




Laboratório de Engenharia de Software

Teste Funcional 3

Arndt von Staa
Departamento de Informática
PUC-Rio
Março 2015

Especificação




Laboratório de Engenharia de Software

- Objetivo desse módulo
 - Apresentar uma modalidade de geração de casos de teste a partir de casos de uso
- Justificativa
 - Casos de uso são utilizados para especificar sistemas
 - Especificações devem ser verificáveis
 - É desejável que, além de serem verificáveis, seja possível gerar os casos de teste diretamente a partir das especificações, mesmo que utilizando técnicas semi-automatizadas

Mar 2015Arndt von Staa © LES/DI/PUC-Rio2

Laboratório de Engenharia de Software

Motivação



- Quanto mais cedo forem gerados os casos de teste, melhor
- Quanto mais automaticamente forem executados e, se possível, gerados, melhor ainda
 - Desenvolvimento Dirigido por Teste de Aceitação
ATDD Acceptance Test Driven Development


Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

3

Laboratório de Engenharia de Software

Teste dirigido por comportamento



- O teste dirigido por comportamento (*behaviour driven testing*) procura gerar semi-automaticamente casos de teste a partir de cenários de uso
 - uma forma alternativa é: gerar casos de teste diretamente a partir de casos de uso

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

4

Teste dirigido por comportamento



- Historieta (user story)
 - Como Administrador do sistema
 - Eu quero ser capaz de gerenciar os dados dos usuários
 - Para poder manter o sistema atualizado
- Um dos cenários
 - Dado que estou autenticado como Administrador
 - Quando eu clicar no botão Listar usuários
 - Então será exibida a lista de todos os usuários cadastrados
 - Quando eu selecionar nesta lista o usuário José
 - E clicar no botão Apagar
 - Então será reexibida a lista dos usuários cadastrados
 - E esta lista não conterá o usuário José

Teste dirigido por comportamento




- Visão resumida
 - Scenario: Admin successfully creates user
 - Given I am logged in with user "admin" and password "admin"
 - When I choose to create a new user and
 - When I enter name "Maria" and
 - When I enter email "maria@gmail.com" and
 - When I enter login "maria" and
 - When I enter password "maria123" and
 - When I click save button
 - Then I should see the user "Maria" in the list of users

- Basta ver o usuário Maria? Ou seria necessário ver também os demais dados?

Laboratório de Engenharia de Software

Teste dirigido por comportamento



```

@When( "I click save button" )
public void clickSaveButton() throws InterruptedException
{
    this.selenium.click( "bar" );
    Thread.sleep( 5000 );
}

@When( "I enter email \"$email\" and" )
public void SetUserEmail( final String email )
{
    this.selenium.type( "email", email );
}

@When( "I enter login \"$login\" and" )
public void setUserLogin( final String login )
{
    this.selenium.type( "chave", login );
}

@When( "I enter name \"$name\" and" )
public void SetUserName( final String name )
{
    this.selenium.type( "nome", name );
}

```


Código gerado para Selenium

incompleto

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
7

Laboratório de Engenharia de Software

Exemplo: Caso de uso Efetuar Login



Fluxo principal	<ol style="list-style-type: none"> 1. O componente limpa os campos e gera os caracteres de controle 2. O usuário digita sua identificação, senha e os caracteres de controle 3. O usuário clica "Login" 4. O controle de acesso verifica se os dados estão lexicamente corretos 5. O controle de acesso verifica se <usuário, senha> está autorizado 6. O controle de acesso retorna ao sistema Sis, fornecendo a condição "autorizar uso" e os direitos de uso correspondentes a <usuário, senha>
Fluxos alternativos	<p>Evento E1/4: Um ou mais campos estão lexicamente incorretos</p> <p style="margin-left: 20px;">E1.1 Até 3 tentativas efetue</p> <p style="margin-left: 40px;">E1.1.1 O controle de acesso informa ao usuário a ocorrência de um erro de uso</p> <p style="margin-left: 40px;">E1.1.2 repete a partir de 1.</p> <p style="margin-left: 20px;">E1.2 Após a terceira tentativa efetue</p> <p style="margin-left: 40px;">E1.2.1 O controle de acesso informa ao usuário que ele não está autorizado a usar o sistema Sis</p> <p style="margin-left: 40px;">E1.2.2 O controle de acesso fornece o conjunto "não autorizado" ao sistema Sis</p> <p style="margin-left: 20px;">Fim repetição</p> <p style="margin-left: 20px;">Fim evento E1.</p>

