

# Model-Based Testing

...

Teste, Verificação e Validação de Software

# Introduction

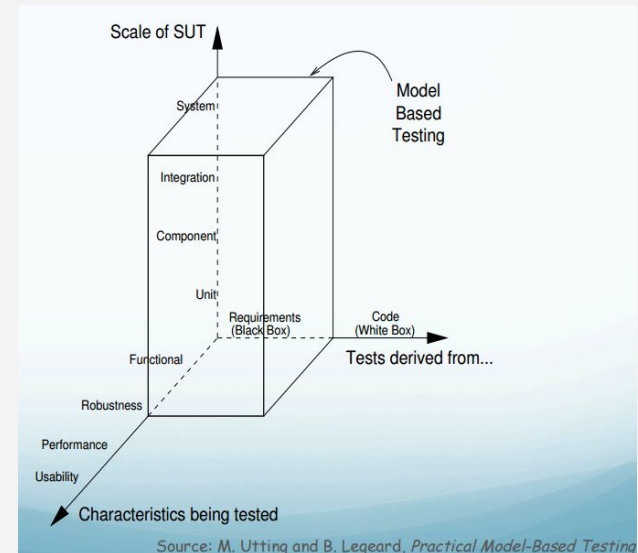
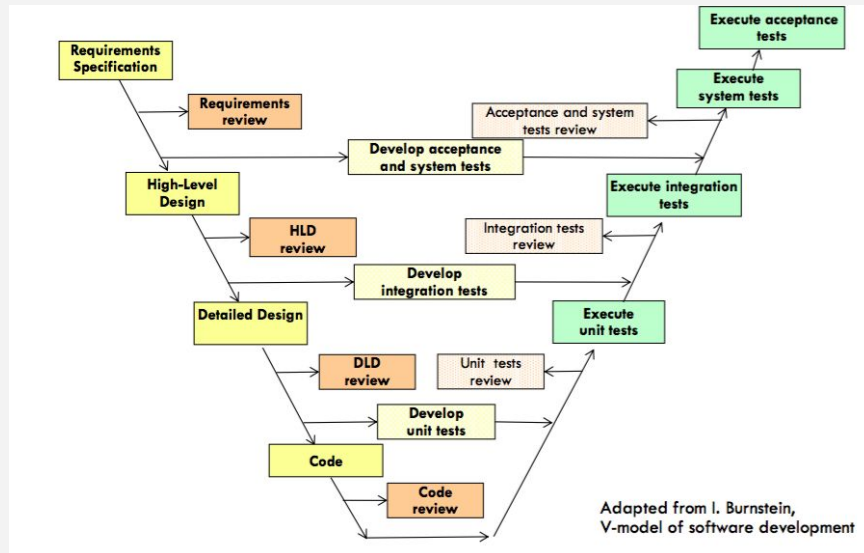
- What is Model-Based Testing
- Why use Model-based Testing
- Advantages vs Disadvantages
- Some tools available
- Selected tool: PBGT
  - UI Pattern Example
  - UI Patterns
  - UI Test Pattern Definition

# What is Model-Based Testing

Model-based testing is the automation of generating tests from system models. System models have a certain level of abstraction, that results in abstract test cases.



Domain Specific Language



# Why use Model-based testing

- Build the right program - fulfills the intended use in the intended environment.
- Model driven development
- Generate functional tests
- Black-box tests
- High number of tests cases (test case generation)

# Advantages vs Disadvantages

- Validates the program  
(design vs implementation)
  - Automated test generation  
(no coding)
  - Model reuse
  - Larger projects/ multiple  
development teams
  - Formal specifications
- Models required
  - No control of test
  - Dynamic models/projects
  - Formal specifications

# Some tools available

	NModel	Spec Explorer	ModelJunit	PBGT
Dependencies	.NET	Visual Studio	JUnit	Eclipse, Chrome
Source	C#	C#	Java	(Web/Android)
Generates Tests	Yes	Yes	Yes	Yes
Visualization	No	Yes	No	Yes
	Spec Explorer predecessor	NModel successor Visual Studio add-in		Tests GUIs only

List of other open source tools:

- <http://robertvbinder.com/open-source-tools-for-model-based-testing/>

# Selected tool: PBGT

## Pattern Based GUI Testing Tool

- Tests web and Android GUIs
- Simple modeling using a drag-and-drop GUI, no coding involved
- Test check creation, with custom field inputs (i.e.: Forms)
- Field mapping through the source GUI (i.e.: Web browser - Chrome)
- Custom coverage settings in test case creation
- Generates and runs the GUI tests automatically

