Atividades de Teste e Cobertura de Código

Erik Aceiro Antonio

- Bacharelado em Ciência da Computação
  - Universidade Mackenzie

- Mestrado em Engenharia Elétrica
  - Sistemas de Comunicações Ópticas
  - Automação e WebLab
- Doutorando em Ciência da Computação
  - Engenharia de Software – Universidade Federal de São Carlos (UFSCa
  - Teste de Software para Sistemas Embarcados Críticos
- Certificado OCA/OCP/SCJP/LPI
- Trabalhei no UOL/Mackenzie como Analista Programador
- Professor Universitário
- Experiência no ensino de 10 anos
- Consultor de TI

aceiro@gmail.com

erik_aceiro@hotmail.com

http://erikblogger.blogspot.com

facebook.com/erik.aceiro

@eaceiro

Apresentar **princípios** e a **motivação** para o uso de atividades de teste

Erik | aceiro@gmail.com

O que é atividade de Teste ?

"Testing is the process of executing a program with the intent of finding errors"

Myers

As Atividades de Teste promovem...

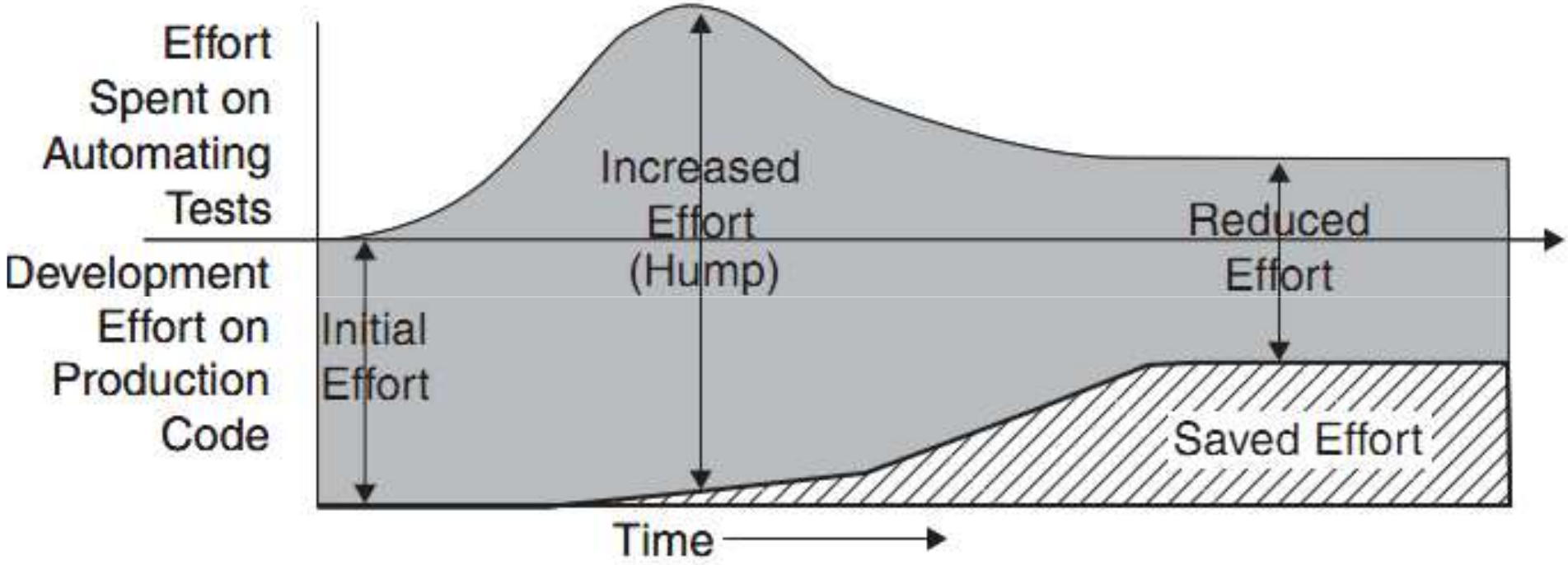
- Auxilia na compreensão do SUT (App)
- Os testes devem reduzir “riscos”
- Reduzir correçoes manuais e “debug”
- Em uma palavra ...

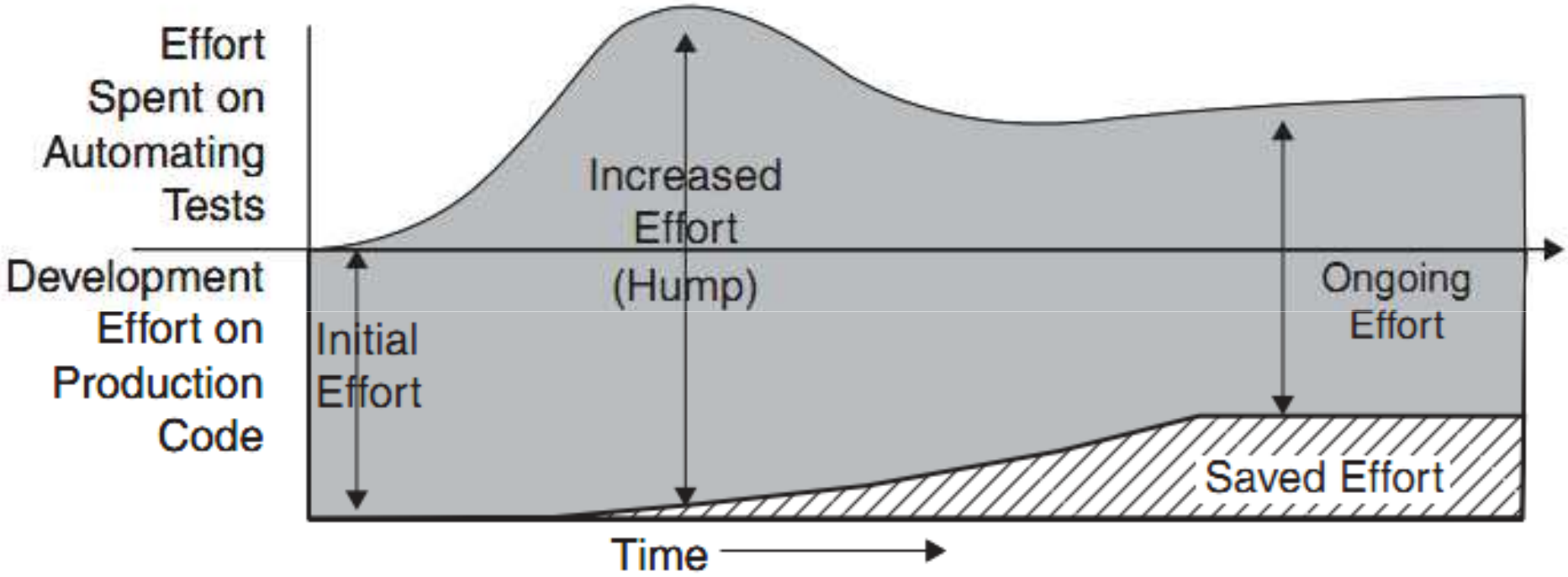
Qualidade

O que é Qualidade?

A busca
pela
qualidade
exige
um
trade-off

Risco
X
Oportunidade


Effort
Spent on
Automating
Tests
Development
Effort on
Production
Code
Initial
Effort
Increased
Effort
(Hump)
Reduced
Effort
Saved Effort
Time


Effort
Spent on
Automating
Tests
Development
Effort on
Production
Code
Increased
Effort
(Hump)
Initial
Effort
Ongoing
Effort
Saved Effort
Time

O que é Qualidade?

"Qualidade" ...

Estamos construindo **certo** o produto ?

Estamos construindo o produto **certo** ?

"Verificação" assegura que

- Estamos construindo **certo** o produto ?

- Assegurar que o software está de acordo com a especificação pré-estabelecida

- Verificar problemas e defeitos em componentes

- Exige a execução de artefatos

"Validação" assegura que

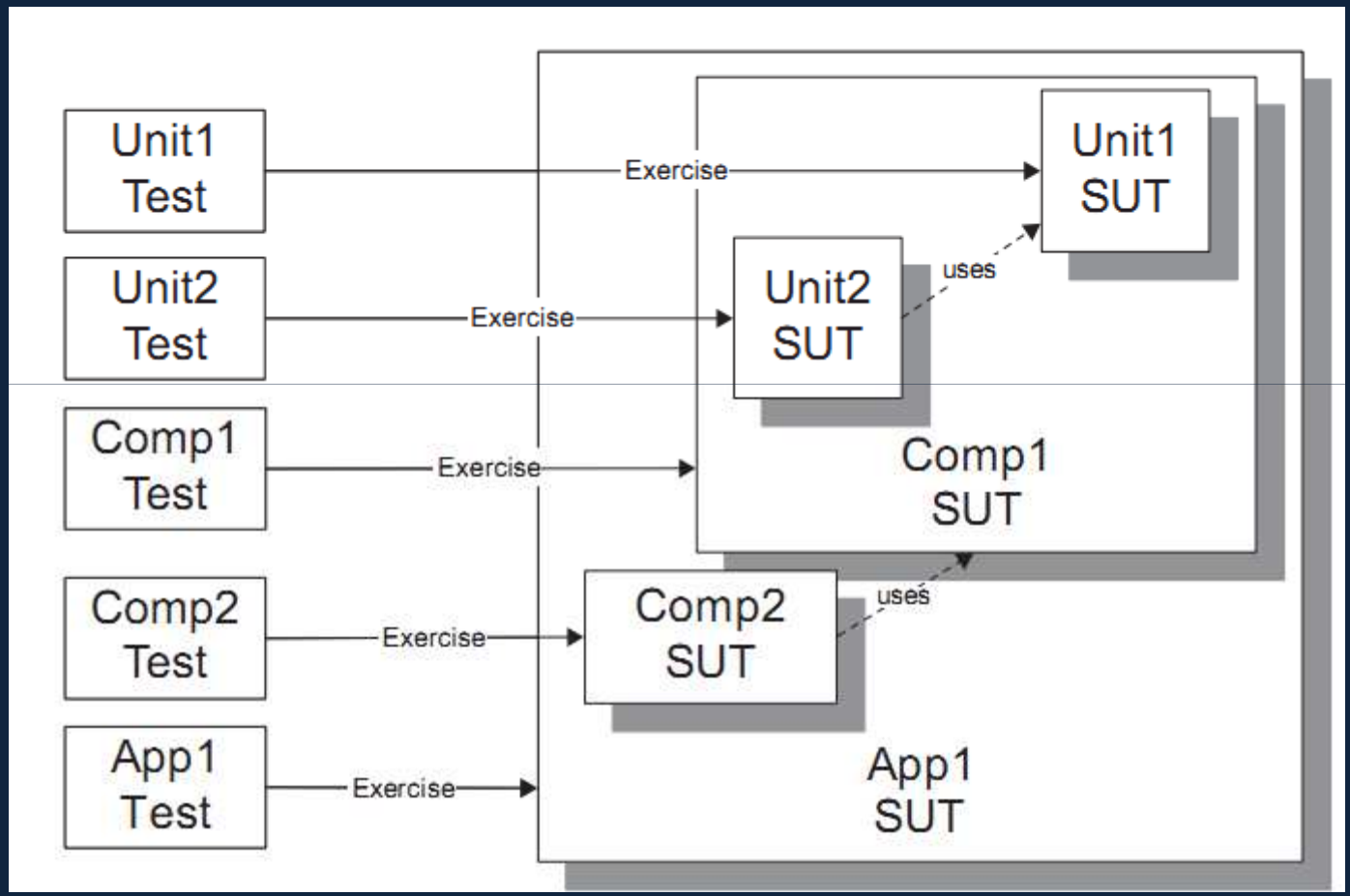
- Estamos construindo o produto **certo** ?

- Assegurar que o software está de acordo com os requisitos do cliente

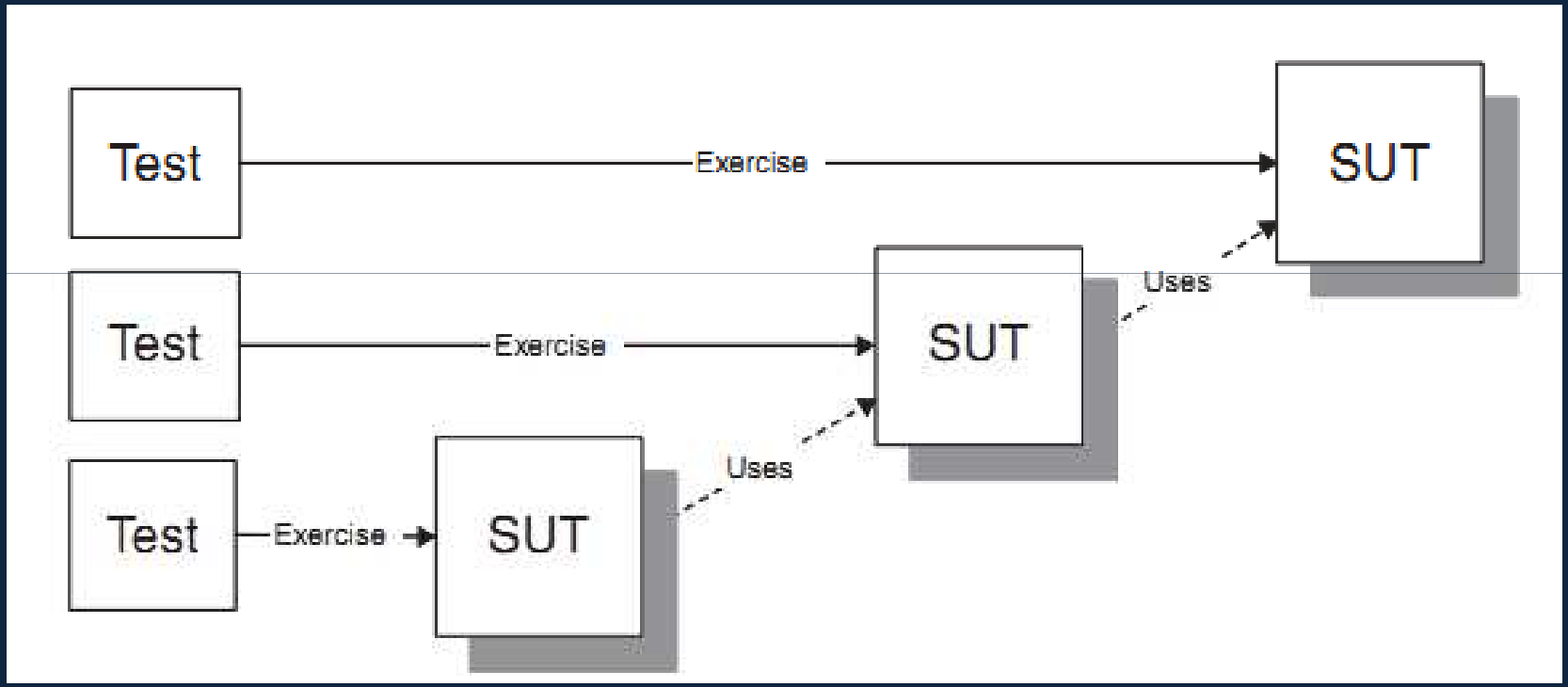
- Validar se a construção dos componentes segue a especificação