O caso de uso completo está na [Aula 04 Especificações, resumo](#)

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
8


Exemplo: Caso de uso Efetuar Login		LES
Laboratório de Engenharia de Software	Fluxos alternativos	<p>Evento E2/5: o usuário digita um par <usuário, senha> que não é confirmado</p> <p>E2.1 Até 3 tentativas efetue</p> <p>E2.1.1 O controle de acesso informa ao usuário a ocorrência de um erro de uso</p> <p>E2.1.2 Repete a partir de 1.</p> <p>E2.2 Após a terceira tentativa efetue</p> <p>E2.2.1 O controle de acesso informa ao usuário que ele não está autorizado a usar o sistema sis</p> <p>E2.2.2 O controle de acesso fornece o conjunto "não autorizado" ao sistema sis</p> <p>Fim repetição</p> <p>Fim evento E2.</p>
		<p>Evento E3/3: o usuário clica "Trocar senha"</p> <p>E3.1 O controle de acesso verifica se os dados estão lexicamente corretos</p> <p>E3.2 O controle de acesso verifica se <usuário, senha> está autorizado</p> <p>E3.3 O controle de acesso ativa o diálogo "Trocar senha"</p> <p>E3.4 Repete a partir de 1</p> <p>Fim evento E3.</p>
Mar 2015		Arndt von Staa © LES/DI/PUC-Rio 9

Exemplo: Caso de uso Efetuar Login		LES
Laboratório de Engenharia de Software	Fluxos alternativos	<p>Evento E4/3: o usuário clica "Esqueci senha"</p> <p>E4.1 O controle de acesso verifica se os dados estão lexicamente corretos</p> <p>E4.2 O controle de acesso verifica se usuário está cadastrado</p> <p>E4.3 O controle de acesso ativa o diálogo "Esqueci senha"</p> <p>E4.4 Se retornar do diálogo</p> <p>E4.4.1 Repete a partir de 1</p> <p>Fim se</p> <p>Fim evento E4.</p>
		<p>Evento E5: o usuário clica "Cancelar" em qualquer lugar</p> <p>E5.1 O sistema solicita confirmação do cancelamento</p> <p>E5.2 Se usuário confirma o cancelamento</p> <p>E5.2.1 O controle de acesso retorna fornecendo o conjunto "cancelar uso" ao sistema sis</p> <p>Fim se</p> <p>E5.3 Repete a partir de 1</p> <p>Fim evento E5.</p>
Mar 2015		Arndt von Staa © LES/DI/PUC-Rio 10

Laboratório de Engenharia de Software

Geração de casos de teste a partir de casos de uso

1. Identificar as **pré-condições necessárias** para poder realizar o teste
2. Criar **cenários de uso normal**
 - segue-se o fluxo principal
 - cada caminho narra a sequência de ações do usuário e os resultados por ele observáveis
 - são usos normais fluxos alternativos que retornam ao fluxo principal
 - pode ser necessário instrumentar o programa de modo que os resultados possam ser efetivamente observados
3. Criar **cenários de uso não normal**
 - segue-se o fluxo principal e aplica-se a ele os eventos dos fluxos alternativos
 - cada caminho que não retorna ao fluxo principal é um cenário não normal e narra a sequência de ações do usuário e os resultados por ele observáveis
4. Criar uma **tabela de decisão**, ou um **roteiro de teste**
 - a tabela de decisão terá pelo menos tantas colunas quantos são os cenários



Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
11


Laboratório de Engenharia de Software

Pré-condição + invariante

- Para poder testar, o sistema **TesteSis** precisa estar vinculado a uma base de dados contendo os dados de usuários cadastrados e a serem utilizados durante os testes, ex:

Por que e o que é **TesteSis** ?

