

Certified Tester

Model-Based Tester Syllabus

CT-MBT

versão 1.0

International Software Testing Qualifications Board



Direitos autorais

Copyright Notice: Este documento pode ser copiado na sua totalidade, ou ter extratos feitos, se a fonte for reconhecida na sua reprodução.

Copyright © International *Software* Testing Qualifications Board (hereinafter called ISTQB®).

Model-Based Tester Working Group: Stephan Christmann (chair), Anne Kramer, Bruno Legeard, Armin Metzger, Natasa Micuda, Thomas Mueller, Stephan Schulz; 2014-2015.

Histórico da Revisão

Versão	Data	Observações
Beta 1.0	08/05/2015	Primeira versão Beta
2015	23/10/2015	Versão final após Beta.
2015br	07/03/2017	Tradução para a Língua Portuguesa

Histórico da versão BSTQB

Versão	Data	Observações
1	14/06/2023	Padronização do layout com o ISTQB

Índice

Direitos autorais	2
Histórico da Revisão.....	3
Índice	4
Agradecimentos.....	6
0 Introdução	7
0.1 Objetivo deste documento	7
0.2 Visão Geral	7
0.3 Objetivos de aprendizagem.....	7
1 Introdução ao teste baseado em modelo ^[90 min]	8
1.1 Objetivos e motivações para o MBT	9
1.1.1 Motivações principais para o MBT	9
1.1.2 Enganos e armadilhas do MBT	10
1.2 Atividades e artefatos no processo MBT	11
1.2.1 Atividades específicas do MBT	11
1.2.2 Artefatos essenciais do MBT (entradas e saídas)	12
1.3 Integração do MBT nos ciclos de desenvolvimento do software.....	12
1.3.1 MBT em ciclos de vida de desenvolvimento de software sequencial e iterativo.....	12
1.3.2 Suporte à engenharia de requisitos	14
2 Modelagem MBT [250 min].....	15
2.1 Modelando MBT	16
2.1.1 Atividades de modelagem MBT.....	16
2.1.2 Assuntos e focos dos modelos MBT	17
2.1.3 Modelos MBT dependem dos objetivos de teste	18
2.2 Linguagem para modelos MBT	18
2.2.1 Principais categorias de linguagem de modelagem para MBT	18
2.2.2 Categorias de linguagens relevantes para diferentes sistemas e objetivos de projetos ..	20
2.3 Boas práticas para atividades de modelagem MBT	20
2.3.1 Características de qualidade para modelos MBT	20
2.3.2 Erros e armadilhas típicas na modelagem do modelo MBT	21
2.3.3 Vinculando requisitos e informações relacionadas ao processo do modelo MBT	21
2.3.4 Diretrizes de modelagem MBT	22
2.3.5 Reutilização de modelos de requisitos ou modelos de sistemas existentes	23
2.3.6 Ferramentas de suporte para atividades de modelagem	23
2.3.7 Desenvolvimento, revisão e validação de modelos iterativos.....	24
3 Critérios de seleção para geração de caso de teste ^[205 min]	25
3.1 Classificação dos critérios de seleção do teste MBT	26

3.1.1	Critérios de seleção dos testes.....	26
3.1.2	Seleção de casos de teste na prática.....	27
3.1.3	Exemplos de critérios de seleção de testes.....	27
3.1.4	Relação com as técnicas de modelagem de testes do nível de fundamental.....	28
3.2	Aplicando critérios de seleção de teste	29
3.2.1	Grau de automação na geração de teste.....	29
3.2.2	Prós e contras de critérios específicos de seleção de testes	29
3.2.3	Boas práticas de seleção de testes MBT	30
4	Implementação e execução do teste MBT ^[120 min]	31
4.1	Especificações da implementação e execução do teste MBT.....	32
4.1.1	Casos de teste abstratos e concretos no contexto MBT	32
4.1.2	Diferentes tipos de execução de teste	32
4.1.3	O impacto das mudanças nos artefatos do MBT	33
4.1.4	Atividades de Adaptação ao Teste em MBT	34
5	Avaliação e implantação de uma abordagem MBT ^[60 min]	36
5.1	Avaliar uma implantação MBT	37
5.1.1	Fatores de ROI para implantação ao MBT	37
5.1.2	Objetivos organizacionais e sua relação com as características do MBT	38
5.1.3	Métricas e principais indicadores de performance.....	39
5.2	Gerenciar e monitorar a implantação de uma abordagem MBT.....	40
5.2.1	Boas práticas na implantação do MBT.....	40
5.2.2	Fatores de custo de MBT.....	40
5.2.3	Integração da ferramenta MBT	41
	Abreviaturas	43
	Marcas Registradas	44
	Referências	45
	Apêndice A – Linguagem de Modelagem Simples	46