- Não exige a execução de artefatos

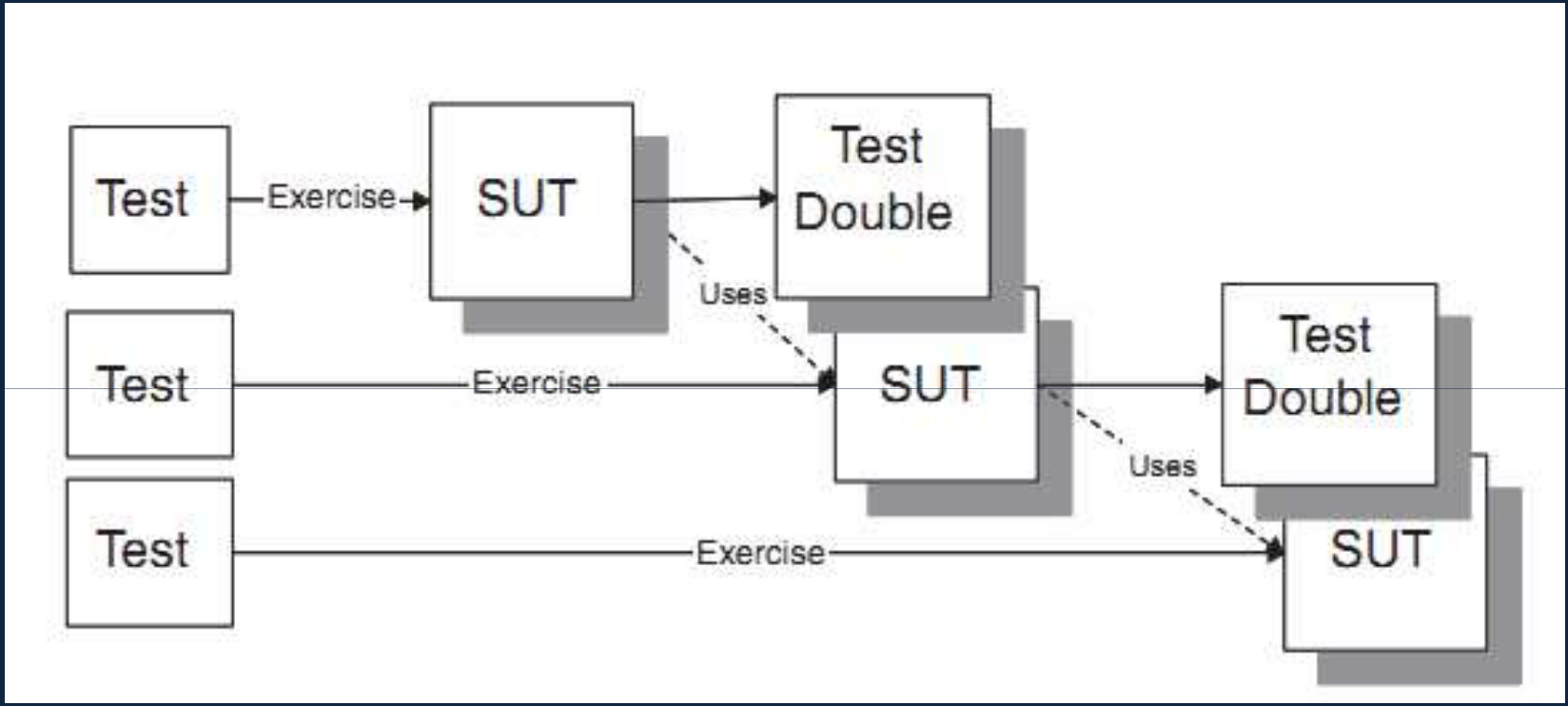
Conceitos preliminares



Top-Down/Inside-out



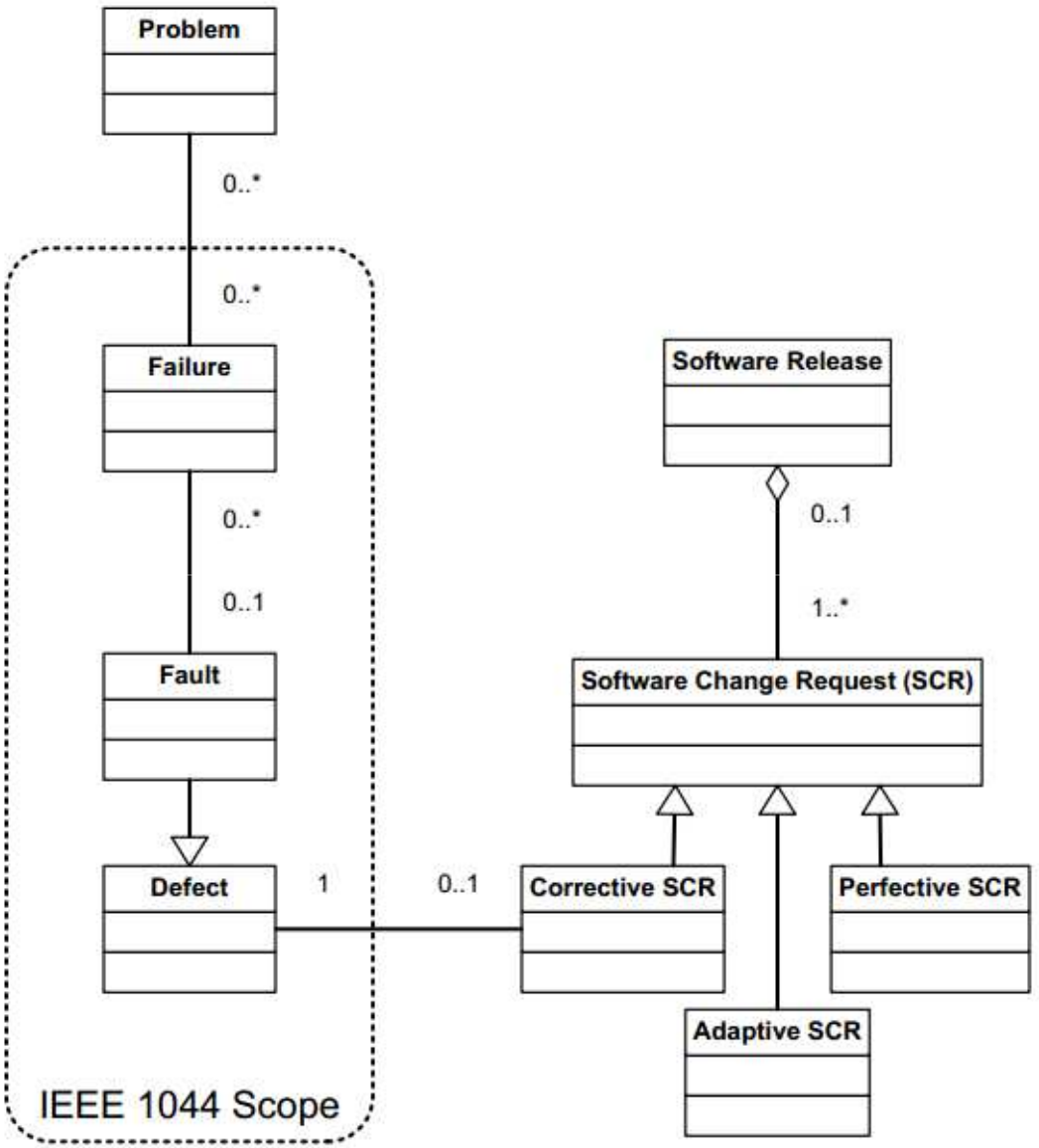
Bottom-up/Outside-in



Uso de Mock, Stubs, Drivers (Controllers), Dummy Objects , Fakes Objects

Taxonomia

Um Defeito é uma Erro, se ele é for encontrado Durante a execução do software.



Ciclo de Vida de Defeitos (IEEE 1044-2009)

Atividades de Verifivação & Validação (V&V)

São divididas em dois grupos básicos

(1) Atividades Estáticas

- Não envolve a execução do SUT

- Inspeção
- Inspeção de Código/Modelos
- Revisão
- Walkthrough (Passo a Passo)
- Checklist
- Programação em Pares (XP)

Table 3.1: Inspection Error Checklist Summary, Part I	
Data Reference	**Computation**
1. Unset variable used?	1. Computations on nonarithmetic variables?
2. Subscripts within bounds?	2. Mixed-mode computations?
3. Non integer subscripts?	3. Computations on variables of different lengths?
4. Dangling references?	4. Target size less than size of assigned value?
5. Correct attributes when aliasing?	5. Intermediate result overflow or underflow?
6. Record and structure attributes match?	6. Division by zero?
7. Computing addresses of bit strings? Passing bit-string arguments?	7. Base-2 inaccuracies?
8. Based storage attributes correct?	8. Variable's value outside of meaningful range?
9. Structure definitions match across procedures?	9. Operator precedence understood?
10. Off-by-one errors in indexing or subscripting operations?	10. Integer divisions correct?
11. Are inheritance requirements met?	
Data Declaration	**Comparison**

Atividades de Verifivação & Validação (V&V)

(2) Atividades Dinâmicas

- Atividades de Teste

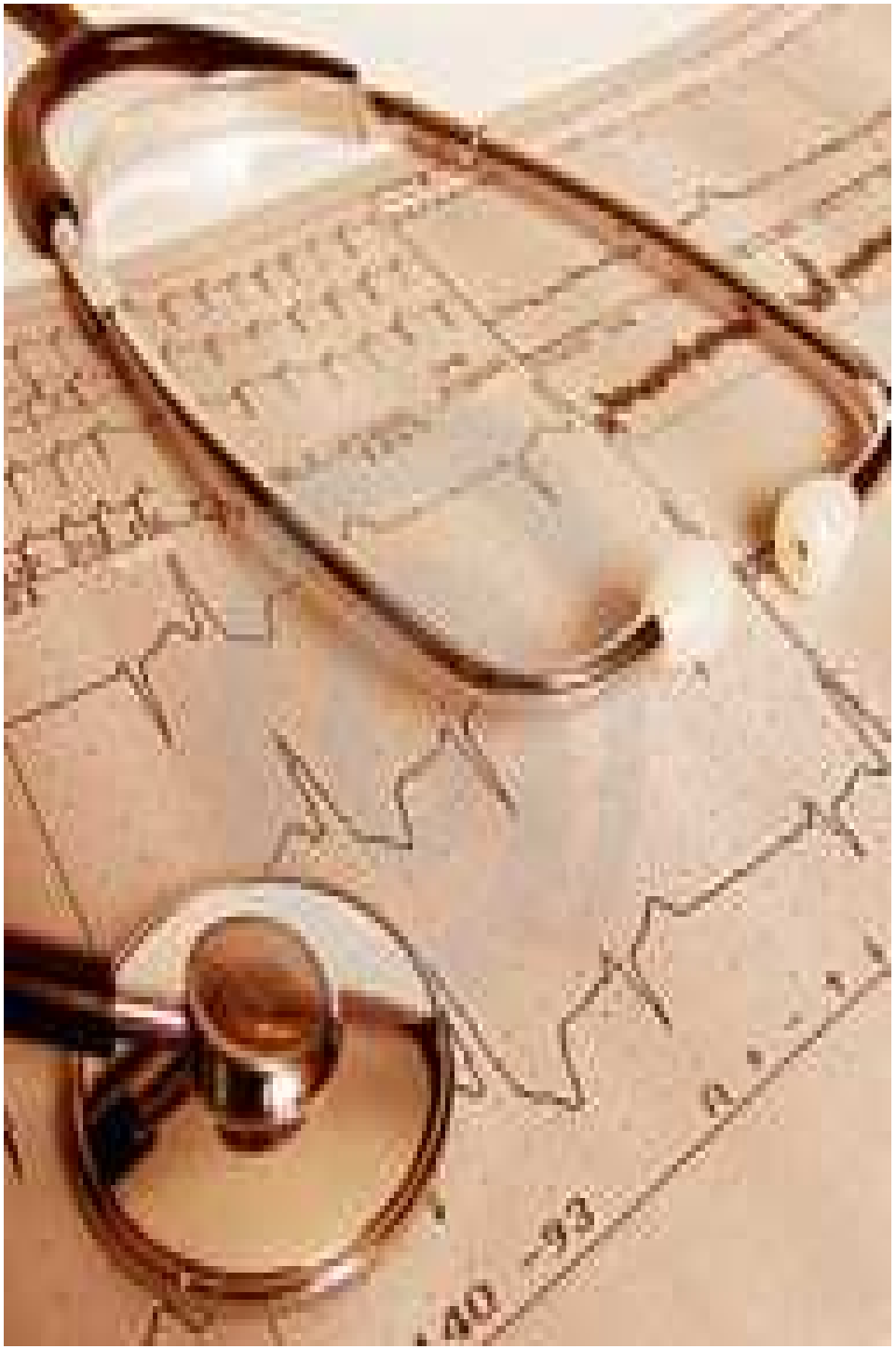
- Envolve a execução do artefato

- Exercita caminhos (*paths*)

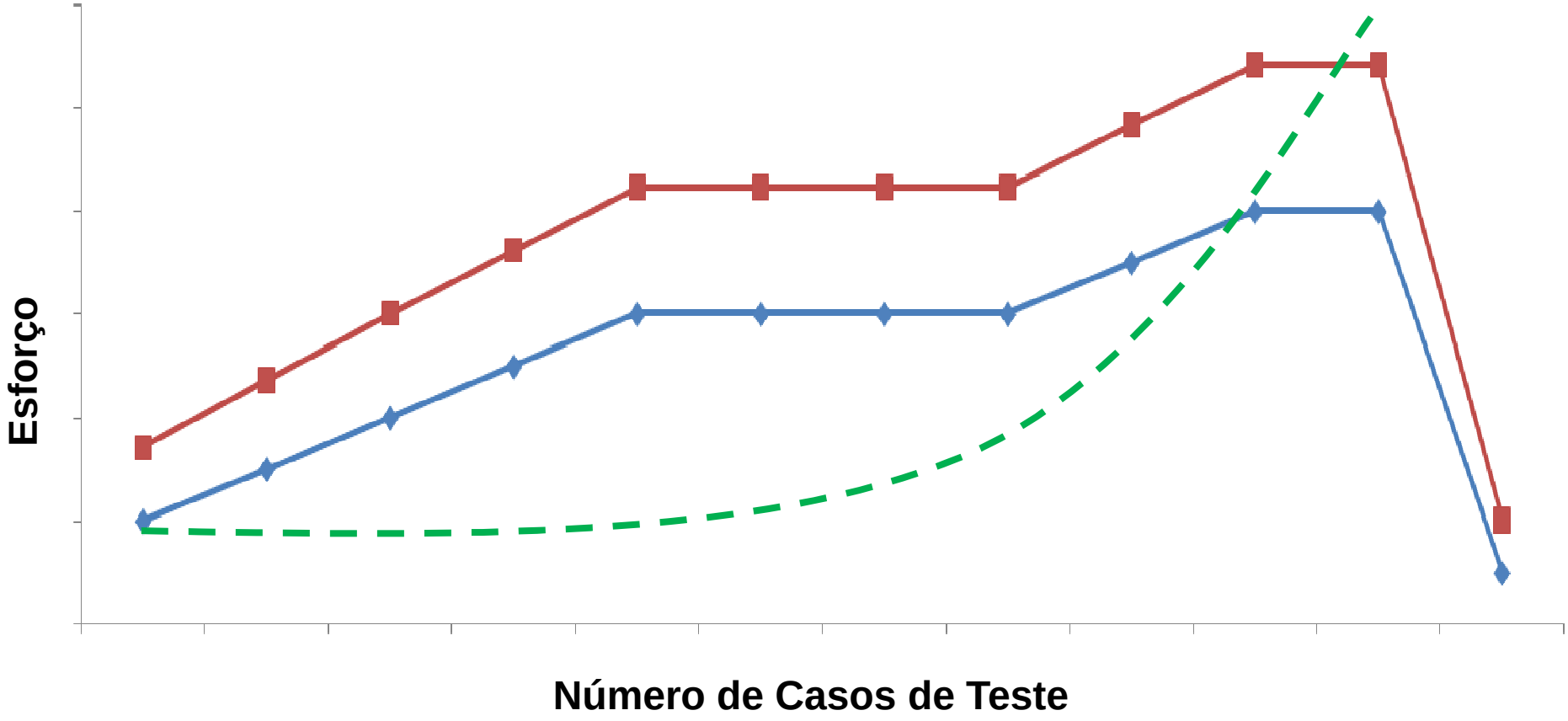
"O mais importante na atividade de teste é o projeto e criação efetivo do caso de teste"

Myers

O melhor teste é aquele que descobre mais problemas

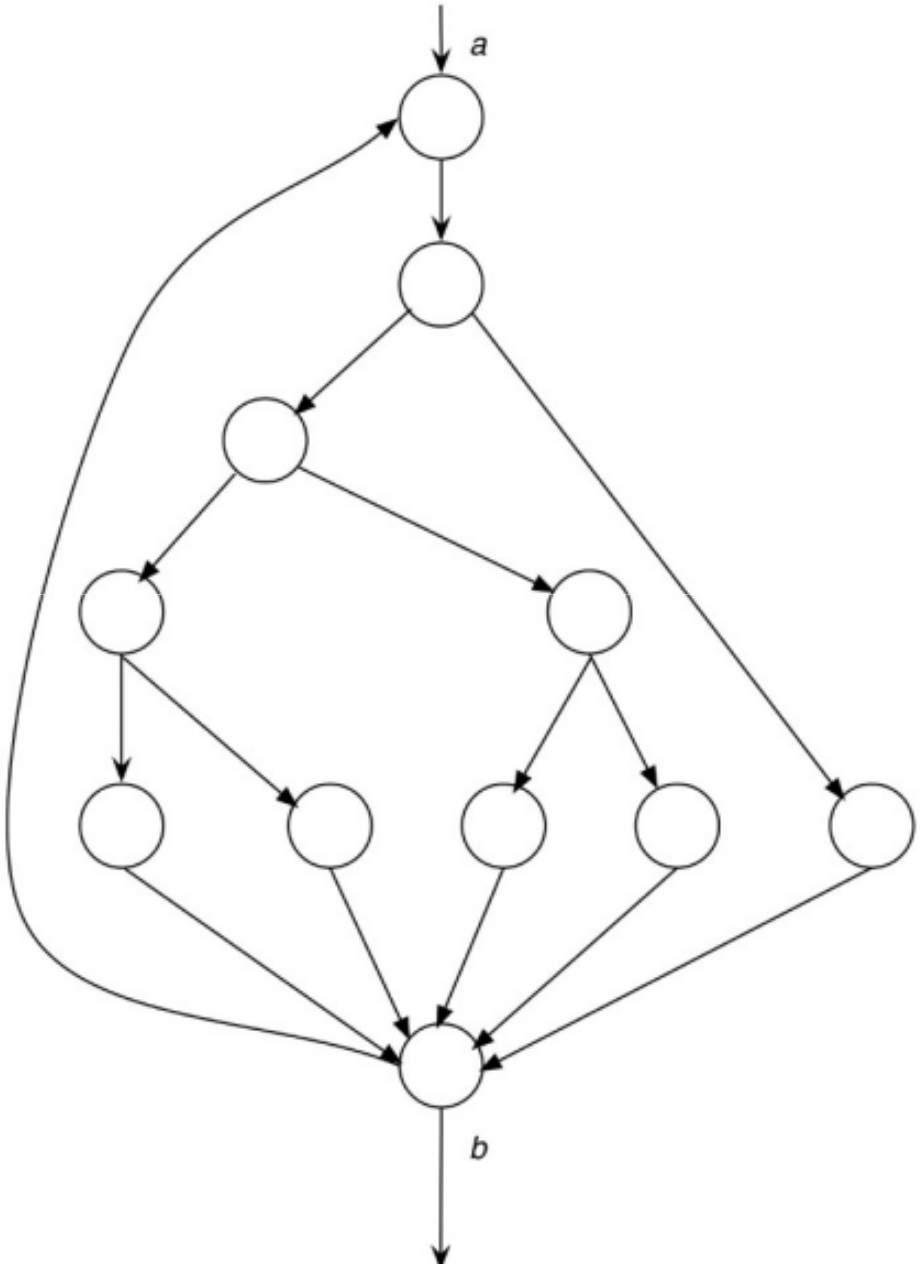


A Economia de uma Atividade de Teste



"it is impractical, often impossible, to find all the errors in a program"

Myers


a
b

100 trilhões de instruções

3,2 milhões de anos

Table 2.1: Vital Program Testing Guidelines

Principle Number	Principle
1	A necessary part of a test case is a definition of the expected output or result.
2	A programmer should avoid attempting to test his or her own program.
3	A programming organization should not test its own programs.
4	Thoroughly inspect the results of each test.
5	Test cases must be written for input conditions that are invalid and unexpected, as well as for those that are valid and expected.
6	Examining a program to see if it does not do what it is supposed to do is only half the battle; the other half is seeing whether the program does what it is not supposed to do.
7	Avoid throwaway test cases unless the program is truly a throwaway program.
8	Do not plan a testing effort under the tacit assumption that no errors will be found.
9	The probability of the existence of more errors in a section of a
10	Testing is an extremely creative and intellectually challenging task.

Principle 9: The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section.

