

# 1 Introdução

Testes são importantes técnicas de controle da qualidade do software. Entretanto, testes tendem a ser pouco eficazes devido à inadequação das ferramentas de teste existentes [NIST, 2002]. São, então, necessários estudos visando desenvolver ferramentas mais eficazes, ou seja, que consigam diminuir o esforço gasto na área de testes. Neste trabalho procuramos explorar tabelas de decisão como instrumento capaz de apoiar a geração automática de casos de teste e o registro do oráculo necessário para verificar automaticamente se o resultado obtido está em conformidade com o especificado.

Existem três grandes classes de teste, a saber: testes estruturais (caixa branca), testes funcionais (caixa preta) e testes mistos (caixa cinza). Nos testes de caixa branca, a geração das massas de teste leva em conta o código fonte do artefato a ser testado. Nos testes caixa preta, a geração das massas de teste leva em conta somente a especificação, ignorando completamente a implementação. Nos testes mistos, leva-se em conta a especificação e algum aspecto da implementação, por exemplo, a especificação das estruturas de dados utilizadas, ou a organização do tratamento de exceções. Para todo teste criado existe um resultado esperado, denominado oráculo. Em todos os tipos de testes os oráculos são criados a partir da especificação ou de alguma propriedade inerente ao artefato sendo desenvolvido. A figura 1 [STAA,2009] ilustra as atividades, e os correspondentes resultados, que são realizadas ao gerar massas de teste.

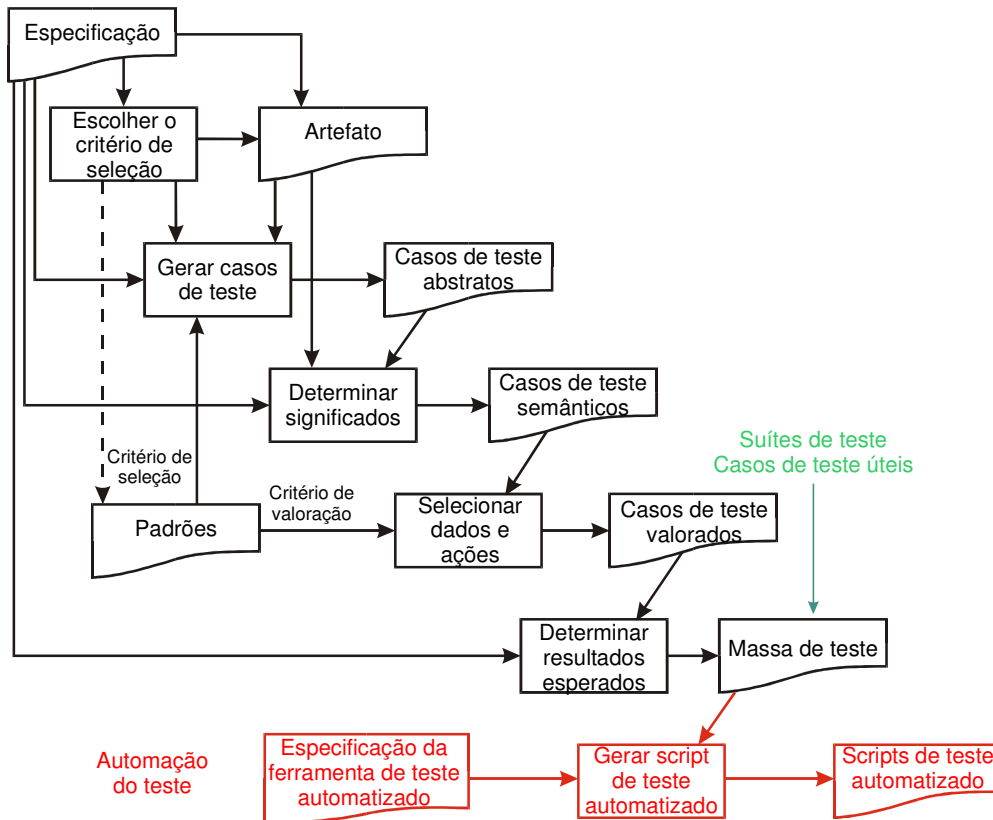


Figura 1: Fases para geração dos testes.

