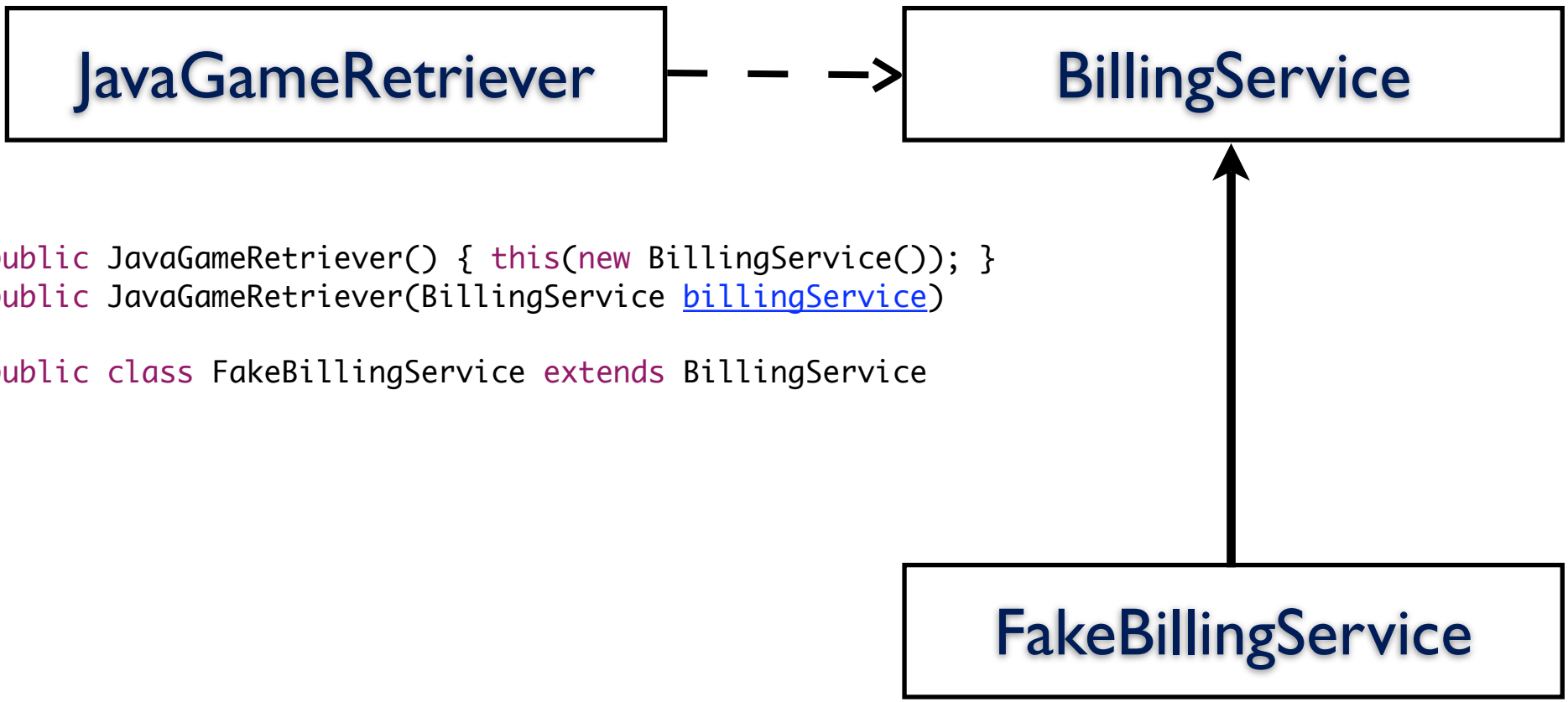
....

```
protected void request(String path, String httpMethod) throws Exception {
    JspCompiler compiler = JspCompilerFactory.newInstance();
    compiler.setWebRoot(getWebRoot());
    compiler.setOutputDirectory(getOutputDirectory());
    Jsp jsp = compiler.compile(path, substituteTaglibs);
    execution = jsp.request(httpMethod, requestAttributes, sessionAttributes);
}
```