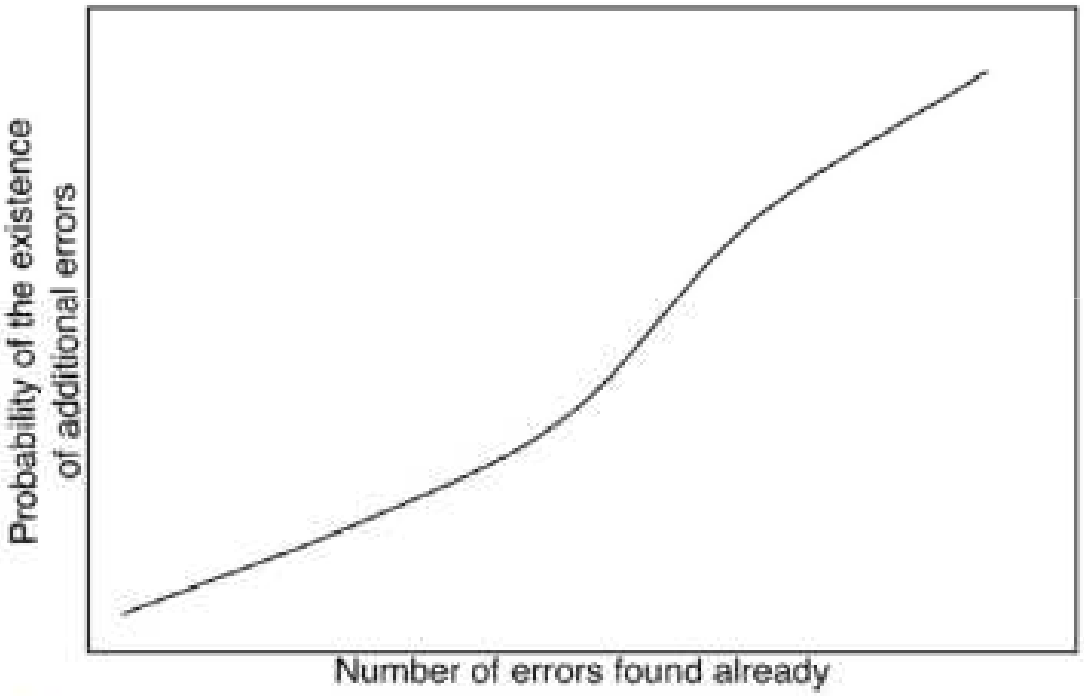


Figure 2.2: The Surprising Errors Remaining/Errors Found Relationship.

Principais Atividades de Teste

Caixa Preta (Teste Funcional)

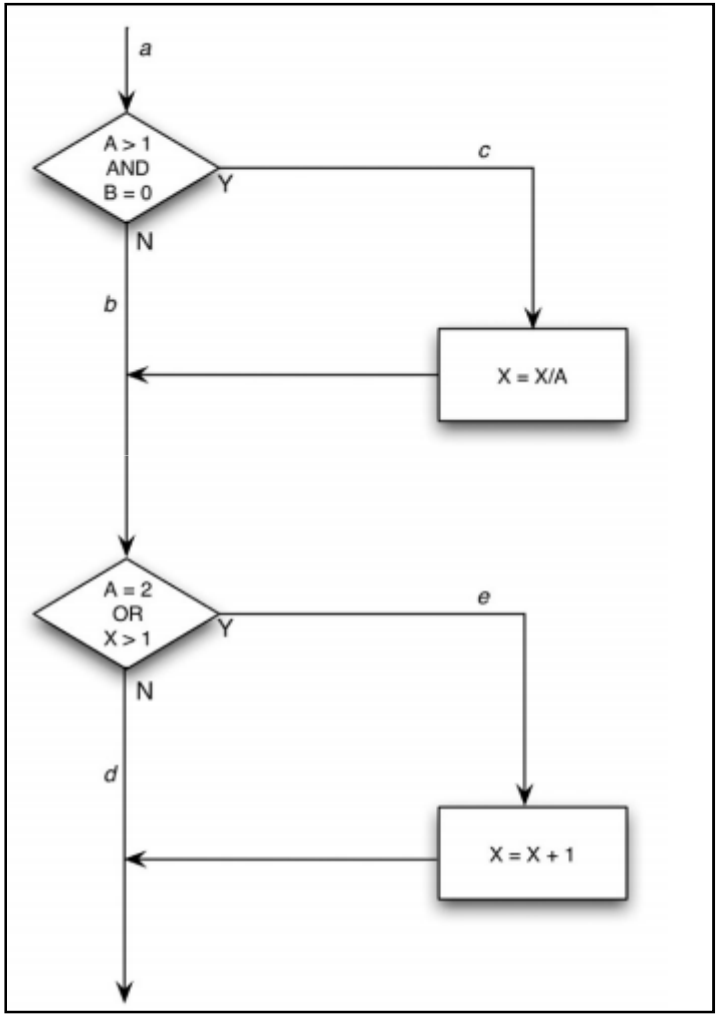
- Critérios
  - Particionamento de Equivalencia
  - Análise do Valor Limite
  - Grafo de Causa-Efeito
  - Erro de Advinhação

Caixa Branca (Teste Estrutural)

- Critérios
  - Statement Coverage (SC)
  - Decision Coverage (DC)
  - Condition Coverage (CC)
  - Decision-Condition coverage (DCC)
  - Multiple-Condition coverage (MC)
  - MC/DC

Teste Caixa Branca – White Box Testing

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


    }
```



QUIZ

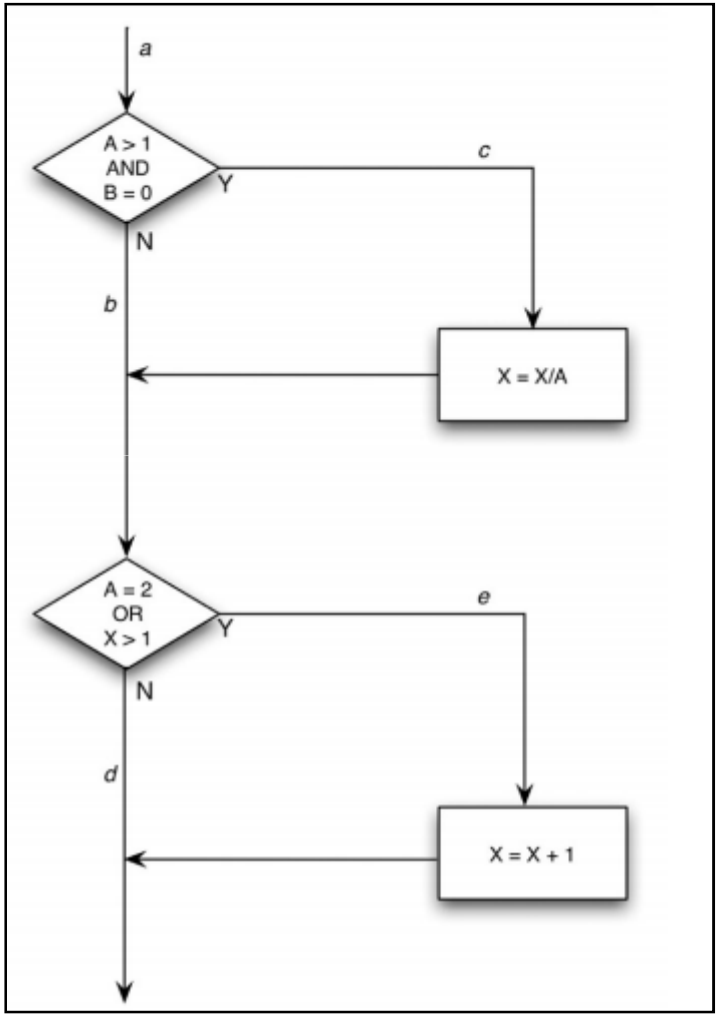
Podemos executar todas as instruções de um programa com loops ?

Teste Caixa Branca – White Box Testing

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

caso de teste

(Entrada; Saída Esperada)

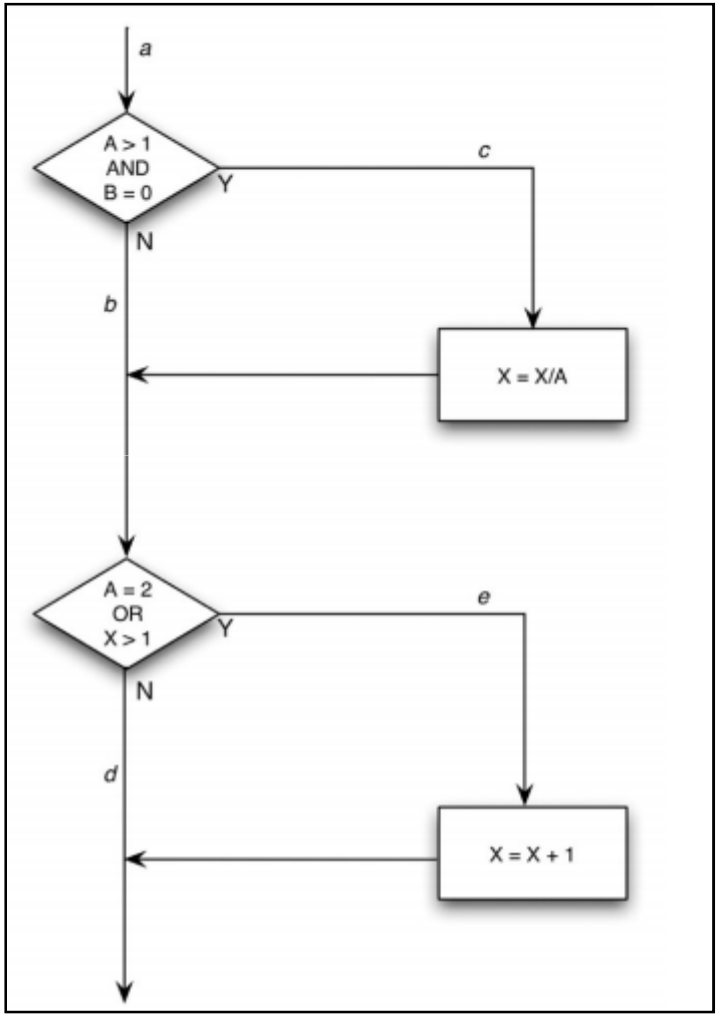


Teste Caixa Branca – White Box Testing

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

caso de teste

(A=2,B=0,X=3; 2.5)

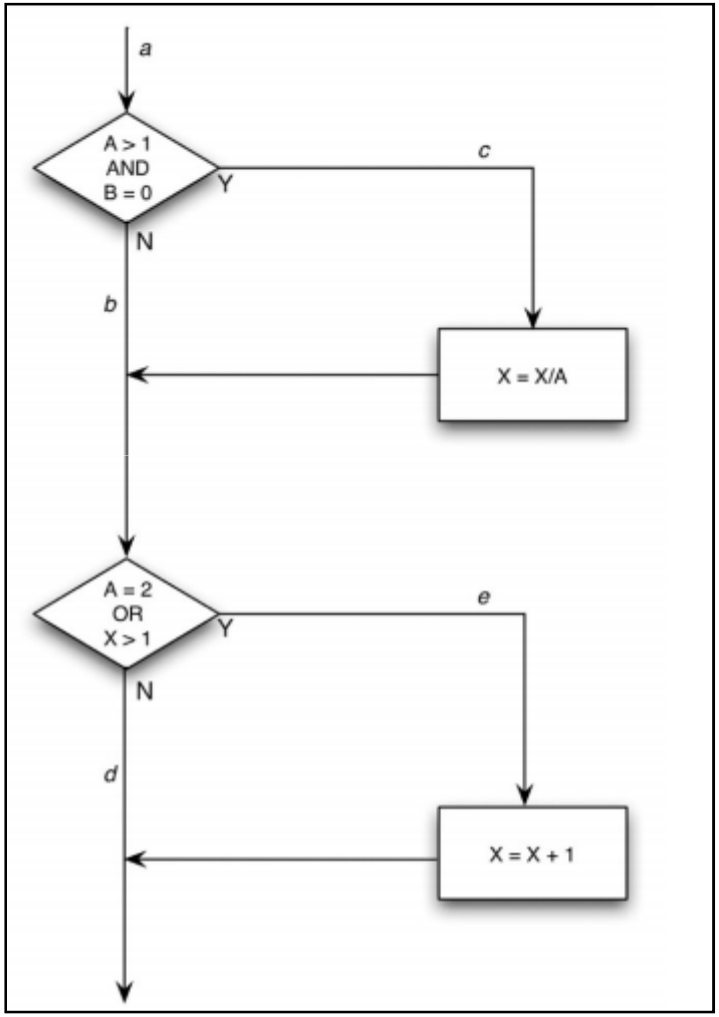


Teste Caixa Branca – Cobertura de Instrução

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

caso de teste

(A=2,B=0,X=3; 2.5)

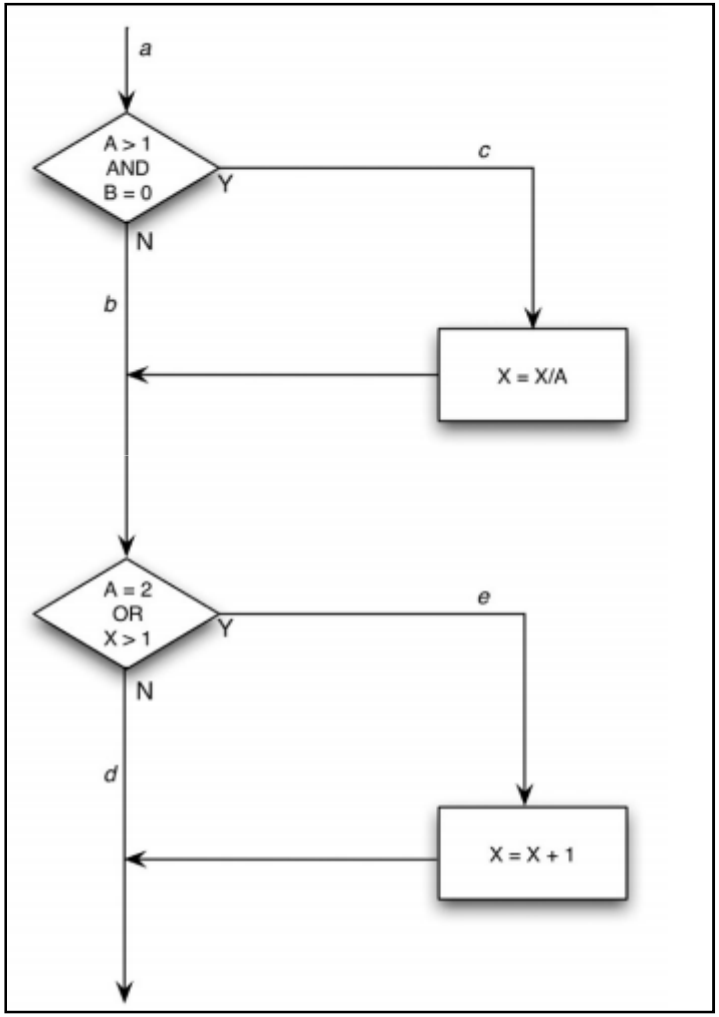


Teste Caixa Branca – Cobertura de Instrução

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

caso de teste

(A=2,B=0,X=3; 2.5)

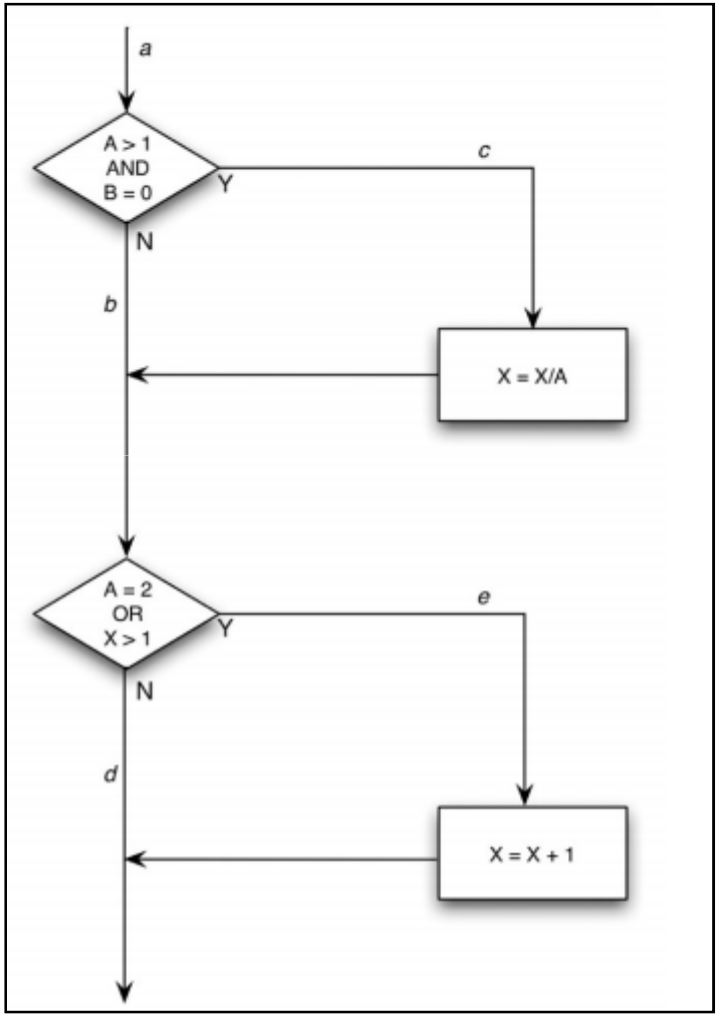


Teste Caixa Branca – Cobertura de Instrução

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

caso de teste

(A=2,B=0,X=3; 2.5)

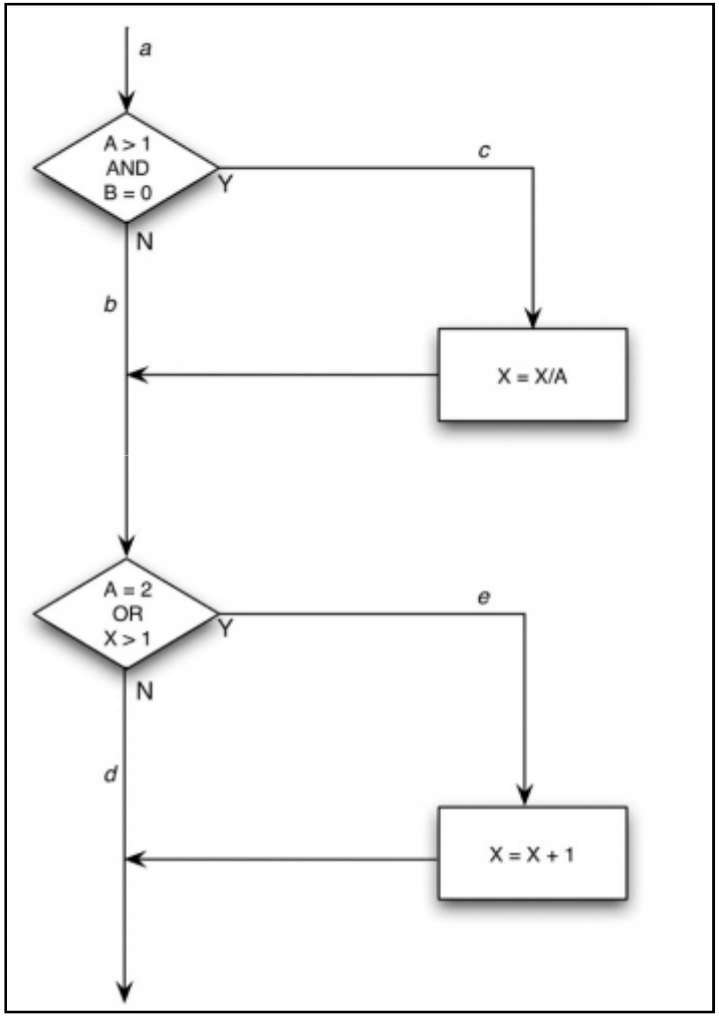


Teste Caixa Branca – Cobertura de Instrução

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;
    }
    if (a==2 || x>1) {
        x=x+1;
    }
}
```

caso de teste

(A=2,B=0,X=3; 2.5)



QUIZ

O critério de cobertura de instrução pode ser considerado um critério forte ?