 - O sistema **SisTeste** deve estar vinculado à base de dados de usuários **TesteSis.usuarios**
 - A base de dados **TesteSis.usuarios** deve estar inicializada e criptografada com a senha de teste **XPTO###**
 - O ideal seria a base de dados ser gerada a cada vez que forem realizados os testes
 - Assegura que o teste não falhe em virtude de mudanças nos dados
 - Assegura que o teste não corrompa a base de dados
 - Pode ser realizado com DBUnit, ou um módulo especificamente projetado para esse fim
 - porém aumenta muito o tempo de execução do teste



Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
12


- Mar 2015

LES

- Laboratório de Engenharia de Software

Mar 2015

Cenário normal do login (caso semântico)




Laboratório de Engenharia de Software

- Os campos de entrada devem estar em branco
- Os caracteres de controle devem estar gerados
- O usuário digita *idUsuário* e *senha* de usuário autorizado
- O usuário digita os *caracteres de controle* válidos
- O usuário clica *Login*
- O componente retorna ao *driver* de teste: { *usuário autorizado* , *direitos de uso* } do usuário selecionado

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
15

Caso de teste do fluxo principal

formulário




Laboratório de Engenharia de Software

IDENTIFICADOR DO CASO DE TESTE: Login001 <div style="text-align: right; font-size: 0.8em;">Explicitando as ações do testador</div>		
DESCRIÇÃO: O usuário realiza um Login bem sucedido		
PRE-CONDICOES: <ul style="list-style-type: none"> O cadastro de usuários está criptografado com a senha de teste XPTO### Sistema TesteSis (i.e. o driver de teste) ativa controle de acesso com o cadastro TesteSis.usuarios Controle de acesso abriu a janela de login 	AÇÕES DO USUÁRIO: <ul style="list-style-type: none"> Verificar se identificação está vazia Verificar se senha está vazia Digitar a identificação: joaoSilva Digitar a senha: joao#### Verificar se campo senha contém somente '*' Digitar exatamente os caracteres de controle Clicar "Login" 	POS-CONDICOES: <ul style="list-style-type: none"> Controle de acesso fechou a janela de login Verificar se o sistema TesteSis recebeu o resultado: {autorizado , {a,b,c}} Usando TesteSis verificar se todos os requisitos de término foram satisfeitos.
CRITÉRIO DE SUCESSO: (a serem verificados durante a execução)		

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
16

Cenário de captcha incorreto




Laboratório de Engenharia de Software

- Os campos de entrada estão em branco
- Os caracteres de controle estão gerados
- O usuário digita *idUsuário* e *senha* de usuário autorizado
- O usuário digita *caracteres do captcha* não válidos
- O usuário clica Login
- O sistema observa erro léxico de *captcha*
- Se for primeiro a terceiro erro inclusive
 - O sistema emite a mensagem de erro “Erro de digitação”
 - Controle de acesso reinicia a aquisição de dados
- Senha:
 - Controle de acesso emite a mensagem de erro “Usuário não autorizado, processamento cancelado.”
 - Retorna {*usuário não autorizado , direitos de uso vazio*}

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
17

Outros cenários



Laboratório de Engenharia de Software

- Pode-se continuar criando cenários de teste de forma similar ao que foi feito até agora
- Problema: como saber se geramos todos os cenários?
- Solução: criar uma tabela de decisão que descreva as condições de todos os cenários possíveis

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
18

Tabela de decisão (incompleta)

Id cenário:	1	2	3	4
Id usuário	vale	vale	vale	não vale
Senha	vale	vale	não vale	qualquer
Captcha	vale	vale	vale	vale
Botão	Login	Cancela	Login	Login
Pós-condição	autorizado	cancelado	Erro: dados Volta início	Erro: dados Volta início