http://sourceforge.net/projects/jsptest

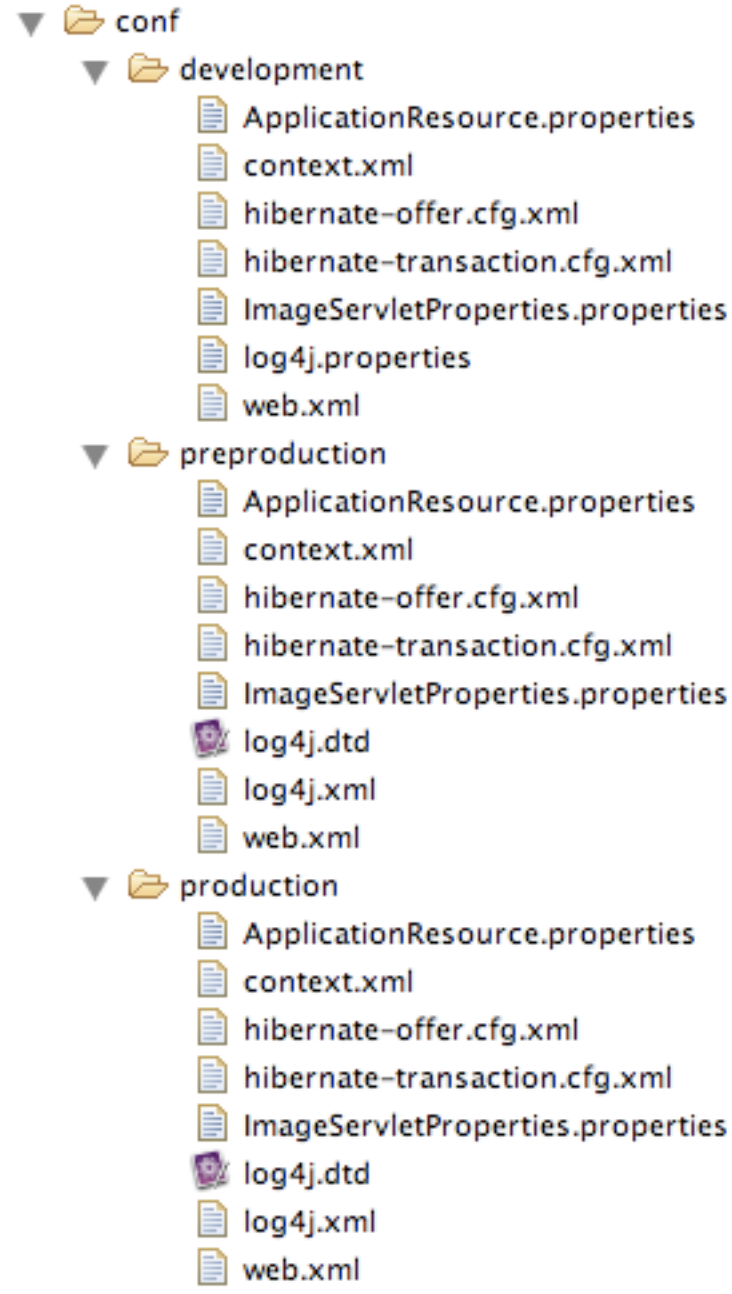
sourcesense

Dipendenze




```
public JavaGameRetriever() { this(new BillingService()); }
public JavaGameRetriever(BillingService billingService)

public class FakeBillingService extends BillingService
```

Configurazioni




```
<target name="prepare">
  <copy todir="./web/WEB-INF/">
    <fileset dir="${conf.dir}">
      <include name="ApplicationResource.properties"/>
      <include name="web.xml" />
    </fileset>
...

<target name="clean">
   <delete file="./web/WEB-INF/
                  ApplicationResource.properties"/>
   <delete file="./web/WEB-INF/web.xml" />
```

---

createWarForPreProduction.sh

```
#!/bin/bash
ant clean
ant war -Dconf.dir=conf/preproduction
```

sourcesense

Risultati

Dopo

Metric	Total	Mean
Number of Static Methods (avg/max per type)	41	0.151
Total Lines of Code	20976	
Afferent Coupling (avg/max per packageFragment)		8.512
Normalized Distance (avg/max per packageFragment)		0.356
Number of Classes (avg/max per packageFragment)	272	6.326
Specialization Index (avg/max per type)		0.246
Instability (avg/max per packageFragment)		0.587
Number of Attributes (avg/max per type)	590	2.169
Number of Packages	43	
Method Lines of Code (avg/max per method)	12126	5.844
Weighted methods per Class (avg/max per type)	3434	12.625
Number of Overridden Methods (avg/max per type)	155	0.57
Number of Static Attributes (avg/max per type)	671	2.467
Nested Block Depth (avg/max per method)		1.336
Number of Methods (avg/max per type)	2034	7.478
Lack of Cohesion of Methods (avg/max per type)		0.228
McCabe Cyclomatic Complexity (avg/max per method)		1.655
Number of Parameters (avg/max per method)		0.76
Abstractness (avg/max per packageFragment)		0.086
Number of Interfaces (avg/max per packageFragment)	14	0.326
Efferent Coupling (avg/max per packageFragment)		5.093
Number of Children (avg/max per type)	114	0.419
Depth of Inheritance Tree (avg/max per type)		2.011