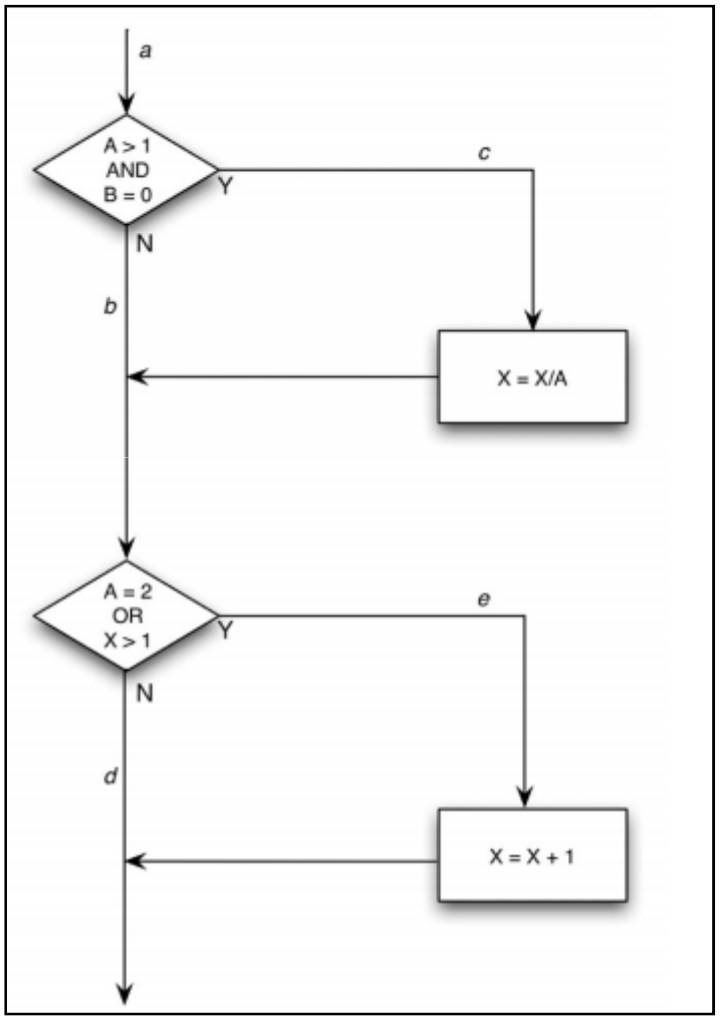
Teste Caixa Branca – Cobertura de Instrução

Defeito

```
public vo  foo(int a, int b, int x) {
    if (a>1 || b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


    }
```

caso de teste

(A=2,B=0,X=3; 2.5)

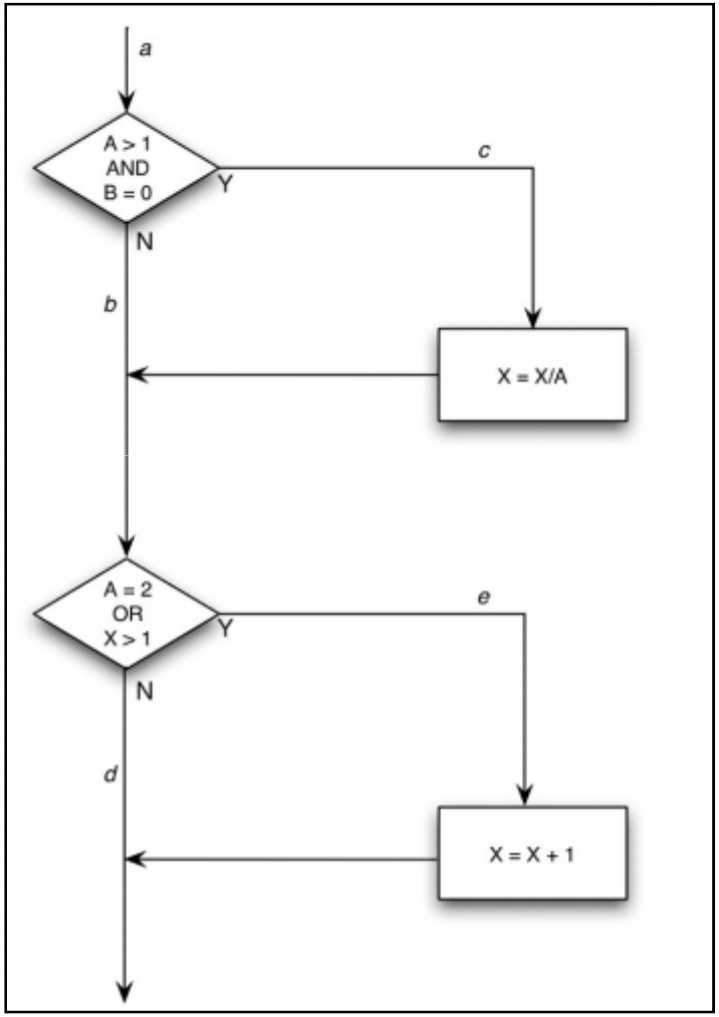


Teste Caixa Branca – Cobertura de Instrução

```
public void foo(int a, int b, int x) {
    if (a>1 || b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

caso de teste

(A=2,B=0,X=3; 2.5)

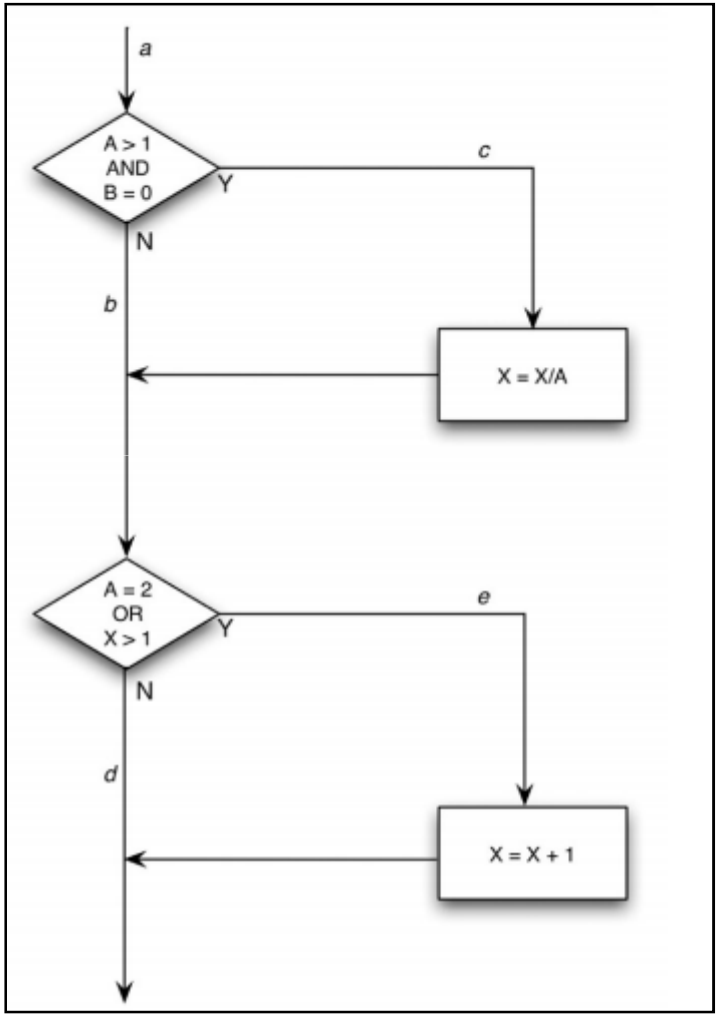


Teste Caixa Branca – Cobertura de Instrução

```
public void foo(int a, int b, int x) {
    if (a>1 || b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

caso de teste

(A=2,B=0,X=3; 2.5)

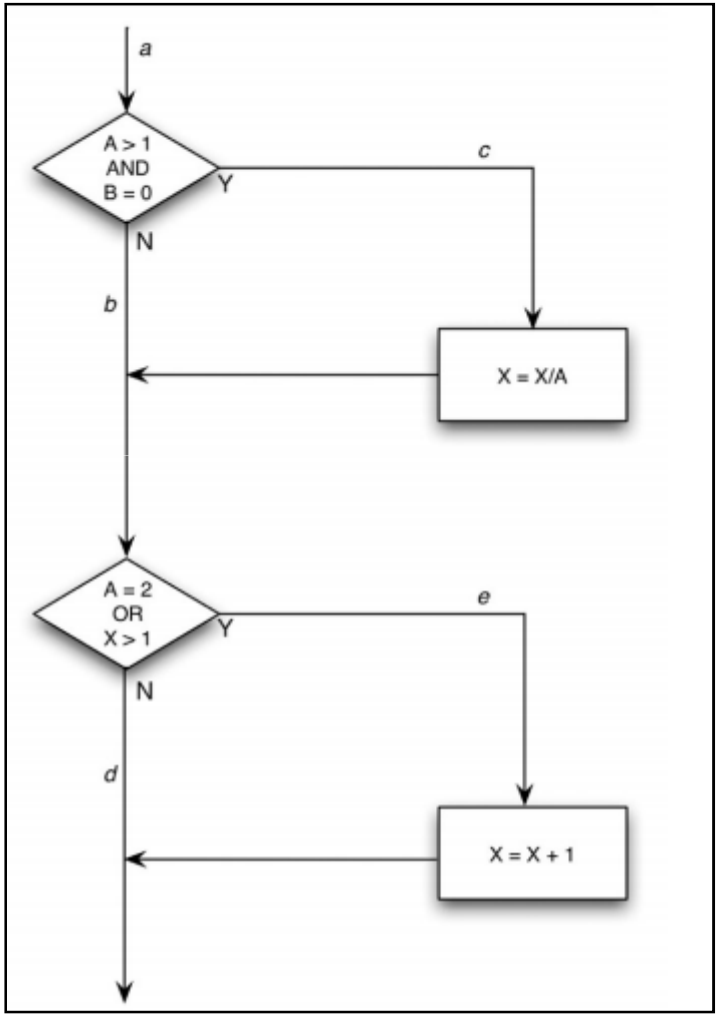


Teste Caixa Branca – Cobertura de Instrução

```
public void foo(int a, int b, int x) {
    if (a>1 || b==0) {
        x=x/a;
    }
    if (a==2 || x>1) {
        x=x+1;
    }
}
```

caso de teste

(A=2,B=0,X=3; 2.5)



Teste Caixa Branca – Cobertura de Instrução

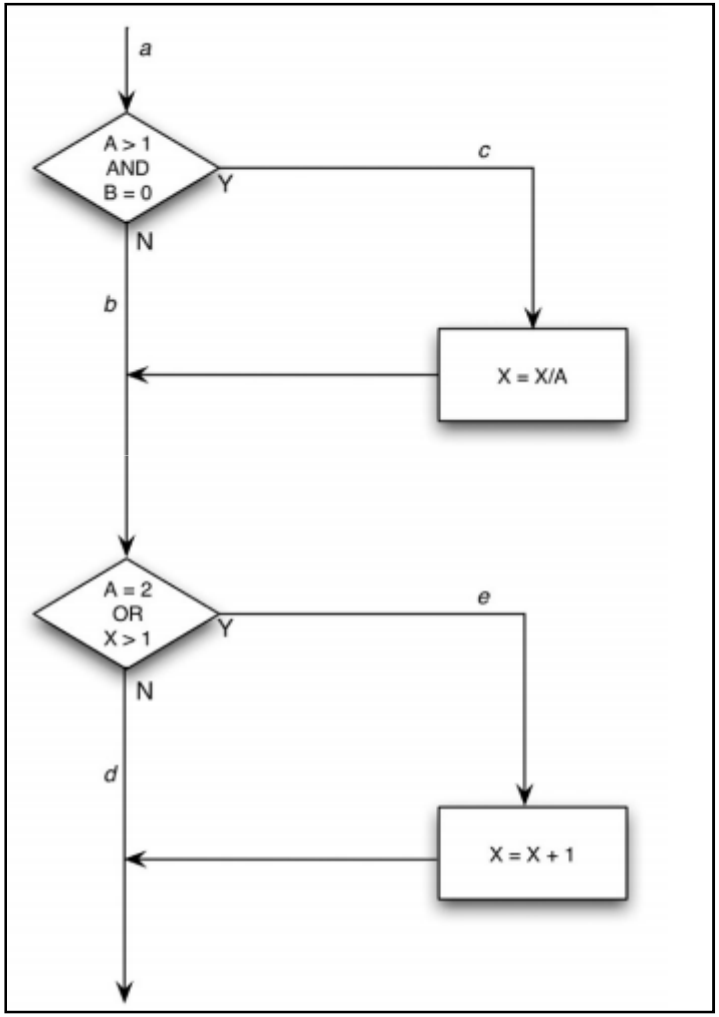
```
public void foo(int a, int b, int x) {
    if (a>1 || b==0) {
        x=x/a;
    }
    if (a==2 || x>1) {
        x=x+1;
    }
}
```

caso de teste

(A=2,B=0,X=3; 2.5)

Defeito NUNCA detectado !!!

Critério Fraco



Particionamento de Equivalência

External condition	Valid equivalence classes	Invalid equivalence classes

DIMENSION *ad*[,ad]...

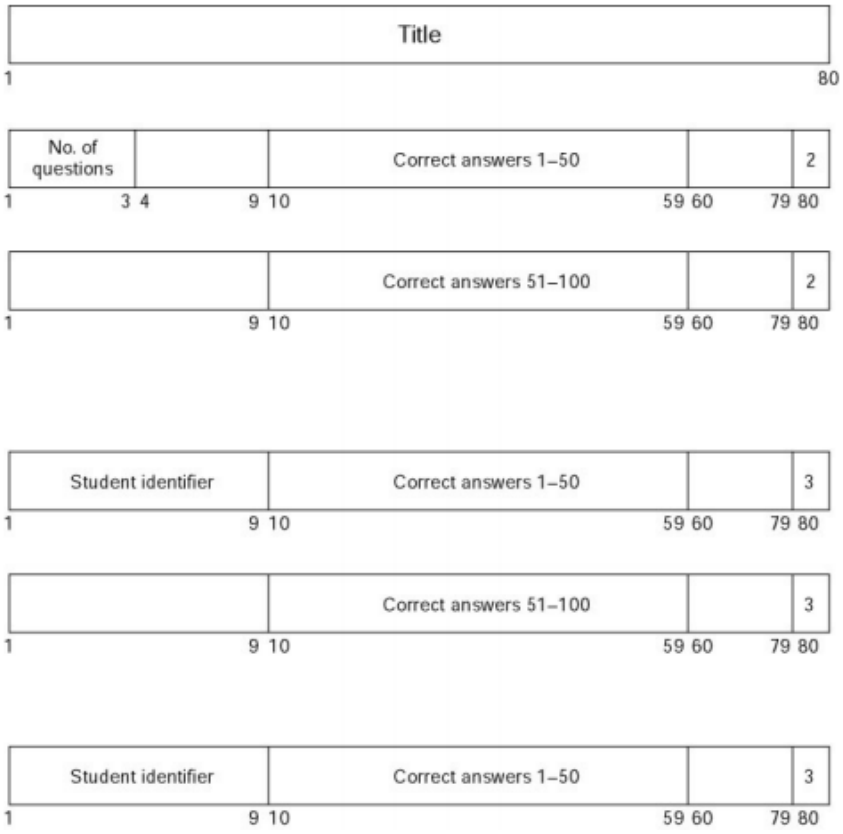
n(d[,d]...)

Table 4.1: Equivalence Classes

Input Condition	**Valid Equivalence Classes**	**Invalid Equivalence Classes**
Number of array descriptors	one (1), > one (2)	none (3)
Size of array name	1-6 (4)	0 (5), > 6 (6)
Array name	has letters (7), has digits (8)	has something else (9)
Array name starts with letter	yes (10)	no (11)
Number of dimensions	1ñ7 (12)	0 (13), > 7 (14)
Upper bound is	constant (15), integer variable (16)	array element name (17), something else (18)
Integer variable name	has letter (19), has digits (20)	has something else (21)
Integer variable starts with letter	yes (22)	no (23)
Constant	- 65534ñ65535 (24)	= 65534 (25), > 65535 (26)
Lower bound specified	yes (27), no (28)	
Upper bound to lower bound	greater than (29), equal (30)	less than (31)
Specified lower bound	negative (32), zero (33), > 0 (34)	
Lower bound is	constant (35), integer variable (36)	array element name (37), something else (38)
Multiple lines	yes (39), no (40)	

Análise do Valor Limite

Explora valores no limite de do domínio



1. **Arquivo vazio**
2. **Título Faltando**
3. **1 caracter para o título**
4. **80 caracters para o título**
5. **1 questão de exemplo**
6. **50 questões de exemplo**
7. **51 questões de exemplo**
8. **999 questões de exemplo**
9. **Uma questão vazia**

Error Guessing (Erro de Adivinhação)

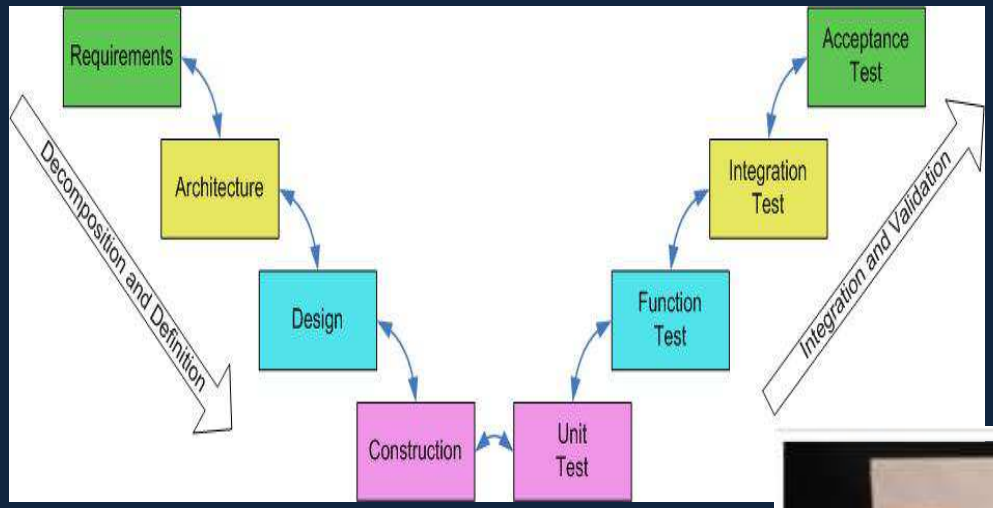
Explora erros típicos em um projeto.