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

19

Laboratório de Engenharia de Software

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

19

Roteiro de teste da coluna 1	
Laboratório de Engenharia de Software	<ol style="list-style-type: none"> 1. Assegure que o driver de teste esteja vinculado ao cadastro TesteSis.usuarios, criptografado com senha XPT0### 2. Usando o <i>driver</i> de teste, ativar o Controle de Acesso 3. Após a janela abrir <ul style="list-style-type: none"> - verificar se os campos estão vazios - digitar idUsuário = joao.silva - digitar a senha correta = joao#### - verificar se o campo senha não exibe os caracteres digitados - digitar corretamente os caracteres de controle - clicar o botão <i>Login</i> - verificar se a janela fechou 4. Usando o <i>driver</i> de teste, verificar se os dados retornados são {Autorizado, { a,b,c }} 5. Usando o <i>driver</i> de teste, verificar todas as condições de término <p>é necessário especificar como o conjunto retornado é codificado.</p>

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

20




Tabela de decisão (incompleta)

Id cenário:	1	2	3	4
Id usuário	vale	vale	vale	não vale
Senha	vale	vale	não vale	qualquer
Captcha	vale	vale	vale	vale
Botão	Login	Cancela	Login	Login
Pós-condição	autorizado	cancelado	Erro: dados Volta início	Erro: dados Volta início

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

21

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

21

Roteiro de teste da coluna 2	
Laboratório de Engenharia de Software	<ol style="list-style-type: none"> 1. Assegure que esteja em uso o cadastro TesteSis.usuarios, criptografado com senha XPTO### 2. Usando o <i>driver</i> de teste, ativar o Controle de Acesso 3. Após a janela abrir o usuário deve <ul style="list-style-type: none"> - verificar se os campos estão vazios - digitar idUsuário = joao.silva - digitar a senha correta = joao#### - digitar corretamente os caracteres de controle - clicar o botão Cancelar - verificar se a janela fechou 4. Usando o <i>driver</i> de teste, verificar se os dados retornados são { cancelado , nulo } 5. Usando o <i>driver</i> de teste, verificar todas as condições de término


Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

22

Laboratório de Engenharia de Software

Cenário de teste do evento dado errado



1. O usuário digita a identificação, a senha e os caracteres do captcha, sendo que a e somente a senha está incorreta
2. O usuário clica *Login*
3. O controle de acesso verifica se os dados estão corretos
4. O controle de acesso verifica se o par <usuário, senha> está autorizado


- Evento E1: o par {usuário, senha} não é encontrado
 - E1.1 Se o número de tentativas for menor ou igual a 3 então
 - E1.1.1 O controle de acesso informa "usuário não conhecido"
 - E1.1.2 repetir a partir de 1
 - E1.2 Senão (Se o número de tentativas for maior do que 3 então)
 - E1.2.1 O controle de acesso informa que o usuário não está autorizado a usar o sistema TesteSis
 - E1.2.2 O controle de acesso fornece o controle "não autorizado" ao sistema TesteSis
 - Fim se
- Fim evento E1.

Devido à repetição este cenário se desdobra em:
 {1º. errado, 2º. correto} ; {1º. 2º e 3º. errados, quarto correto} ; {1º. 2º. 3º. e 4º. errados }

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
23

Laboratório de Engenharia de Software

Roteiro de teste do evento dado errado



1. Assegure que esteja em uso o cadastro
TesteSis.usuarios , criptografado com senha **XPTO###**
2. Usando o *driver* de teste, ativar o Controle de Acesso
3. Após a janela abrir o usuário deve
 - a) digitar idUsuário = **joao.silva**
 - b) digitar a senha incorreta = **joao###**
 - c) digitar corretamente os caracteres do captcha
 - d) clicar o botão *Login*
 - e) observar que recebeu a mensagem "Usuário/senha não conhecido"
 - f) clicar o botão **OK** da mensagem
 - g) verificar se retornou à janela de dados, com os campos usuário e senha apagados e caracteres de controle diferentes da vez anterior
 - h) digitar idUsuário = **joao.silva**
 - i) digitar a senha correta = **joao####**
 - j) digitar corretamente os caracteres de controle
 - k) clicar o botão *Login*
 - l) verificar se a janela fechou
4. Usando o *driver* de teste, verificar se os dados retornados são
{ autorizado , { a,b,c } }
5. Usando o *driver* de teste, verificar todas as condições de término

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
24

Laboratório de Engenharia de Software

LES

O resto fica para exercício 😊

Mar 2015Arndt von Staa © LES/DI/PUC-Rio25

Laboratório de Engenharia de Software

LES

Como automatizar a geração de cenários?