Agradecimentos

Este documento foi produzido pelo International *Software* Testing Qualifications Board Foundation Level Working Group.

A equipe do ISTQB® Foundation Level Certified Model-Based Tester (CT-MBT) agradece ao grupo de revisão e a todos os Boards Nacionais pelas suas sugestões.

No momento em que o CT-MBT Syllabus foi concluído, o Model-Based Tester Working Group teve a seguinte adesão: Stephan Christmann (MBT Working Group Chair), Anne Kramer, Bruno Legeard (MBT Author Group Co-Chair), Armin Metzger (MBT Author Group Co-Chair), Natasa Micuda (MBT Exam Question Group Chair), Thomas Mueller (ISTQB CTFL Working Group Chair), Stephan Schulz (MBT Review Group Chair).

Autores: Stephan Christmann, Lars Frantzen, Anne Kramer, Bruno Legeard, Armin Metzger, Thomas Mueller, Ina Schieferdecker, Stephan Weissleder.

Grupo de perguntas de exames: Bob Binder, Renzo Cerquozzi, Debra Friedenberg, Willibald Krenn, Karl Meinke, Natasa Micuda, Michael Mlynarski, Ana Paiva.

Grupo de revisores de questões de exame: Eddie Jaffuel, Ingvar Nordstrom, Adam Roman, Lucjan Stapp

Grupo de revisão: Stephen Bird, Thomas Borchsenius, Mieke Gevers, Paul Jorgensen, Beata Karpinska, Vipul Kocher, Gary Mogyorodi, Ingvar Nordström, Hans Schaefer, Romain Schmechta, Stephan Schulz, Szilard Szell, Tsuyoshi Yumoto.

A equipe agradece também as seguintes pessoas, dos Boards Nacionais e da comunidade de especialistas em testes baseados em modelos, que participaram na revisão, comentários e votação do Foundation Level Certified Model-Based Tester Syllabus: Patricia Alves, Clive Bates, Graham Bath, Rex Black, Armin Born, Bertrand Cornanguer, Carol Cornelius, Winfried Dulz, Elizabeta Fournernet, Debra Friedenberg, Kobi Halperin, Kimmo Hakala, Matthias Hamburgo, Kari Kakkonen, Juran van de Laar, Alon Linetzki, Judy McKay, Ramit Manohar, Rik Marselis, Natalia Meergus, Ninna Morin, Klaus Olsen, Tal Pe'er, Michael Pilaeten, Meile Posthuma, Ian Ross, Mark Utting e Ester Zabar.

Este documento foi formalmente aprovado para ser liberado pela Assembléia Geral do ISTQB® em 23 de outubro de 2015.

0 Introdução

0.1 Objetivo deste documento

Este syllabus forma a base para o *ISTQB® Certified Model-Based Tester, Foundation Level*. É uma extensão de especialista para o *ISTQB® Certified Tester Foundation Level (CTFL)*. O ISTQB® fornece este syllabus da seguinte forma:

1. Para os Conselhos Nacionais, para traduzir para o seu idioma local e para credenciar provedores de treinamento. Os Conselhos Nacionais podem adaptar o syllabus às suas necessidades linguísticas específicas e modificar as referências para se adaptarem às suas publicações locais.
2. Para os Exames, para derivar questões de exame em sua língua local com base nos objetivos de aprendizagem.
3. Aos fornecedores de formação, para produzir material de formação e determinar métodos de ensino adequados.
4. Para os candidatos de certificação, como uma fonte para se preparar para o exame.
5. Para a comunidade internacional de *software* e engenharia de sistemas, para avançar na profissão de *software* e testes de sistemas e como base para livros e artigos.