# PBGT - Pattern Based GUI Testing

## UI Pattern Example



Outlook

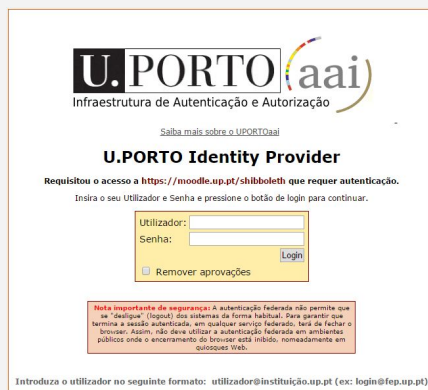
Conta Microsoft O que é isto?

E-mail ou telefone

Palavra-passe

☐ Manter a sessão iniciada

Iniciar sessão



U.PORTO aai

Infraestrutura de Autenticação e Autorização

[Saiba mais sobre o U.PORTO aai](#)

**U.PORTO Identity Provider**

Requisitou o acesso a <https://moodle.up.pt/shibboleth> que requer autenticação.

Insira o seu Utilizador e Senha e pressione o botão de login para continuar.

Utilizador:

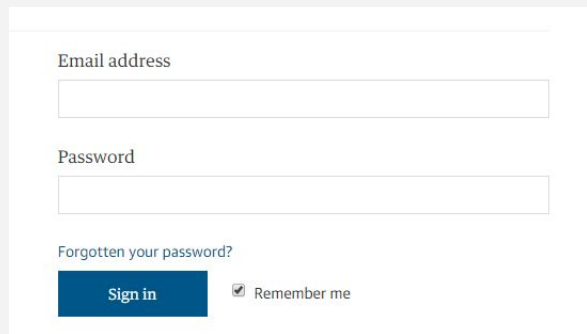
Senha:

Login

☐ Remover aprovações

Nota importante de segurança: A autenticação federada não permite que se "deslogue" (logout) dos sistemas de forma habitual. Para garantir que termina a sessão autenticada, em qualquer serviço federado, terá de fechar o browser. Assim, não deve utilizar a autenticação federada em ambientes públicos onde o encerramento do browser está inhibido, nomeadamente em computadores Web.

Introduza o utilizador no seguinte formato: utilizador@instituição.up.pt (ex: login@fep.up.pt)




Email address

Password

Forgotten your password?

Sign in

☒ Remember me



Autenticação

Utilizador:

Utilizador

Senha:

Senha

Iniciar sessão

Username

Password

Button

LOGIN UI  
PATTERN



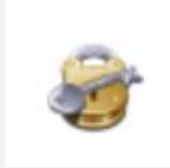
# PBGT - Pattern Based GUI Testing

## UI Patterns

### Behavioral Elements



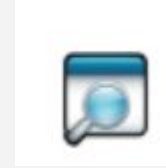
Call



Login



MasterDetail

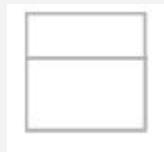


Find



Sort

### Structural Elements



Group



Form

# PBGT - Pattern Based GUI Testing Tool

## UI Test Pattern Definitions

UI Test Pattern describes a generic test strategy that can be defined as:

**< Goal, V, A, C, P >**

Where:

- ❑ **Goal** is the ID of the test;
- ❑ **V** is a set of pairs {[variable, inputData]} relating test input data with the variables involved in the test;
- ❑ **A** is the sequence of actions to perform during test case execution;
- ❑ **C** is the set of possible checks to perform during test case execution and;
- ❑ **P** is a Boolean expression (precondition) defining the conditions over variables that determine when it is possible to execute the test.

# PBGT - Pattern Based GUI Testing Tool

## UI Login Pattern Test Case Specification

Login example:

- ❑ **Goal** - Valid;
- ❑ **V** - {[username, validUsername],[password, validPassword]}
- ❑ **A** - Actions:
  1. Insert "validUsername" in username
  2. Insert "validPassword" in password
  3. Click the button
- ❑ **C** - Check:
  1. Page changed
  2. validUsername appears in the page
- ❑ **P** - execute only if ...

# PBGT - Pattern Based GUI Testing

Videos and Exercises