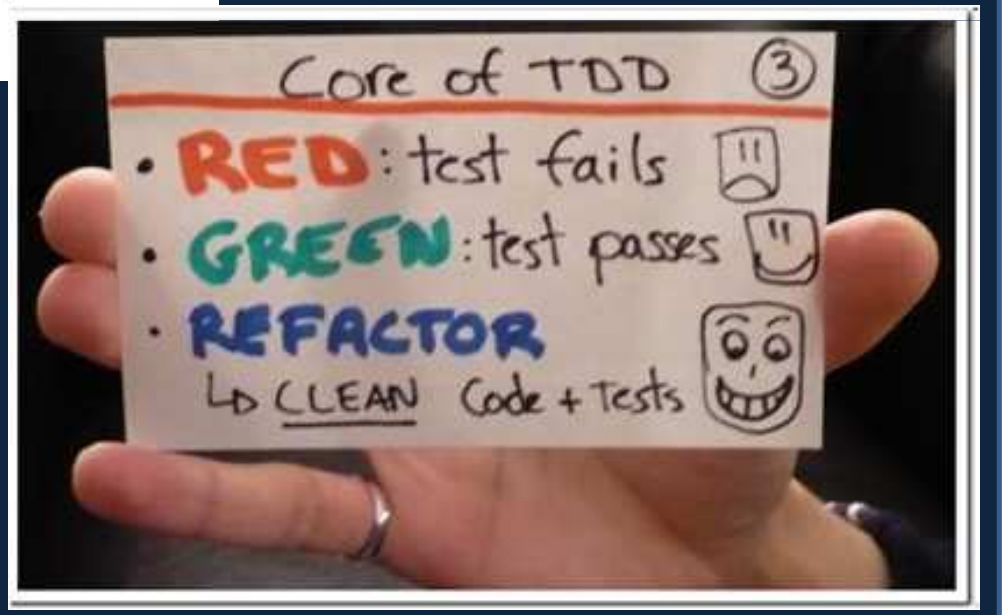
Error Guessing (Erro de Adivinhação)

1. **A lista de entrada é vazia**
2. **O nome das variáveis foram definidas com clareza**
3. **Todas as entradas foram salvas**

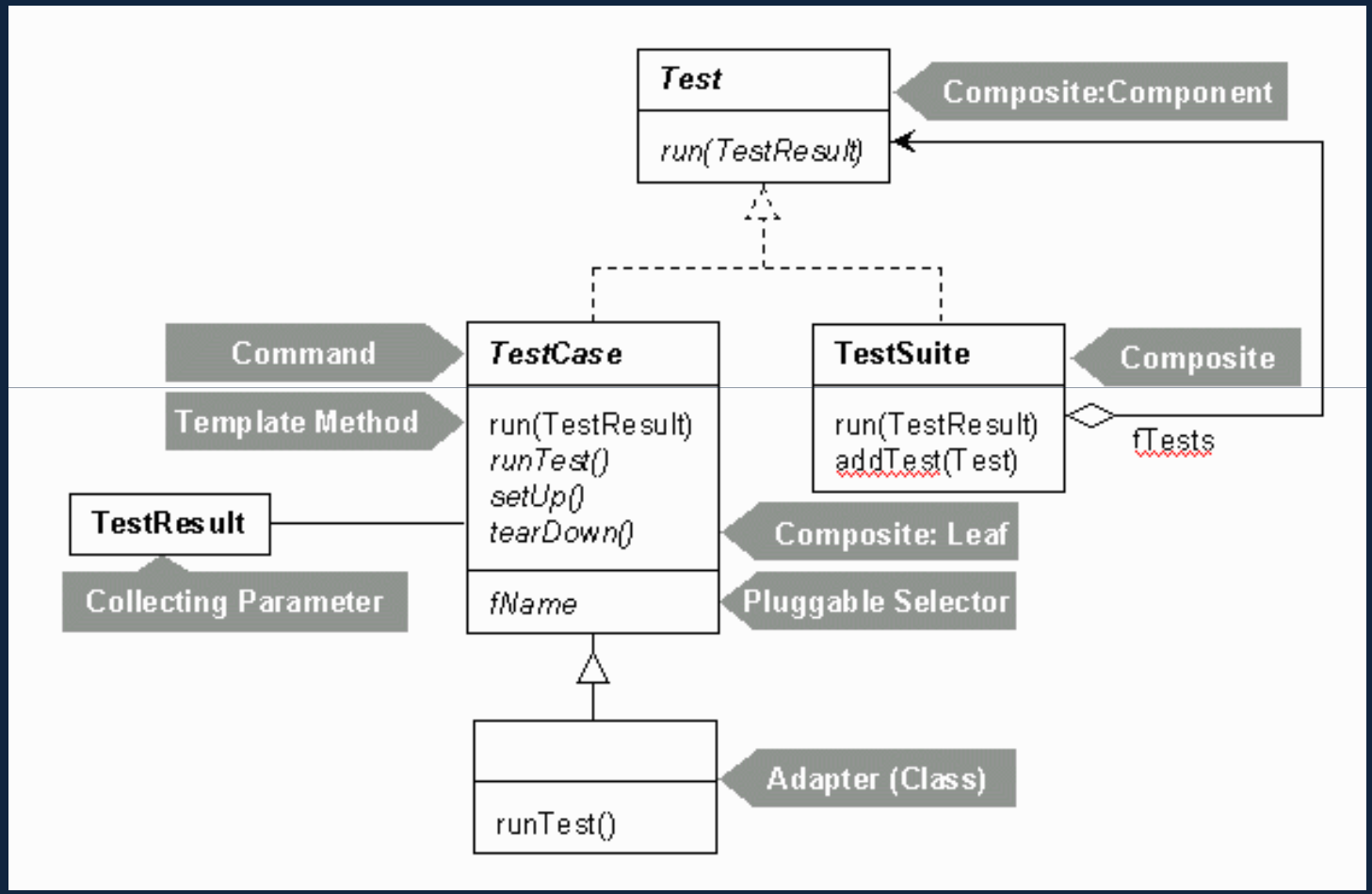
...
...

Processo de Teste Tradicional e Ágil





jUnit (Zoo de Patterns)



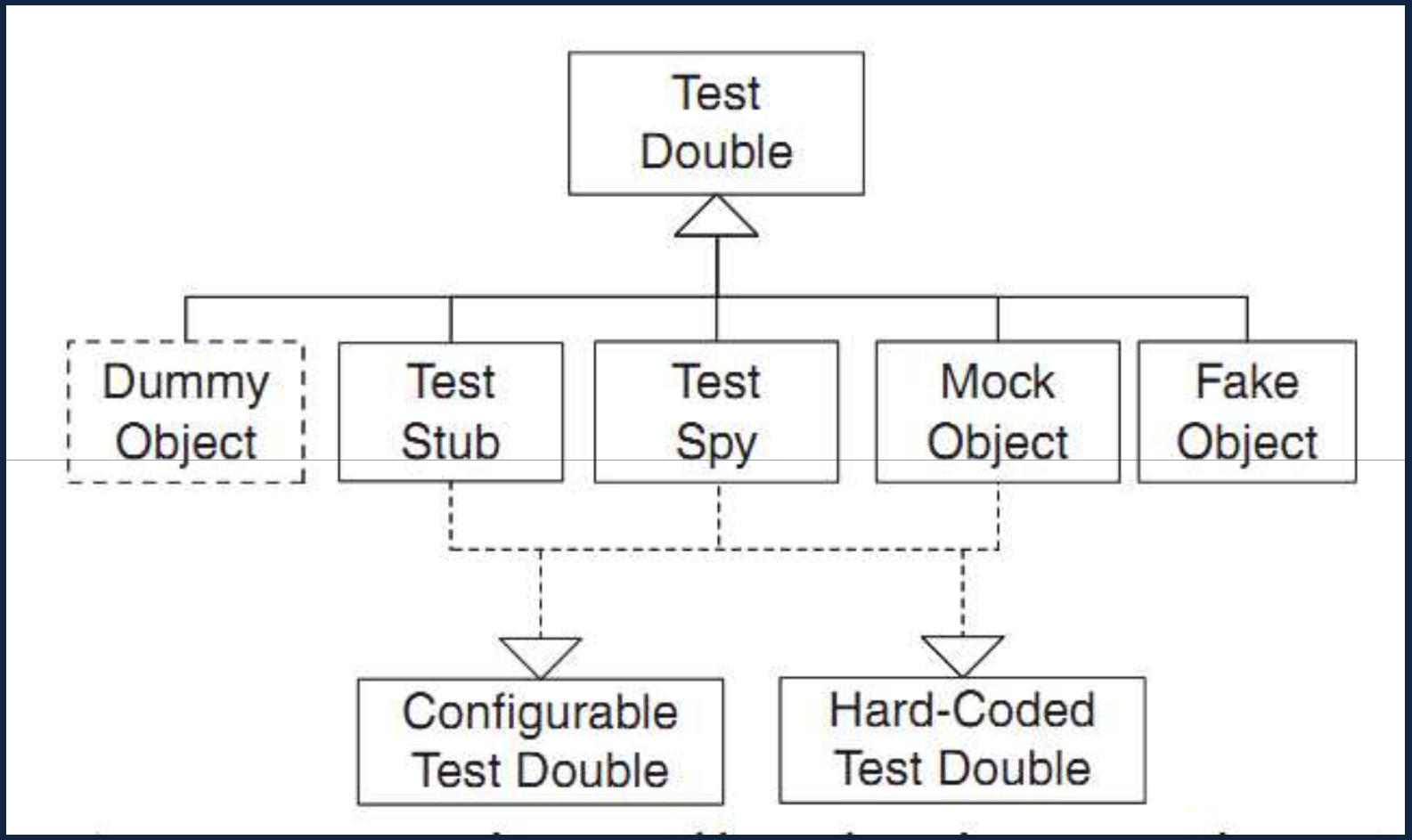
xUnit

Tool		Book Term							
Language	xUnit Member	Testcase Class	Test Suite Factory	Test Method	Fixture setup	Fixture teardown	Suite Fixture Setup	Suite Fixture Teardown	Expected Exception Test
Java 1.4	JUnit 3.8.2	Subclass of TestCase	static suite()	testXxx()	setUp()	tearDown()	Not applicable	Not applicable	Subclass of Expected Exception Test
Java 5	JUnit 4.0+	import org.junit.Test	static suite()	@Test	@Before	@After	@Before Class	@After Class	@Exception
.NET	CsUnit	[TestFixture]	[Suite]	[Test]	[SetUp]	[TearDown]	Not applicable	Not applicable	[Expected Exception()]
.NET	NUnit 2.0	[TestFixture]	[Suite]	[Test]	[SetUp]	[TearDown]	Not applicable	Not applicable	[Expected Exception()]
.NET	NUnit 2.1+	[TestFixture]	[Suite]	[Test]	[SetUp]	[TearDown]	[Test Fixture SetUp]	[TestFixture TearDown]	[Expected Exception()]
.NET	MbUnit 2.0	[TestFixture]	[Suite]	[Test]	[SetUp]	[TearDown]	[Fixture Setup]	[Fixture Teardown]	[Expected Exception()]
.NET	MSTest	[TestClass]	Not applicable	[Test Method]	[Test Initialize]	[Test Cleanup]	[Class Initialize]	[Class Cleanup]	[Expected Exception()]
PHP	PHPUnit	Subclass of TestCase	static suite()	testXxx()	setUp()	tearDown()	Not applicable	Not applicable	Subclass of Expected Exception Test

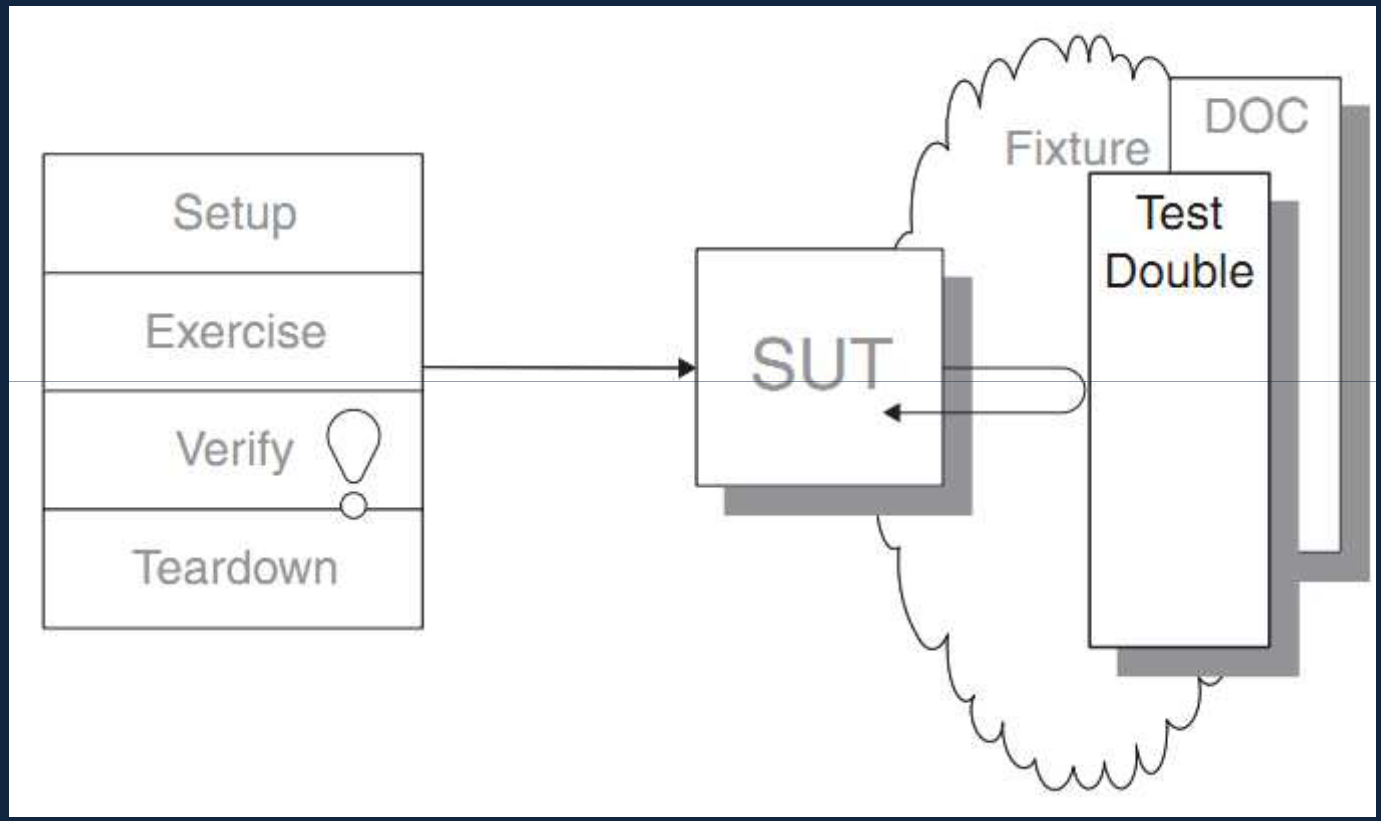
Continued...

Tool		Book Term							
Language	xUnit Member	Testcase Class	Test Suite Factory	Test Method	Fixture Setup	Fixture Teardown Teardown	Suite Fixture Test	Suite Fixture	Expected Exception
Python	PyUnit	Subclass of unittest. TestCase	Test Loader()	testXxx	setUp	tearDown	Not applicable	Not applicable	assert raise
Ruby	Test::Unit	Subclass of Test::Unit:: TestCase	Classname. suite()	testXxx()	setup()	teardown	Not applicable	Not applicable	assert_raise
Smalltalk	SUnit	Superclass: TestCase	TestSuite named:	testXxx	setUp	tearDown	To be determined	To be determined	should:raise:
VB 6	VbUnit	Implements IFixture	Implements ISuite	TestXxx()	IFixture_ Setup()	IFixture_ TearDown	IFixture Frame_ Create()	IFixture Frame_ Destroy	on error...
SAP ABAP	ABAP Unit	FOR TESTING	Automatic	Any	setup	teardown	class_setup	class_ teardown	To be determined

Terminologia básica



Pattern - Setup/Exercise/Verify/Teardown



Pattern – Arrange/Act/Assert (3A's)

Como organizar os casos de teste ?

Arrange – Inicialização de todas as pré-condições
Act – Um objeto ou método sob teste
Assert – Resultados esperados

Vantagens

- Auxilia na compreensão
- Melhora a manutenção
- Auxilia na detecção de Smells

Pattern – Arrange/Act/Assert (3A's)

Como organizar os casos de teste ?