- No módulo (classe) de teste específico
 - cada ação do usuário mapeia sobre uma função (método)
 - cada verificação mapeia sobre uma função de verificação
 - uma espécie de assertiva
 - prefiro a terminologia: “verificar” ao invés de “assert”
 - para cada caso de teste cria-se uma sequência de chamadas que implementa o caso de teste
- É necessário que o módulo de teste específico seja capaz de interagir com a interface de usuário do sistema

Mar 2015Arndt von Staa © LES/DI/PUC-Rio26

Como automatizar?



- Pode-se gerar um script com uma ferramenta de *capture and replay* (SQUISH)

1. O programa deve ser ativado.
2. O usuário deve selecionar o nome *Thiago* no combobox *ComboAluno*.
3. O usuário deve ativar o botão *CalcularMedia*.
4. O programa deve exibir o diálogo de calculo de média.
5. O usuário deve preencher o campo *P1* com o valor 7.3.
6. O usuário deve preencher o campo *P2* com o valor 6.7.
7. O usuário deve preencher o campo *P3* com o valor 4.1.
8. O usuário deve preencher o campo *P4* com o valor 8.5.
9. O usuário deve ativar o botão *Calcular*.
10. O programa deve exibir o valor 6.65 no campo *Media*.
11. O usuário deve ativar o botão *Aceitar*.
12. O programa deve fechar o diálogo.
13. O usuário deve ativar o botão *Fechar*.
14. O programa deve terminar.

Araújo, T.P.; Staa, A.v.; *Um Método Baseado em Comportamento com Foco no Desenvolvimento de Aplicações Baseadas em Interfaces Gráficas*;

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

27

Como automatizar?



Usando a ferramenta SQUISH é produzido o script na linguagem Python

```
1. def main():
2.     waitForObjectItem(":groupBox1.comboBoxName_QComboBox", "Thiago")
3.     clickItem(":groupBox1.comboBoxName_QComboBox", "Thiago", 51, 7, 1, Qt.LeftButton)
4.     waitForObject(":groupBox1.Calcular média_QPushButton")
5.     clickButton(":groupBox1.Calcular média_QPushButton")
6.     waitForObject(":Calcular Média.lineEditP1_QLineEdit")
7.     mouseClick(":Calcular Média.lineEditP1_QLineEdit", 49, 13, 1, Qt.LeftButton)
8.     waitForObject(":Calcular Média.lineEditP1_QLineEdit")
9.     type(":Calcular Média.lineEditP1_QLineEdit", "7.3")
10.    waitForObject(":Calcular Média.lineEditP2_QLineEdit")
11.    mouseClick(":Calcular Média.lineEditP2_QLineEdit", 21, 8, 1, Qt.LeftButton)
12.    waitForObject(":Calcular Média.lineEditP2_QLineEdit")
13.    type(":Calcular Média.lineEditP2_QLineEdit", "6.7")
14.    waitForObject(":Calcular Média.lineEditP3_QLineEdit")
15.    mouseClick(":Calcular Média.lineEditP3_QLineEdit", 37, 10, 1, Qt.LeftButton)
16.    waitForObject(":Calcular Média.lineEditP3_QLineEdit")
17.    type(":Calcular Média.lineEditP3_QLineEdit", "4.1")
18.    waitForObject(":Calcular Média.lineEditP4_QLineEdit")
19.    mouseClick(":Calcular Média.lineEditP4_QLineEdit", 42, 10, 1, Qt.LeftButton)
20.    waitForObject(":Calcular Média.lineEditP4_QLineEdit")
21.    type(":Calcular Média.lineEditP4_QLineEdit", "8.5")
22.    waitForObject(":Calcular Média.Calcular_QPushButton")
23.    clickButton(":Calcular Média.Calcular_QPushButton")
24.    waitForObject(":Calcular Média.Aceitar_QPushButton")
25.    clickButton(":Calcular Média.Aceitar_QPushButton")
26.    waitForObject(":GBDD Example.Fechar_QPushButton")
27.    clickButton(":GBDD Example.Fechar_QPushButton")
```


Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

28

Laboratório de Engenharia de Software

Uso de dados variáveis




- Ao invés de dados constantes pode-se utilizar variáveis
 - Assegurar que o Cadastro de usuários contenha
`{idUsuario1, senha1} → {conjunto1}`
 - Usando o *driver* de teste, ativar o componente Controle de Acesso solicitando autorização de uso
 - Após a janela abrir o usuário deve
 - fornecer idUsuário = `idUsuario1`
 - fornecer senha = `senha1`
 - digitar os caracteres de controle ← Como resolver esse?
 - selecionar o botão `idLogin` Varia a cada vez que se pode fornecer dados.
 - verificar se a janela fechou
 - No *driver* de teste verificar se o conjunto de direitos de uso retornado é `{Login, conjunto1}`

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
29

Laboratório de Engenharia de Software

Uso de dados variáveis a cada ativação




- Solução 1 – fixar os dados
 - durante o teste gerar sempre os mesmos caracteres de controle
 - precisa alterar o cenário
 - precisa testar a geração de caracteres de controle em separado
- Solução 2 – uso de função “**call back**” (instrumentação)
 - instrumenta-se o gerador de caracteres de controle para
 - compilação condicional ativa sse em modo teste (`_DEBUG`)
 - guardar os caracteres em alguma variável global
 - disponibilizar uma função que fornece os caracteres guardados
 - instrumenta-se o “digitador” de caracteres de controle para
 - buscar os caracteres guardados
 - verificar se os caracteres buscados são sempre diferentes dos que foram buscados no acesso anterior

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
30

Laboratório de Engenharia de Software

Uso de dados variáveis



- Vantagens do uso de dados variáveis
 - pode-se gerar automaticamente dados de teste
 - por exemplo a partir de uma tabela de decisão
 - a automação da valoração dos casos de teste reduz os custos dos testes
 - a geração de um grande número de dados aleatórios aumenta a chance de se criar “sequências de uso extensas”, ou pouco comuns, ou até mesmo incoerentes ao se considerar o domínio da aplicação
 - teste do macaco, ou do gato?
 - a geração de dados aleatórios usando sequências de dados diferentes aumenta a chance de exercitar “defeitos encapsulados”, i.e. defeitos que ainda não foram exercitados
 - a cada ativação do teste usar uma outra semente para o gerador de números aleatórios: usualmente o relógio

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
31

Laboratório de Engenharia de Software

Referências bibliográficas



- Araújo, T.P.; Staa, A.v.; Um Método Baseado em Comportamento com Foco no Desenvolvimento de Aplicações Baseadas em Interfaces Gráficas; Monografias em Ciência da Computação no. 26/09; DI/PUC-Rio; 2009
- Caldeira, L.R.N.; *Geração semi-automática de massas de testes funcionais a partir da composição de casos de uso e tabelas de decisão*; Dissertação de Mestrado, DI/PUC-Rio; 17/ago/2010
- Heumann, J.; “Generating Test Cases from Use Cases”; *The Rational Edge e-zine* www.ibm.com/developerworks/rational/rationaledge/ ; New York, NY: International Business Machines; 2001; Buscado em: 22/jan/2009; URL: www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf
- Lachtermacher, L.; : *Geração Automática de Dados de Teste através de Tabelas de Decisão*; Dissertação de Mestrado, DI/PUC-Rio; 5/mar/2010

Mar 2015
Arndt von Staa © LES/DI/PUC-Rio
32

Laboratório de Engenharia de Software

LES

FIM

Mar 2015

Arndt von Staa © LES/DI/PUC-Rio

33