O ISTQB® pode permitir que outras entidades usem este syllabus para outros fins, desde que busquem e obtenham autorização prévia por escrito.

0.2 Visão Geral

O Syllabus ISTQB® Foundation Level Certified Model-Based Tester é um módulo especializado no Foundation Level. A certificação ISTQB® Certified Tester Foundation Level (CTFL) é um pré-requisito necessário para o exame ISTQB® Foundation Level Certified Model-Based Tester (CT-MBT).

Uma pessoa com o Certificado CT-MBT amplia seu entendimento dos testes adquiridos no Nível de Fundamental para habilitar os recursos necessários para o processo, metodologia e técnicas de Testes Baseados em Modelos (MBT).

Resultados esperados de Negócio, trajetórias de carreira para os testadores e o público-alvo para este plano de estudos são apresentados no documento de síntese relacionado [ISTQB_FLMBT_OVW].

0.3 Objetivos de aprendizagem

Os Objetivos de Aprendizagem suportam os Resultados de Negócios e são usados para criar exames para alcançar a certificação CT-MBT.

Em geral, todas as partes deste syllabus são examináveis a um nível K1, isto é, o candidato reconhecerá, e recordará termos e conceitos estabelecidos no syllabus. Os objetivos de aprendizagem relevantes nos níveis K2 e K3 são indicados no início de cada capítulo dentro deste syllabus.

1 Introdução ao teste baseado em modelo [90 min]

Conceitos-chave

Modelo MBT, teste baseado em modelo

Objetivos de Aprendizagem

1.1 Objetivos e motivações para MBT

FM-1.1.1 (K2): Descrever os benefícios esperados do MBT.

FM-1.1.2 (K2): Descrever expectativas enganosas e armadilhas de MBT.

1.2 Atividades e artefatos MBT no processo fundamental de teste

FM-1.2.1 (K2): Resumir as atividades específicas do MBT quando implementadas em um processo de teste.

FM-1.2.2 (K1): Lembrar dos principais artefatos MBT (entradas e saídas).

1.3 Integração do MBT no ciclo de vida do desenvolvimento de software

FM-1.3.1 (K2): Explicar como o MBT se integra aos processos do ciclo de vida do desenvolvimento de *software*.

FM-1.3.2 (K2): Explicar como o MBT suporta a engenharia de requisitos.

1.1 Objetivos e motivações para o MBT

O Teste Baseado em Modelo (MBT) é uma abordagem avançada de teste usando modelos para elaboração dos testes. Ele estende e suporta técnicas de modelagem de teste clássico, como particionamento de equivalência, análise de valor de limite, tabela de decisão, transição de estado e teste de caso de uso. A ideia básica é melhorar a qualidade e a eficiência do projeto do teste e as atividades de implementação dos testes

- Criando um modelo MBT abrangente, normalmente usando ferramentas, com base nos objetivos do teste do projeto.
- Fornecendo um modelo MBT como uma especificação de projeto de teste. Este modelo inclui muita informação formal e detalhamento, que é suficiente para gerar automaticamente os casos de teste diretamente a partir do modelo MBT.

O MBT e seus artefatos estão estreitamente integrados com os processos da organização, bem como com os métodos, ambientes técnicos, ferramentas e quaisquer processos específicos do ciclo de vida.

1.1.1 Motivações principais para o MBT

Existem dois aspectos principais das atividades de teste que fundamentalmente fornecem a motivação para o MBT e descrevem-no como é suportada a melhoria da qualidade dos testes:

Eficácia

- A modelagem é um processo que promove uma comunicação estreita com os stakeholders.
- Uma comunicação aprimorada ajuda a criar uma percepção e compreensão comuns dos requisitos no domínio dado e a detectar possíveis mal-entendidos.
- No caso de modelos gráficos do MBT, os stakeholders do projeto (p. ex., analistas de negócios) podem ser envolvidas com mais facilidade.
- A modelagem suporta a melhoria contínua de competências para testadores em um determinado domínio.
- O nível de abstração do modelo MBT facilita a identificação de partes de um sistema onde os testes provavelmente encontrarão problemas.
- A possibilidade de gerar e analisar casos de teste antes de existir qualquer sistema real.