Prima

Metric	Total	Mean
Number of Static Methods (avg/max per type)	48	0.21
Total Lines of Code	20007	
Afferent Coupling (avg/max per packageFragment)		11.29
Normalized Distance (avg/max per packageFragment)		0.432
Number of Classes (avg/max per packageFragment)	229	7.387
Specialization Index (avg/max per type)		0.254
Instability (avg/max per packageFragment)		0.483
Number of Attributes (avg/max per type)	561	2.45
Number of Packages	31	
Method Lines of Code (avg/max per method)	12244	7.09
Weighted methods per Class (avg/max per type)	3186	13.913
Number of Overridden Methods (avg/max per type)	142	0.62
Number of Static Attributes (avg/max per type)	676	2.952
Nested Block Depth (avg/max per method)		1.444
Number of Methods (avg/max per type)	1679	7.332
Lack of Cohesion of Methods (avg/max per type)		0.256
McCabe Cyclomatic Complexity (avg/max per method)		1.845
Number of Parameters (avg/max per method)		0.83
Abstractness (avg/max per packageFragment)		0.113
Number of Interfaces (avg/max per packageFragment)	14	0.452
Efferent Coupling (avg/max per packageFragment)		6.161
Number of Children (avg/max per type)	94	0.41
Depth of Inheritance Tree (avg/max per type)		2.048

...di tutto il progetto

sourcesense

Risultati

Dopo

Metric	Total	Mean
Number of Static Methods (avg/max per type)	0	0
Total Lines of Code	957	
Afferent Coupling	1	
Normalized Distance	0.109	
Number of Classes	10	
Specialization Index (avg/max per type)		1.19
Instability	0.909	
Number of Attributes (avg/max per type)	11	1.1
Method Lines of Code (avg/max per method)	624	12.48
Weighted methods per Class (avg/max per type)	123	12.3
Number of Overridden Methods (avg/max per type)	13	1.3
Number of Static Attributes (avg/max per type)	43	4.3
Nested Block Depth (avg/max per method)		1.86
Number of Methods (avg/max per type)	50	5
Lack of Cohesion of Methods (avg/max per type)		0.175
McCabe Cyclomatic Complexity (avg/max per method)		2.46
Number of Parameters (avg/max per method)		1.64
Abstractness	0.2	
Number of Interfaces	0	
Efferent Coupling	10	
Number of Children (avg/max per type)	7	0.7
Depth of Inheritance Tree (avg/max per type)		3.9

Prima

Metric	Total	Mean
Number of Static Methods (avg/max per type)	0	0
Total Lines of Code	1280	
Afferent Coupling	2	
Normalized Distance	0.024	
Number of Classes	12	
Specialization Index (avg/max per type)		2.242
Instability	0.857	
Number of Attributes (avg/max per type)	11	0.917
Method Lines of Code (avg/max per method)	885	14.048
Weighted methods per Class (avg/max per type)	161	13.417
Number of Overridden Methods (avg/max per type)	25	2.083
Number of Static Attributes (avg/max per type)	46	3.833
Nested Block Depth (avg/max per method)		1.968
Number of Methods (avg/max per type)	63	5.25
Lack of Cohesion of Methods (avg/max per type)		0.167
McCabe Cyclomatic Complexity (avg/max per method)		2.556
Number of Parameters (avg/max per method)		1.746
Abstractness	0.167	
Number of Interfaces	0	
Efferent Coupling	12	
Number of Children (avg/max per type)	9	0.75
Depth of Inheritance Tree (avg/max per type)		4.083

...delle servlet

sourcesense

Risultati

> Coverage: 10 % su 26056 loc
> Tempi: consegna on time - 7 settimane in un team di 3

Non perdere la testa

- Pianifica per feature
- Automatizza tutto
- Scrivi test
- Rifattorizza
- Lavora (interattivamente) il meno possibile!