Arrange – Inicialização de todas as pré-condições
Act – Um objeto ou método sob teste
Assert – Resultados esperados

Vantagens

- Auxilia na compreensão
- Melhora a manutenção
- Auxilia na detecção de Smells

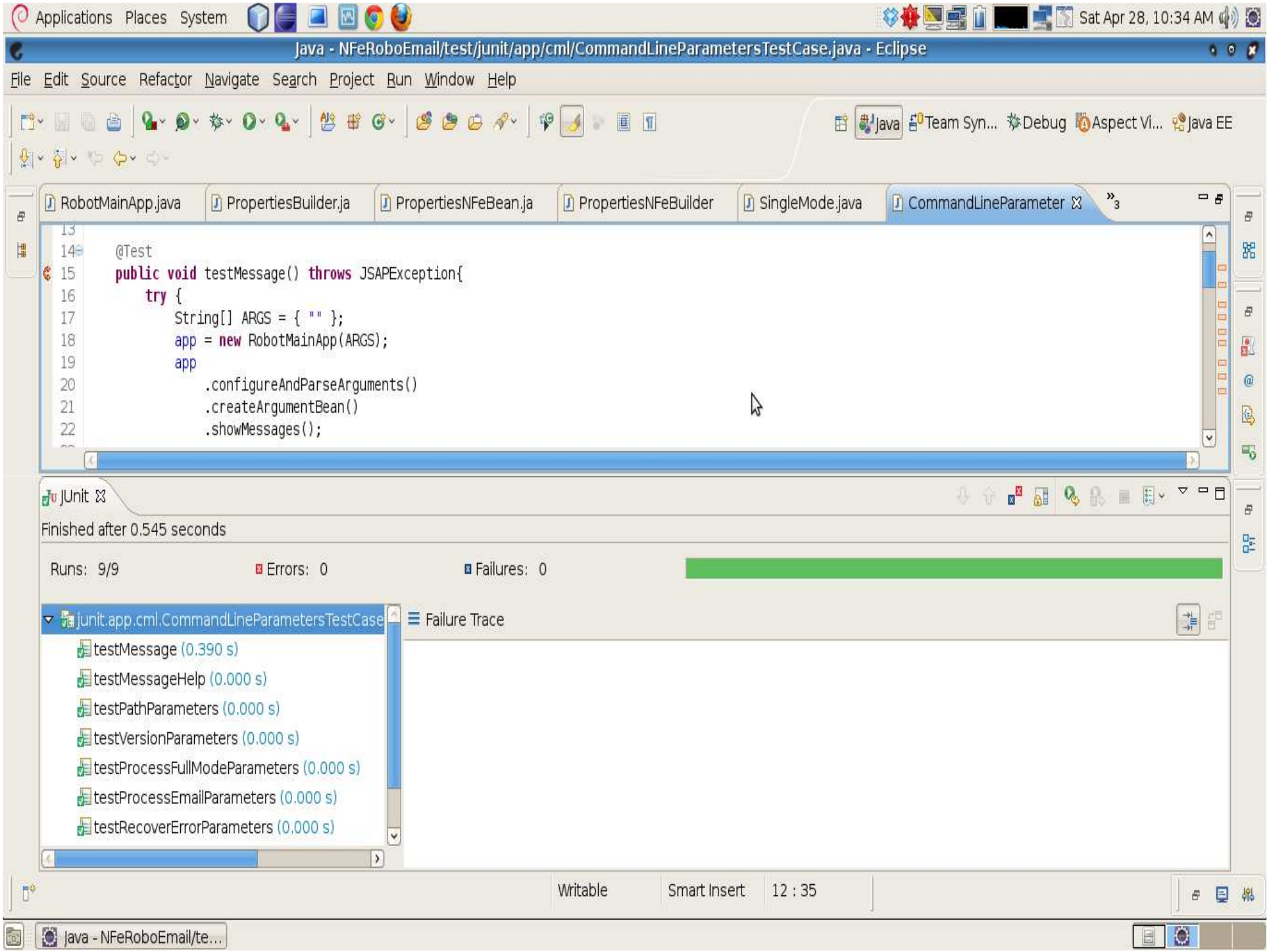
Variações de nomenclatura

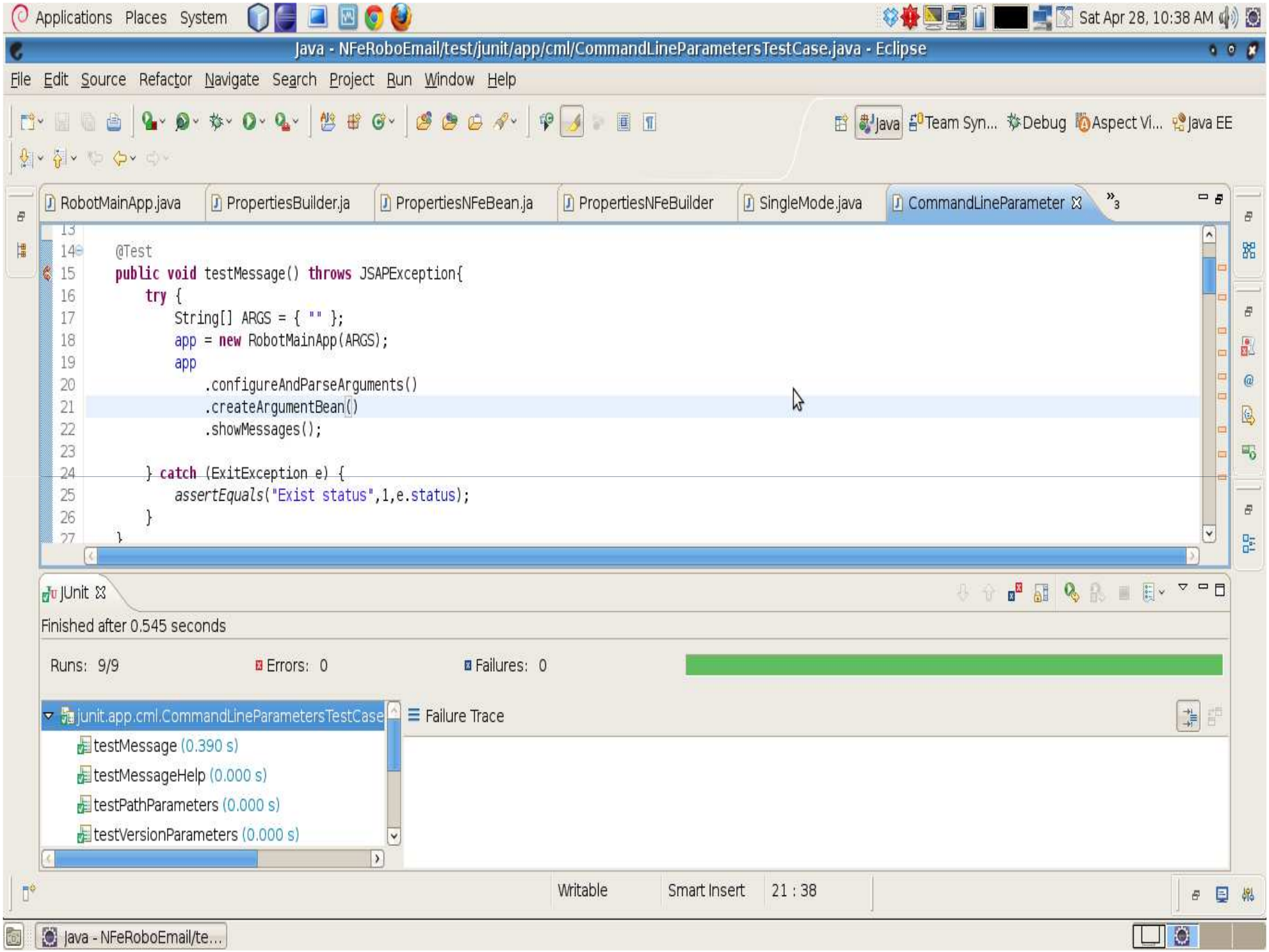
Build/Operate/Check/Cleat (BOCC Pattern)

```
import static junit.org.Assert.*;

public class MySingleTest{

@Test
   public void test() {
      String input  = "abc";
      String result = Util.reverse(input);
      assertEquals("cba", result);
   }
}

...
```

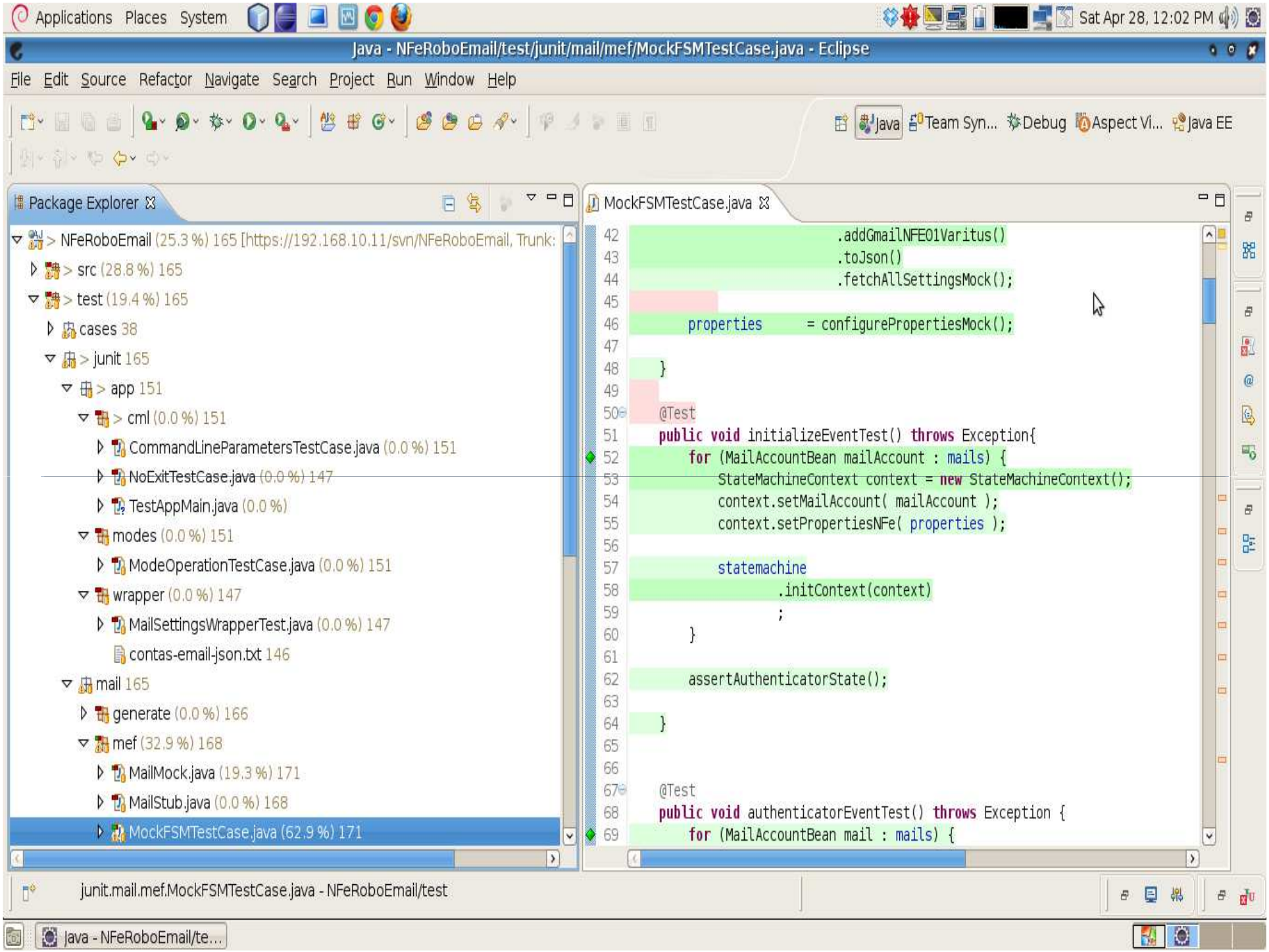
Exemplos

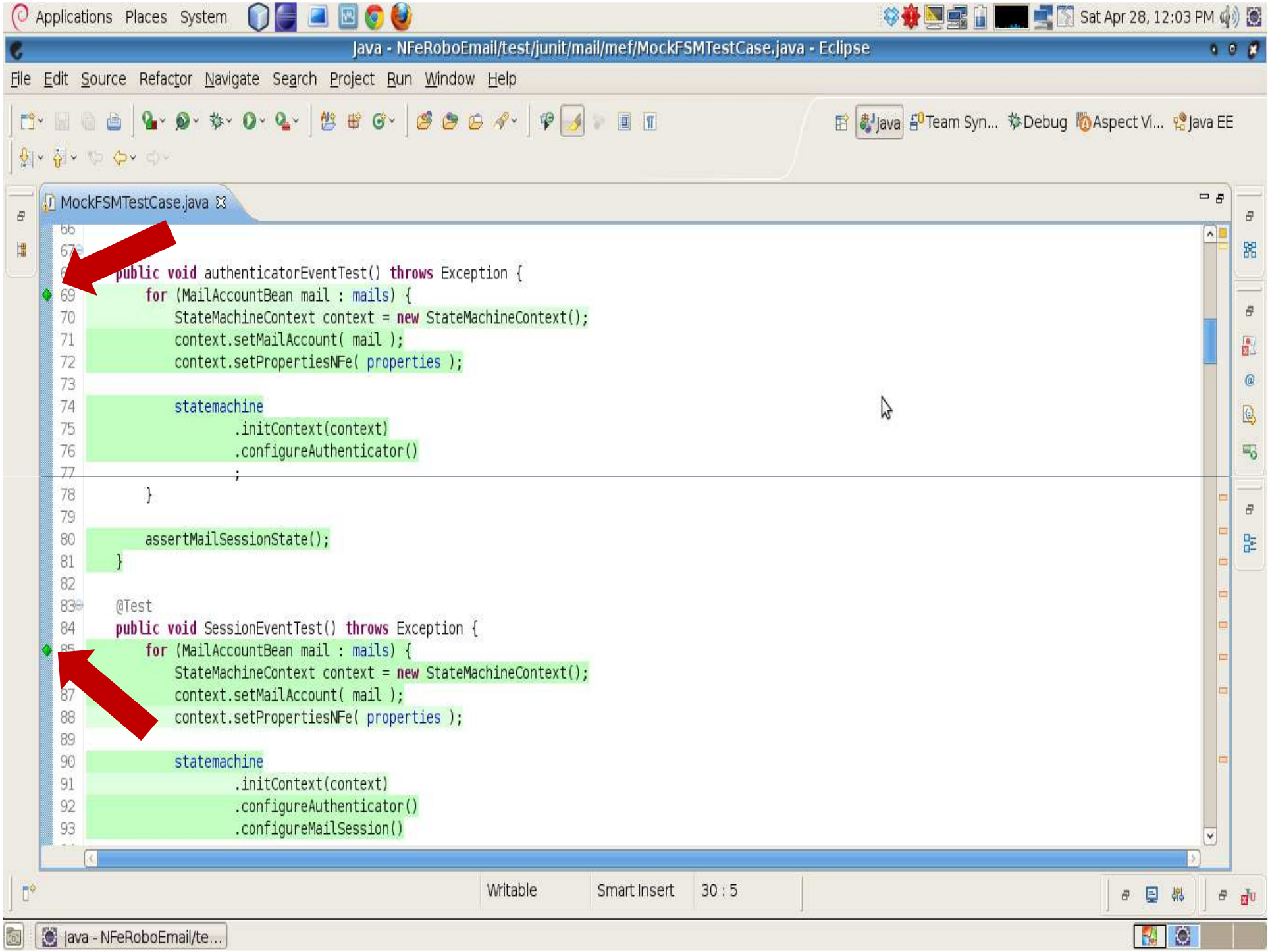
```
Usage: java br.com.varitus.nfe.app.RobotMainApp

    [(-m|--process-email) <process-email>] [(-e|--recover-error) <recover-
    error>] [(-o|--recover-other) <recover-other>] (-p|--path) <path> [-f|--
    process-full-mode] [-r|--recover-full-mode] [-v|--version] [-h|--help]


  [(-m|--process-email) <process-email>]
        E-mail da conta do cliente na base de dados da VARITUS que será
        processado

  [(-e|--recover-error) <recover-error>]
        E-mail da conta do cliente na base de dados da VARITUS. As mensagens
        dessa conta serão movidas da pasta NFE_ERRO para a caixa de entrada

  [(-o|--recover-other) <recover-other>]
        E-mail da conta do cliente na base de dados da VARITUS. As mensagens
        dessa conta serão movidas da pasta NFE_OUTRO para a caixa de entrada

  (-p|--path) <path>
        O path clientes da NF-e, esse caminho é o valor da tag configurada no
        path_cliente.properties em ~/nfe/
```

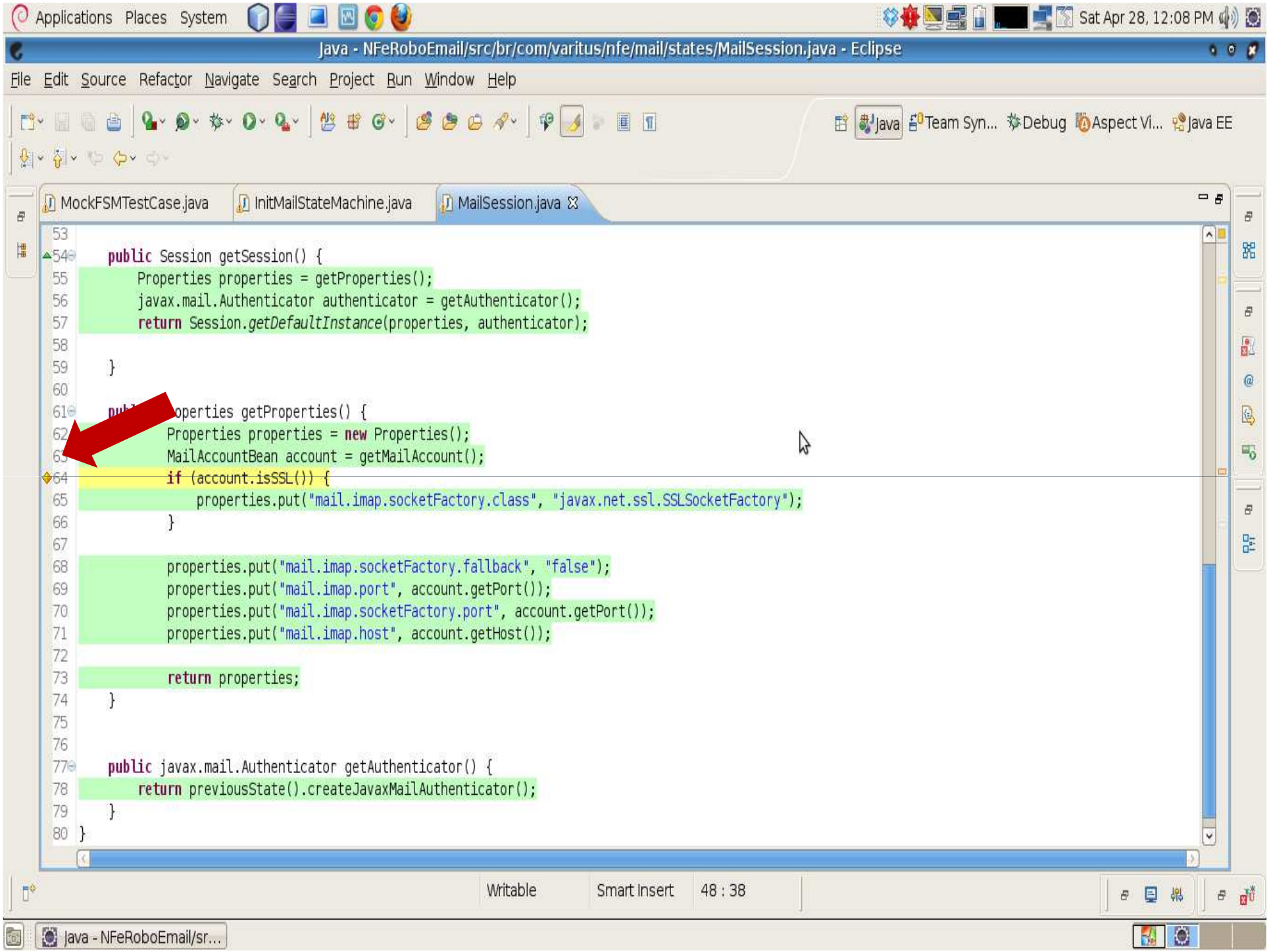

Applications Places System
Sat Apr 28, 10:34 AM
Java - NFeRoboEmail/test/junit/app/cml/CommandLineParametersTestCase.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Team Syn...
Debug
Aspect Vi...
Java EE
RobotMainApp.java
PropertiesBuilder.ja
PropertiesNFeBean.ja
PropertiesNFeBuilder
SingleMode.java
CommandLineParameter
@Test
public void testMessage() throws JSAPException{
try {
String[] ARGS = { "" };
app = new RobotMainApp(ARGS);
app
.configureAndParseArguments()
.createArgumentBean()
.showMessages();
JUnit
Finished after 0.545 seconds
Runs: 9/9
Errors: 0
Failures: 0
junit.app.cml.CommandLineParametersTestCase
testMessage (0.390 s)
testMessageHelp (0.000 s)
testPathParameters (0.000 s)
testVersionParameters (0.000 s)
testProcessFullModeParameters (0.000 s)
testProcessEmailParameters (0.000 s)
testRecoverErrorParameters (0.000 s)
Failure Trace
Writable
Smart Insert
12 : 35
Java - NFeRoboEmail/te...