Eficiência

- A modelagem precoce e a verificação e validação de modelos ajudam a evitar defeitos nas fases iniciais do processo de desenvolvimento. Os modelos MBT podem ser compartilhados com os stakeholders do projeto (p. ex., analistas de negócios, desenvolvedores) para verificar os requisitos e identificar lacunas existentes. Dessa forma, o MBT suporta o princípio do "teste precoce".
- Os artefatos de projetos anteriores de MBT podem ser reutilizados para o desenvolvimento de testes.
- O MBT suporta automação (p. ex., para gerar *testware*) e ajuda a reduzir defeitos que podem ser introduzidos quando o *testware* é criado e mantido manualmente.

- Diferentes conjuntos de testes podem ser gerados a partir do mesmo modelo (p. ex., com diferentes critérios de seleção de teste) promovendo uma adaptação eficiente às mudanças na base de teste ou nos objetivos do teste.
- O MBT pode ser usado para vários objetivos de teste e para cobrir diferentes níveis e tipos de teste.
- O MBT ajuda a reduzir os custos de manutenção quando os requisitos mudam porque o modelo fornece um único ponto de manutenção.

1.1.2 Enganos e armadilhas do MBT

As equipes de teste que estão indo para o MBT devem ter expectativas realistas das vantagens e limitações desta abordagem. As expectativas, equívocos e enganos incluem:

MBT resolve todos os problemas

O MBT não substitui as técnicas de modelagem de teste tradicionais (ver [ISTQB_FL_SYL]), mas as suporta de uma forma que permite aos testadores melhorar sua compreensão do domínio e realizar testes com mais eficiência e eficácia. Uma equipe de teste que ignora ou usa incorretamente essas técnicas de modelagem de teste não resolverá este problema introduzindo o MBT.

MBT é apenas uma questão de ferramental

O suporte de ferramental apropriado é essencial para o sucesso do MBT, mas adquirir a ferramenta não deve ser o primeiro passo. Em vez disso, a decisão de introduzir o MBT deve basear-se na definição de objetivos mensuráveis no que diz respeito às melhorias nos testes. O MBT afeta todo o processo de teste (capítulo 1.2). Portanto, a introdução do MBT requer um forte apoio da gerência para liderar as mudanças de processo e ferramenta na organização.

Os modelos estão sempre corretos

Como no projeto de teste manual, o testador pode introduzir defeitos ao criar o modelo MBT. O MBT exige verificação completa e validação através da aplicação de análises, modelo de análise estática, simulação de modelo e assim por diante. Qualquer alteração no modelo se propaga para todos os *testwares* gerados relacionados ao elemento do modelo modificado, portanto, cada alteração deve ser cuidadosamente revisada antes da implementação.

A explosão do caso de teste ocorrerá

A aplicação do MBT melhora os métodos e a cobertura na modelagem do teste. No caso de geração de casos de teste puramente combinatório, pode levar a uma explosão do caso de teste. Este desafio pode ser resolvido através da adaptação das estratégias e algoritmos de geração de casos de teste e da aplicação de mecanismos de filtragem inteligentes.

1.2 Atividades e artefatos no processo MBT

1.2.1 Atividades específicas do MBT

Ao implementar testes baseados em modelos, o processo de teste inclui atividades específicas do MBT que normalmente não são usadas na análise e na modelagem do teste clássico. Esses incluem:

- As atividades de modelagem MBT (atividades relacionadas à administração do MBT, ao desenvolvimento e integração da abordagem de modelagem, a definição de diretrizes de modelagem, ao desenvolvimento da estrutura do modelo, ao desenvolvimento de elementos do modelo, por exemplo, diagramas específicos do MBT, atividades relacionadas com ferramentas etc.).
- A geração de *testware* com base na abordagem MBT e critérios de seleção, por exemplo, geração de casos de teste.