Uma boa parte das técnicas de geração de casos de teste [MYERS, 2004] utiliza explicitamente ou implicitamente tabelas de decisão como instrumento intermediário na geração dos casos de teste específicos. Em linhas gerais os métodos de geração analisam as especificações e/ou o código e produzem uma série de condições que os diversos casos de teste devem satisfazer – os casos de teste abstratos. Estas condições precisam ser traduzidas para as condições a serem satisfeitas pelos dados utilizados nos testes – os casos de teste semânticos. A lista dessas condições semânticas pode ser organizada como uma tabela de decisão [MYERS, 2004]. Isto sugere a possibilidade de desenvolver geradores automáticos de casos de teste capazes de produzirem casos de teste valorados a partir das tabelas de decisão. Sugere também que as tabelas de decisão indiquem os oráculos necessários para verificar se a execução dos testes resultou de acordo com o esperado. Desta forma, podem-se gerar as massas de teste úteis, ou, como também chamadas, as suítes de teste. A partir dos casos de teste úteis e da especificação das ferramentas de teste automatizadas, podem ser criados os scripts de teste a elas destinadas. Ainda não existe na literatura nenhum projeto que tenha

utilizado a tabela de decisão para o planejamento do teste e nem a partir deste gerar a partir deste os casos de teste. Esta seqüência de trabalho permite não somente automatizar a geração da massa de testes como também automatizar a execução dos próprios testes.

### **1.1.Objetivo da Dissertação**

Essa pesquisa tem como objetivo implementar um editor de tabelas de decisão e os correspondentes geradores de casos de teste e de scripts de teste e, a seguir, avaliar os benefícios que o uso de tabelas de decisão pode trazer ao processo de automação dos testes, tanto no planejamento (geração de casos de testes valorados a partir dos casos de teste semânticos) quanto na execução dos casos de teste.

Para a parte de planejamento, usaremos tabelas de decisão como entrada para a geração dos casos de testes valorados. Pela estrutura bem definida das tabelas de decisão pode-se verificar se o conjunto de casos de teste é completo e não ambíguo. A utilização e avaliação explicitamente das tabelas de decisão ainda não foi apresentado na literatura. Além disso, pode-se verificar se cada caso de teste contém oráculos. Por ora a verificação da coerência entre a combinação de condições e os respectivos oráculos será realizada por inspeção ou revisão realizada por uma pessoa ou equipe.

Depois do planejamento do teste, ou seja, do preenchimento completo da tabela de decisão, a próxima etapa é a geração dos casos de teste. Para cada caso de teste, representado através das colunas da tabela de decisão, são gerados dados aleatórios considerando as condições descritas nas tabelas. A última etapa é a transformação dos casos de teste em scripts de teste utilizando o framework FEST [FEST,2009]. Esse framework gera scripts com a estrutura de um script JUnit, só que adicionando a interação com a interface. É importante lembrar que cada caso de teste possui um ou mais oráculos, e que estes precisam ser capazes de verificar o resultado levando em conta os dados utilizados a cada teste.

### **1.2.Tabela de Decisão**

Uma tabela de decisão é uma ferramenta que tem como objetivo expressar as combinações de condições necessárias para a execução de um determinado

conjunto de ações [MYERS, 2004]. Uma tabela de decisão é particionada em duas regiões, a saber: condições e ações. A região de condições estabelece as condições e os possíveis valores para elas. A região de ações identifica as ações que devem ser realizadas em resposta a uma determinada combinação de condições. Os valores das condições e quais ações devem ser acionadas são chamadas de regras, ou seja, cada coluna da tabela de decisão é uma regra.

A tabela de decisão a seguir representa a escolha de qual será o transporte utilizado para ir a um jogo de futebol.

	Como irá ao jogo?	R1	R2	R3	R4
C1	Estádio perto da residência?	Sim	Sim	Não	Não
C2	Dia chuvoso	Não	Sim	Sim	Não
A1	Ir de carro		X	X	X
A2	Ir a pé	X			

Uma das grandes vantagens da tabela de decisão é que é possível verificar se é não ambígua e completa.

No caso da utilização de tabela de decisão para testes, cada uma das colunas representa um caso de teste semântico. Já as ações podem ser entendidas como sendo os oráculos, ou seja, as indicações dos valores esperados ou das formas de calculá-los em função dos valores utilizados nas condições. Para a geração do caso de teste valorado, utilizado futuramente na criação do script para execução dos testes, é em geral possível gerar automaticamente os dados. Essa geração pode ser realizada através de gramáticas geradoras ou de uma seleção randômica de possíveis valores.

### 1.3. Organização da Dissertação

A dissertação está organizada em seis capítulos. O segundo capítulo apresenta o estado da arte para três assuntos: geração automática de casos de teste, tabela de decisão e geração automática de dados de teste. No terceiro capítulo é apresentado o processo de teste desenvolvido e no quarto capítulo são apresentadas as ferramentas que auxiliaram o processo.

Os próximos dois capítulos apresentam experimentos feitos para a avaliação do processo, sendo o capítulo cinco de casos pequenos desenvolvidos apenas para a dissertação e o capítulo seis de um estudo de caso do processo realizado em um sistema em produção. Por fim, são apresentadas as conclusões e trabalhos futuros.