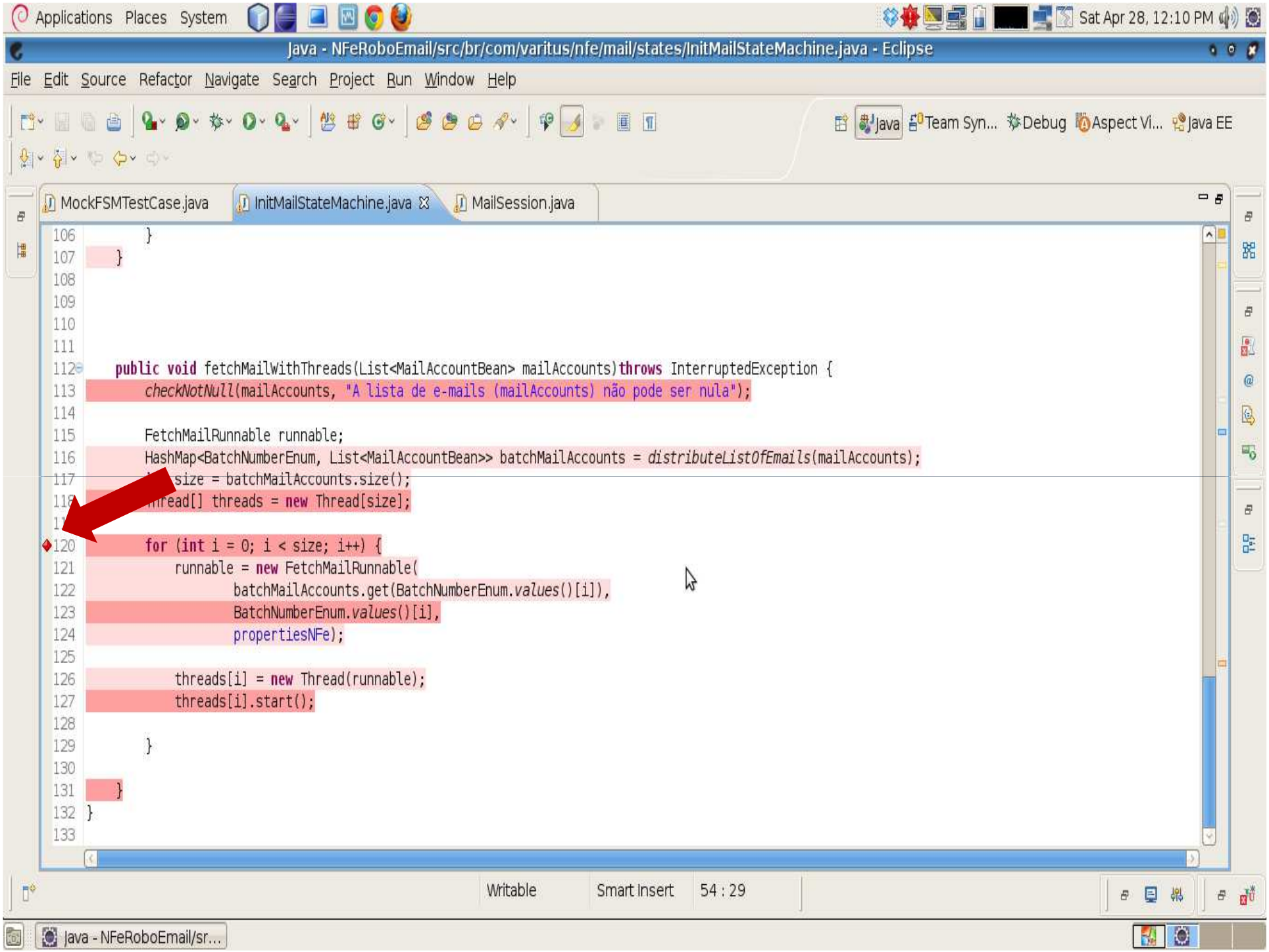

Applications Places System
Sat Apr 28, 10:38 AM
Java - NFeRoboEmail/test/junit/app/cml/CommandLineParametersTestCase.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Team Syn...
Debug
Aspect Vi...
Java EE
RobotMainApp.java
PropertiesBuilder.ja
PropertiesNFeBean.ja
PropertiesNFeBuilder
SingleMode.java
CommandLineParameter
@Test
public void testMessage() throws JSAPException{
try {
String[] ARGS = { "" };
app = new RobotMainApp(ARGS);
app
.configureAndParseArguments()
.createArgumentBean()
.showMessages();
} catch (ExitException e) {
assertEquals("Exist status",1,e.status);
}
JUnit
Finished after 0.545 seconds
Runs: 9/9
Errors: 0
Failures: 0
junit.app.cml.CommandLineParametersTestCase
testMessage (0.390 s)
testMessageHelp (0.000 s)
testPathParameters (0.000 s)
testVersionParameters (0.000 s)
Failure Trace
Writable
Smart Insert
21 : 38
Java - NFeRoboEmail/te...

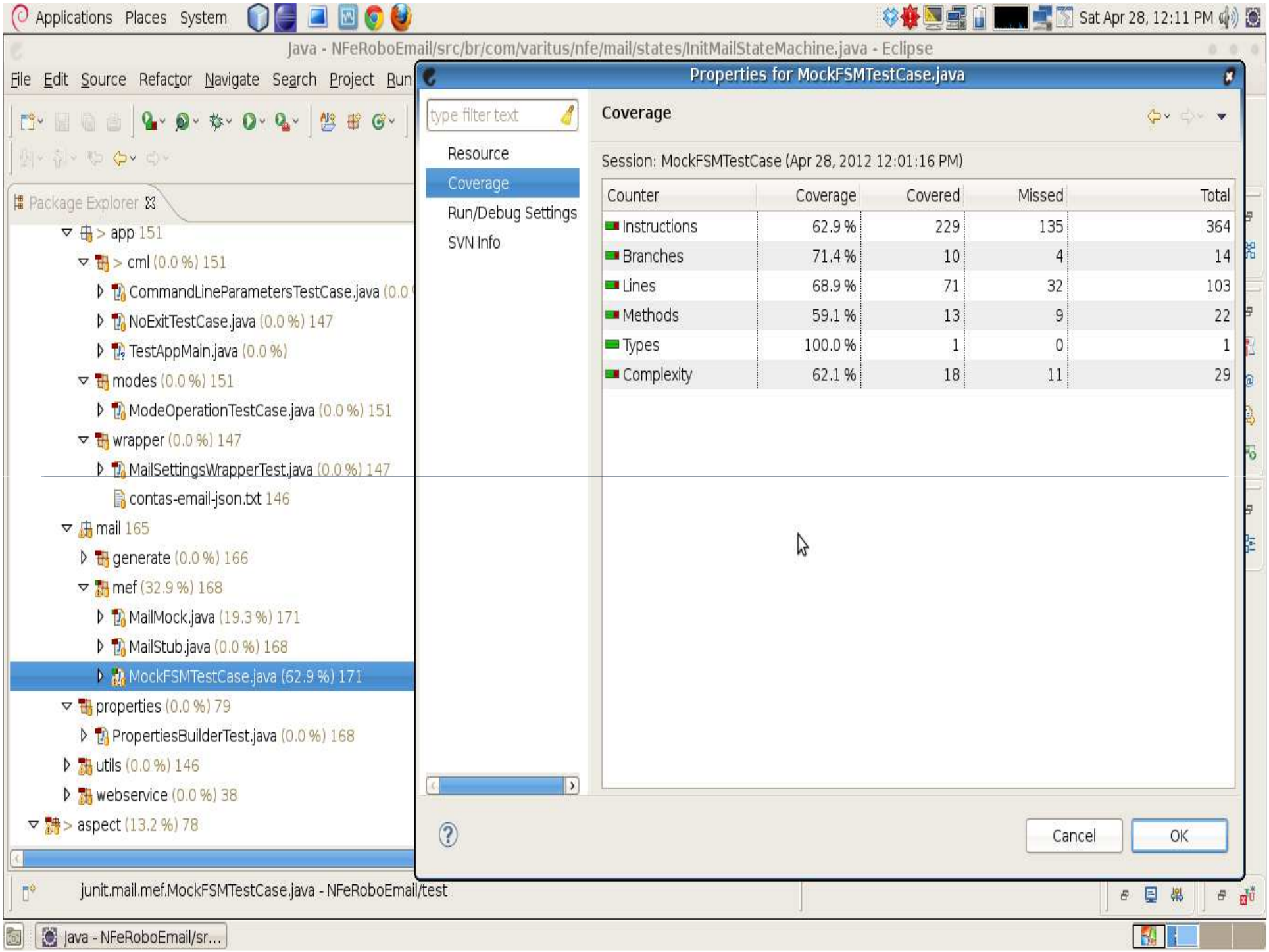
```
class FuncionarioDAOTest{
    private FuncionarioDAO dao;

    @Mock
    private Transacao transacao;

    @Before
    public void init(){
        MockitoAnnotations.initMocks(this);
        dao = new FuncionarioDAO(transacao);
    }

    @Test
    public void testQueChecaUsuario(){
        when(transacao.executar("1234"))
            .thenReturn("Eva    1234 1111");
        Funcionario umFuncionario = dao.buscarFuncionario("1234");

        assertEquals("Eva" ,  umFuncionario.getNome() );
        assertEquals(1234  ,  umFuncionario.getSetor());
        assertEquals("1111",  umFuncionario.getDepto());

        verify(transacao, atMostOnce()).executar("1234");

    }
}
```


Applications Places System
Sat Apr 28, 12:02 PM
Java - NFeRoboEmail/test/junit/mail/mef/MockFSMTestCase.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Team Syn...
Debug
Aspect Vi...
Java EE
Package Explorer
> NFeRoboEmail (25.3 %) 165 [https://192.168.10.11/svn/NFeRoboEmail, Trunk:
> src (28.8 %) 165
> test (19.4 %) 165
cases 38
> junit 165
> app 151
> cml (0.0 %) 151
CommandLineParametersTestCase.java (0.0 %) 151
NoExitTestCase.java (0.0 %) 147
TestAppMain.java (0.0 %)
modes (0.0 %) 151
ModeOperationTestCase.java (0.0 %) 151
wrapper (0.0 %) 147
MailSettingsWrapperTest.java (0.0 %) 147
contas-email-json.txt 146
mail 165
generate (0.0 %) 166
mef (32.9 %) 168
MailMock.java (19.3 %) 171
MailStub.java (0.0 %) 168
MockFSMTestCase.java (62.9 %) 171
MockFSMTestCase.java
.addGmailNFE01Varitus()
.toJson()
.fetchAllSettingsMock();
properties = configurePropertiesMock();
}
@Test
public void initializeEventTest() throws Exception{
for (MailAccountBean mailAccount : mails) {
StateMachineContext context = new StateMachineContext();
context.setMailAccount(mailAccount);
context.setPropertiesNFe(properties);
statemachine
.initContext(context)
;
}
assertAuthenticatorState();
}
@Test
public void authenticatorEventTest() throws Exception {
for (MailAccountBean mail : mails) {
junit.mail.mef.MockFSMTestCase.java - NFeRoboEmail/test
Java - NFeRoboEmail/te...


Applications Places System
Sat Apr 28, 12:03 PM
Java - NFeRoboEmail/test/junit/mail/mef/MockFSMTestCase.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Team Syn...
Debug
Aspect Vi...
Java EE
MockFSMTestCase.java
public void authenticatorEventTest() throws Exception {
for (MailAccountBean mail : mails) {
StateMachineContext context = new StateMachineContext();
context.setMailAccount(mail);
context.setPropertiesNFe(properties);
statemachine
.initContext(context)
.configureAuthenticator()
;
}
assertMailSessionState();
}
@Test
public void SessionEventTest() throws Exception {
for (MailAccountBean mail : mails) {
StateMachineContext context = new StateMachineContext();
context.setMailAccount(mail);
context.setPropertiesNFe(properties);
statemachine
.initContext(context)
.configureAuthenticator()
.configureMailSession()
Writable
Smart Insert
30 : 5
Java - NFeRoboEmail/te...


Applications Places System
Sat Apr 28, 12:08 PM
Java - NFeRoboEmail/src/br/com/varitus/nfe/mail/states/MailSession.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Team Syn...
Debug
Aspect Vi...
Java EE
MockFSMTestCase.java
InitMailStateMachine.java
MailSession.java
53
54 public Session getSession() {
55 Properties properties = getProperties();
56 javax.mail.Authenticator authenticator = getAuthenticator();
57 return Session.getDefaultInstance(properties, authenticator);
58
59 }
60
61 operties getProperties() {
62 Properties properties = new Properties();
63 MailAccountBean account = getMailAccount();
64 if (account.isSSL()) {
65 properties.put("mail.imap.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
66 }
67
68 properties.put("mail.imap.socketFactory.fallback", "false");
69 properties.put("mail.imap.port", account.getPort());
70 properties.put("mail.imap.socketFactory.port", account.getPort());
71 properties.put("mail.imap.host", account.getHost());
72
73 return properties;
74 }
75
76
77 public javax.mail.Authenticator getAuthenticator() {
78 return previousState().createJavaxMailAuthenticator();
79 }
80 }
Writable
Smart Insert
48 : 38
Java - NFeRoboEmail/sr...


Applications Places System
Sat Apr 28, 12:10 PM
Java - NFeRoboEmail/src/br/com/varitus/nfe/mail/states/InitMailStateMachine.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Team Syn...
Debug
Aspect Vi...
Java EE
MockFSMTestCase.java
InitMailStateMachine.java
MailSession.java
106 }
107 }
112 public void fetchMailWithThreads(List<MailAccountBean> mailAccounts)throws InterruptedException {
113 checkNotNull(mailAccounts, "A lista de e-mails (mailAccounts) não pode ser nula");
115 FetchMailRunnable runnable;
116 HashMap<BatchNumberEnum, List<MailAccountBean>> batchMailAccounts = distributeListOfEmails(mailAccounts);
117 size = batchMailAccounts.size();
118 hread[] threads = new Thread[size];
120 for (int i = 0; i < size; i++) {
121 runnable = new FetchMailRunnable(
122 batchMailAccounts.get(BatchNumberEnum.values()[i]),
123 BatchNumberEnum.values()[i],
124 propertiesNFe);
126 threads[i] = new Thread(runnable);
127 threads[i].start();
129 }
131 }
132 }
Writable
Smart Insert
54 : 29
Java - NFeRoboEmail/sr...


Applications Places System
Sat Apr 28, 12:11 PM
Java - NFeRoboEmail/src/br/com/varitus/nfe/mail/states/InitMailStateMachine.java - Eclipse
File Edit Source Refactor Navigate Search Project Run
Package Explorer
> app 151
> cml (0.0 %) 151
CommandLineParametersTestCase.java (0.0
NoExitTestCase.java (0.0 %) 147
TestAppMain.java (0.0 %)
modes (0.0 %) 151
ModeOperationTestCase.java (0.0 %) 151
wrapper (0.0 %) 147
MailSettingsWrapperTest.java (0.0 %) 147
contas-email-json.txt 146
mail 165
generate (0.0 %) 166
mef (32.9 %) 168
MailMock.java (19.3 %) 171
MailStub.java (0.0 %) 168
MockFSMTestCase.java (62.9 %) 171
properties (0.0 %) 79
PropertiesBuilderTest.java (0.0 %) 168
utils (0.0 %) 146
webservice (0.0 %) 38
> aspect (13.2 %) 78
junit.mail.mef.MockFSMTestCase.java - NFeRoboEmail/test
Java - NFeRoboEmail/sr...
Properties for MockFSMTestCase.java
type filter text
Resource
Coverage
Run/Debug Settings
SVN Info
Coverage
Session: MockFSMTestCase (Apr 28, 2012 12:01:16 PM)
Counter Coverage Covered Missed Total
Instructions 62.9 % 229 135 364
Branches 71.4 % 10 4 14
Lines 68.9 % 71 32 103
Methods 59.1 % 13 9 22
Types 100.0 % 1 0 1
Complexity 62.1 % 18 11 29
Cancel
OK

Avaliação da Cobertura de Código Usando Programação Orientada a Aspecto como Apoio da Testabilidade em *Poly-Paradigm Programming*

Matheus Vinícius Binotto[1], Rafael Leandro Araujo[1], Erik A. Antonio[2]
[1]Sistemas de Informação, FHO, Araras, SP.
[2]Depto. de Computação, LaPES, UFSCar, São Carlos, SP.