No mínimo, o teste baseado em modelo impacta a análise de teste e as atividades de projeto de teste no processo descrito no *Foundation Level* [ISTQB_FL_SYL]. Dependendo dos objetivos do teste, o MBT também pode ser usado com um escopo mais amplo relativo a todas as partes do processo fundamental de teste. As seguintes atividades específicas do MBT devem ser consideradas:

- O planejamento e o controle do teste podem incluir a implementação das atividades específicas do MBT do processo fundamental de teste (ferramentas MBT, diretrizes, métricas específicas, artefatos MBT como parte da linha de base para o planejamento de projetos etc.).
- A análise de teste e modelagem podem incluir atividades avançadas de modelagem MBT, critérios de seleção de teste e métricas de cobertura de teste.
- A Implementação e execução do teste podem incluir geração e adaptação ao teste MBT.
- A avaliação dos critérios de saída e relatório podem incluir métricas avançadas de cobertura (com base em informações estruturais ou explícitas do modelo) e análise de impacto baseada em MBT.
- As atividades de encerramento do teste podem incluir o estabelecimento de bibliotecas modelo para reutilização em futuros projetos.

Um ponto de foco do impacto do MBT no processo de teste é a automação de processos e a geração de artefatos. MBT promove um deslocamento do projeto do teste em fases mais adiantadas do projeto comparado com o projeto clássico do teste. Ele pode servir como um método de verificação de requisitos antecipados e pode promover uma melhor comunicação, especialmente usando modelos gráficos. Testes baseados em modelos frequentemente aumentam os testes tradicionais. Isso fornece uma oportunidade para a equipe de teste verificarem a consistência entre os casos de teste MBT gerados automaticamente e os casos de teste tradicionais criados manualmente.

Embora as atividades do processo fundamental de teste usando MBT sejam semelhantes às realizadas no processo de teste tradicional, a implementação pode mudar significativamente. Conforme descrito no capítulo 1.1, o MBT:

- Tem impacto na qualidade, esforço, comunicação e stakeholders envolvidos no processo de teste.
- Desloca as atividades de projeto de teste para fases anteriores no processo de teste.

Para garantir a adoção de MBT, é importante estabelecer o aceite para o método e as mudanças relacionadas com as equipes que trabalham com MBT.

1.2.2 Artefatos essenciais do MBT (entradas e saídas)

Os modelos MBT podem ser obtidos desenvolvendo modelos MBT dedicados ou reutilizando modelos desenvolvidos para o projeto do sistema (ver capítulo 2.3.5). Os modelos MBT suportam diferentes níveis de abstração e informações de teste específicas de acordo com o conceito de modelagem aplicado. Os artefatos obtidos a partir dos modelos MBT refletirão estes diferentes níveis de abstração.

De acordo com o nível de informação e abstração, o MBT é integrado ao processo de teste com diferentes artefatos de entrada e saída.

Artefatos de entrada:

- Estratégia de teste.
- A base de teste, incluindo os requisitos e outros objetivos de teste, condições de teste, informações orais e modelos ou modelos existentes.
- Relatórios de incidentes e defeitos, *logs* de teste e *logs* de execução de teste de atividades ou de execução de teste anteriores.
- Diretrizes de processo e métodos, documentos de ferramentas.

Os artefatos de saída incluem diferentes tipos de *testware*, como:

- Modelos MBT.
- Partes do plano de teste (características a serem testadas, ambiente de teste, ...), cronograma de teste, métricas.
- Cenários de testes, conjuntos de testes, cronogramas de execução, especificações do projeto de teste.
- Casos de teste, especificações de procedimentos de teste, dados de teste, scripts, camada de adaptação de teste (especificações e código).
- Matriz de rastreabilidade bidirecional entre os testes gerados e a base de teste, especialmente os requisitos, e os relatórios de defeitos.

1.3 Integração do MBT nos ciclos de desenvolvimento do software

1.3.1 MBT em ciclos de vida de desenvolvimento de software sequencial e iterativo

Duas categorias principais de modelos de ciclo de vida de desenvolvimento de *software* são consideradas nesse capítulo: sequencial, como o Modelo V, e iterativo-incremental, como o desenvolvimento ágil. Em ambos os casos, a abordagem MBT deve ser adaptada principalmente aos objetivos de teste do projeto para que os benefícios do MBT possam ser obtidos de acordo com os objetivos do teste.

As organizações implementam de maneiras diferentes os ciclos de vida do desenvolvimento de *software*, assim, o processo de teste utilizando MBT deve ser adaptado em conformidade. Por

exemplo, MBT pode se concentrar em testes de aceite em um projeto e em testes de sistemas automatizados em outro, dependendo dos objetivos do teste do projeto.