1. Objetivos

O desenvolvimento de software tem se tornado uma atividade cada vez mais complexa e isso tem impactado na forma como os sistemas são projetados, compreendidos e testados [1]. O uso de Orientação a Aspecto (OA) combinado com outros paradigmas de desenvolvimento fornece mecanismos para se tratar problemas relacionados com a modularidade, restrição arquitetural e testabilidade. Pode-se notar também que, contribuições como Programação *Poliglota* ou *Poly-Paradigm Programming* (PPP) [2] tem recebido atenção da Comunidade de Engenharia de Software, em especial no que se refere ao uso combinado de diferentes paradigmas. Este artigo visa descrever uma avaliação do uso de OA no apoio ao tratamento de interesses transversais relacionados à testabilidade de código, no contexto de *Poly-Paradigm Programming*.

verificar que a cobertura de código em termos das métricas de complexidade ciclomática e linhas de código sugerem que a segunda versão desenvolvida com OA, auxilia na melhora da qualidade do projeto de casos de testes elaborados com OO e TDD. A Figura 1 ilustra os resultados obtidos.

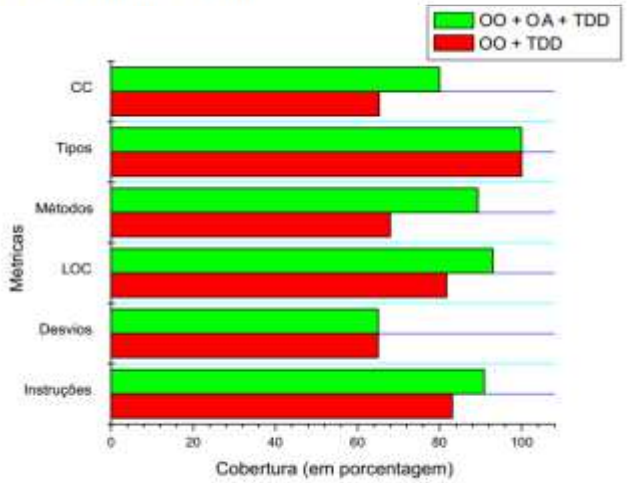
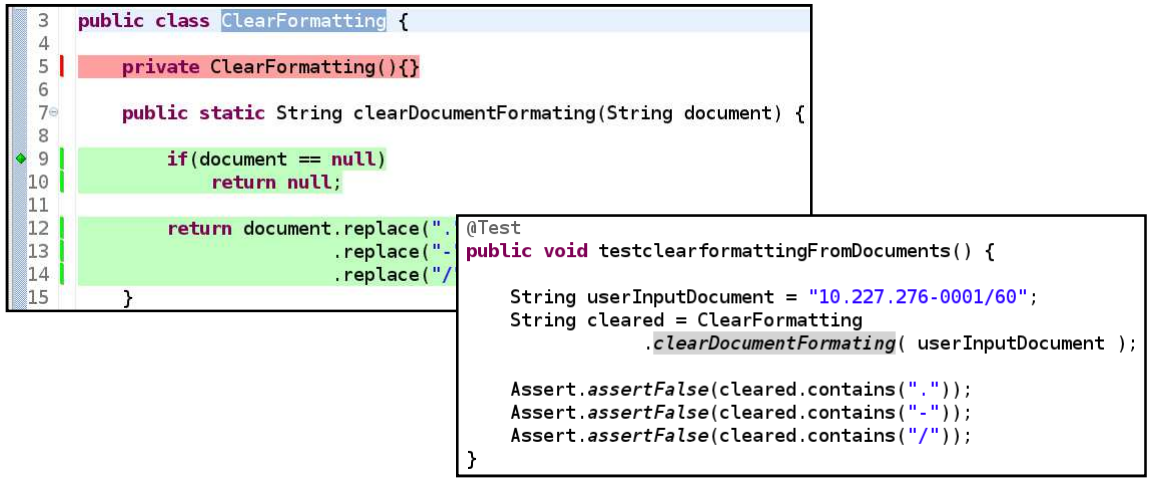


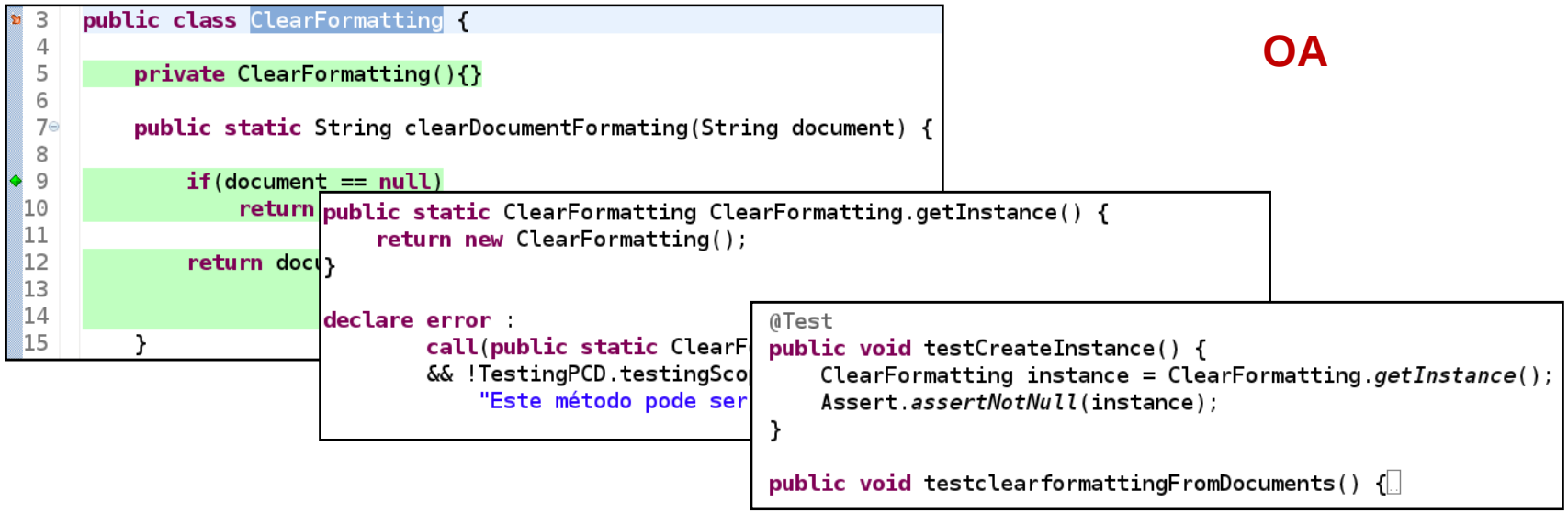
Figura 1 - Sumarização da cobertura da aplicação



```
3  public class ClearFormatting {
4
5      private ClearFormatting(){}
6
7      public static String clearDocumentFormating(String document) {
8
9          if(document == null)
10             return null;
11
12         return document.replace(".
13                         .replace("-
14                         .replace("/
15     }
```

```
@Test
public void testclearformattingFromDocuments() {

    String userInputDocument = "10.227.276-0001/60";
    String cleared = ClearFormatting
                .clearDocumentFormating( userInputDocument );

    Assert.assertFalse(cleared.contains("."));
    Assert.assertFalse(cleared.contains("-"));
    Assert.assertFalse(cleared.contains("/"));
}
```

OO

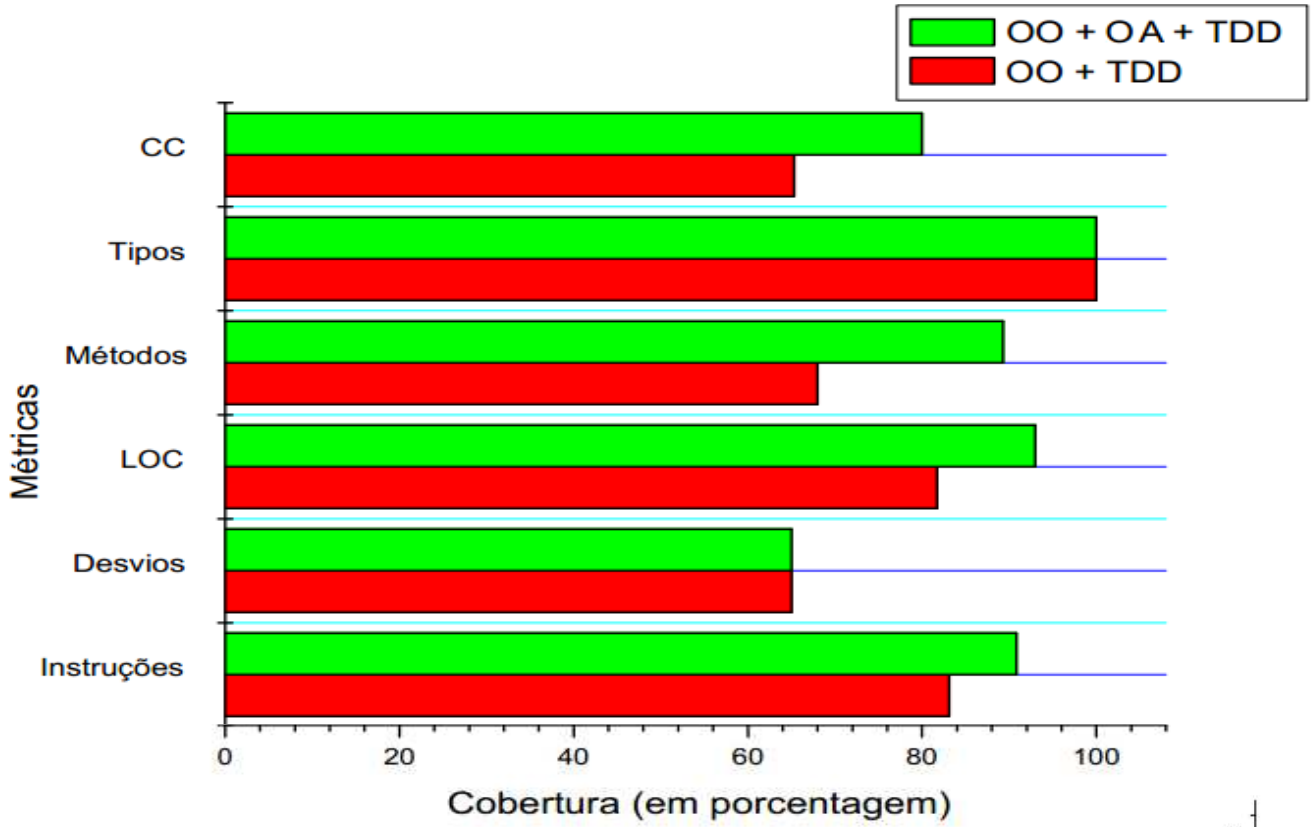


```
3  public class ClearFormatting {
4
5      private ClearFormatting(){}
6
7      public static String clearDocumentFormating(String document) {
8
9          if(document == null)
10             return
11
12         return doc
13
14
15     }
```

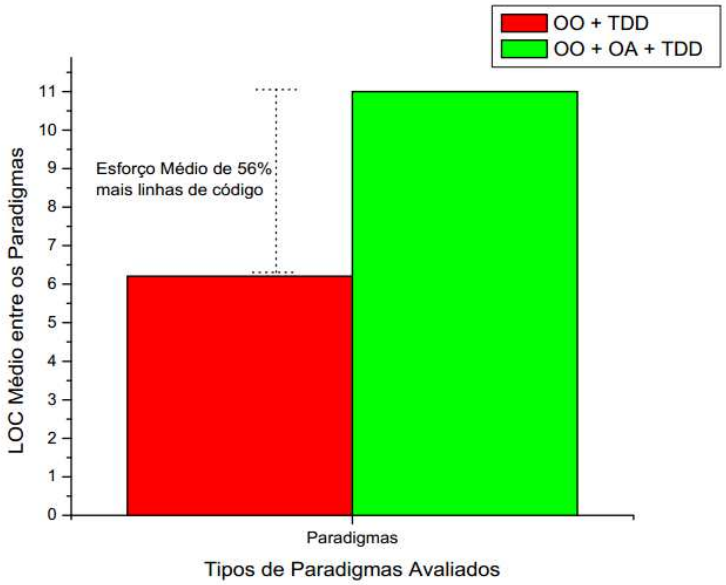
```
public static ClearFormatting ClearFormatting.getInstance() {
    return new ClearFormatting();
}

declare error :
        call(public static ClearF
        && !TestingPCD.testingSco
            "Este método pode ser
```

```
@Test
public void testCreateInstance() {
    ClearFormatting instance = ClearFormatting.getInstance();
    Assert.assertNotNull(instance);
}

public void testclearformattingFromDocuments() {
```

OA



$$Esforço = \frac{LOC_{OO+TDD}}{LOC_{OO+OA+TDD}} \qquad (1)$$

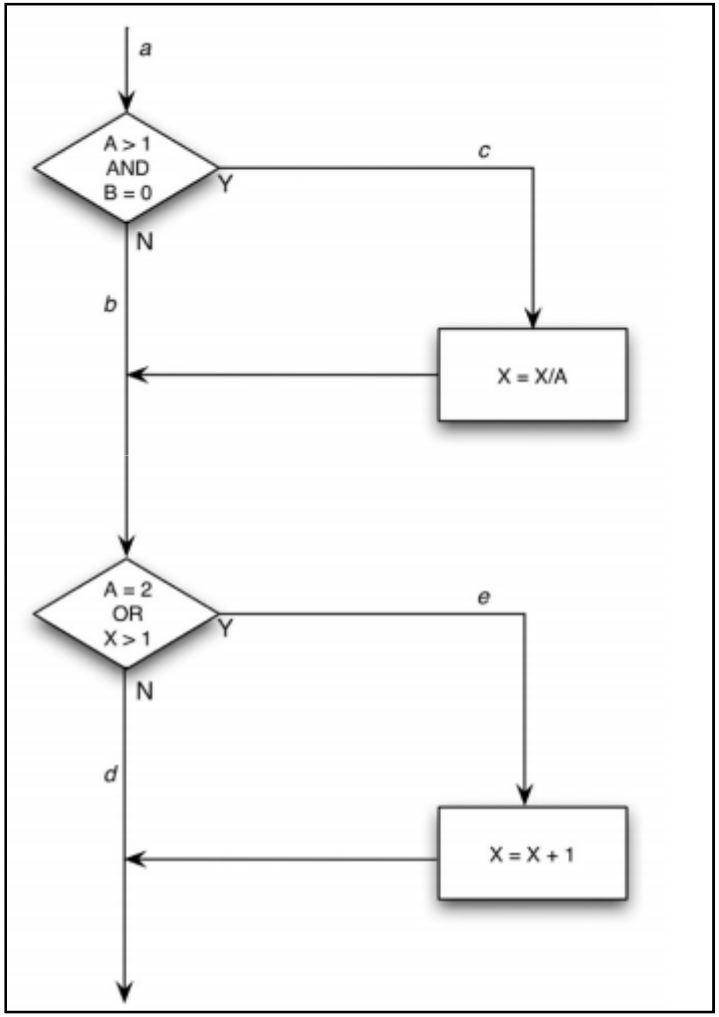


Prática

Teste Caixa Branca – White Box Testing

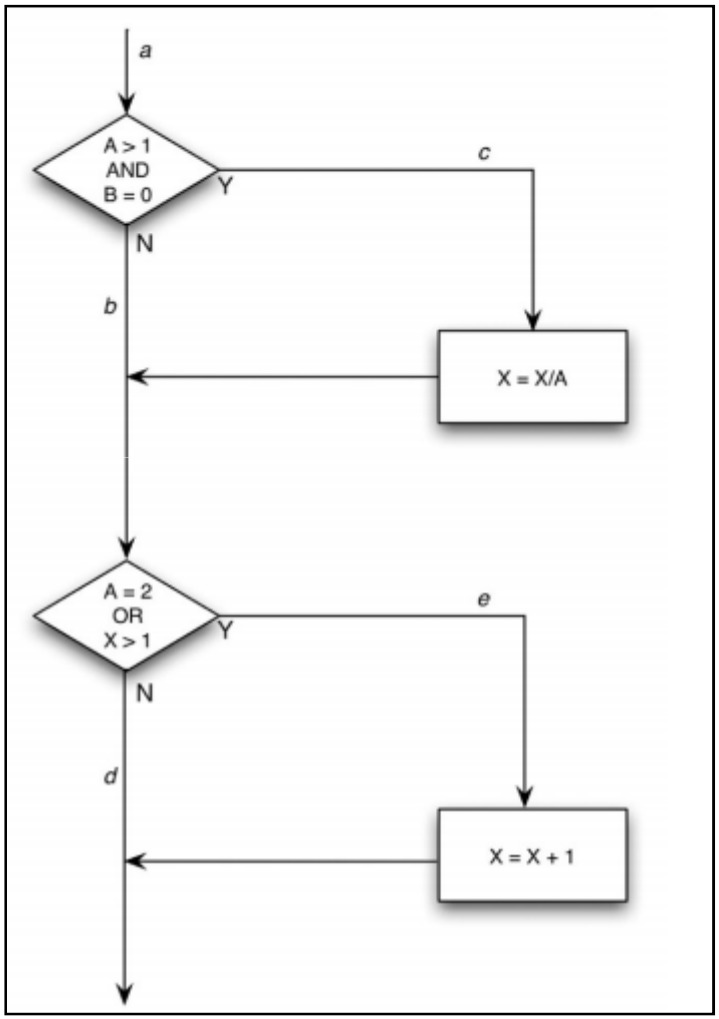
```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```

- Criar a classe Java
- Criar o Teste de Unidade
- Criar Casos de Testes com critério
  - Statement Coverage
  - Decision/Branch Coverage
  - Condition Coverage



Teste Caixa Preta– Black Box Testing

```
public void foo(int a, int b, int x) {
    if (a>1 && b==0) {
        x=x/a;

    }
    if (a==2 || x>1) {
        x=x+1;
    }


}
```



- Criar casos de teste com classes de equivalência
- Criar casos de teste com análise do valor limite
- Criar casos de teste com Erros de Advinhação típicos

aceiro@gmail.com

erik_aceiro@hotmail.com

http://erikblogger.blogspot.com

facebook.com/erik.aceiro

@eaceiro