A seguir, são comuns os ciclos de desenvolvimento de *software* sequencial e iterativo:

Níveis de Teste

- O MBT é utilizado principalmente em níveis de teste mais elevados (integração, sistema e aceite), devido à sua capacidade de abstrair a complexidade dos requisitos funcionais e do comportamento esperado.
- O MBT é pouco usado no teste de componente (unidade), mas algumas abordagens são baseadas em técnicas de anotação de código.

Tipos de Teste

- O MBT é usado principalmente para testes funcionais com modelos MBT que representam o comportamento esperado do sistema e/ou do seu ambiente.
- Os modelos MBT enriquecidos ou dedicados podem ser usados para testes não-funcionais (segurança, carga, estresse, confiabilidade).

Os testadores normalmente possuem os modelos MBT, mas podem optar por compartilhá-los. Por exemplo, os modelos (ou partes deles) podem ser compartilhados com os stakeholders não técnicos para validação e verificação dos requisitos. Os desenvolvedores podem estar interessados nos modelos para fins de automação de teste, particularmente quando os scripts automatizados gerados a partir do modelo MBT serão usados para a integração contínua para fornecer retorno rápido e frequente.

Atividades específicas para MBT em um ciclo de vida de desenvolvimento de *software* sequencial incluem:

- Integração dos testes baseados em modelos como parte da estratégia de teste para o projeto, em relação à avaliação de risco, contexto do projeto e objetivos do teste, incluindo a rastreabilidade de requisitos para elementos do modelo MBT.
- Iniciar a modelagem MBT o mais rápido possível no projeto:
 - Estimular a comunicação entre stakeholders.
 - Permitir a detecção antecipada de requisitos pouco claros, incompletos ou inconsistentes.
- Adaptar as atividades e papéis do planejamento de testes para incluir:
 - Novos elementos para o *testware* (p. ex., modelos MBT, critérios de seleção de teste).
 - Relatórios de progresso sobre as atividades do MBT.

As principais adaptações de MBT em um ciclo de vida ágil [ISTQB_FLAT_SYL] são as seguintes:

- Os modelos MBT são desenvolvidos de forma iterativa e incremental e os testes são gerados de acordo com o conteúdo da iteração.
- Histórias de Usuários fazem parte da base de teste. Cada história de usuário coberta está vinculada dentro do modelo para gerenciar automaticamente a rastreabilidade bidirecional entre história de usuários e testes.
- O MBT pode ser usado para implementar o método de desenvolvimento orientado por teste de aceite (ATDD).

- No que diz respeito à boa prática ágil, os testadores que utilizam testes baseados em modelos fazem parte da equipe ágil, juntamente com os desenvolvedores e representantes de negócios - consulte o capítulo 1.1 do [ISTQB_FLAT_SYL].

1.3.2 Suporte à engenharia de requisitos

O MBT ajuda na engenharia de requisitos das seguintes maneiras:

- Facilita a comunicação entre negócios, desenvolvimento e testes em torno do comportamento esperado do *software* a ser desenvolvido, fornecendo modelos gráficos MBT, como modelos de processos de negócios ou diagramas de estado. Esses modelos ajudam todos os stakeholders a discutir e validar os detalhes do comportamento esperado.
- Esclarece e melhora a qualidade dos requisitos ou histórias de usuários compartilhando modelos MBT completos ou parciais com representantes de negócios ou desenvolvedores para produzir uma compreensão comum do comportamento esperado do *software*. Testes gerados a partir de modelos MBT incluem possíveis cenários de usuário que podem ser revistos por representantes de negócios ou desenvolvedores.
- Suporta a validação antecipada dos requisitos, mesmo nos estágios em que eles ainda estão sujeitos a alterações.

A engenharia de requisitos e o MBT trabalham juntos. Os requisitos e riscos relacionados são insumos para modelagem MBT e atividades de geração de teste, e as atividades MBT podem automatizar a criação e manutenção de *links* de rastreabilidade bidirecional entre requisitos e testes.

2 Modelagem MBT [250 min]

Conceitos-chave

Modelo de teste

Objetivos de Aprendizagem

2.1 Modelagem MBT

FM-2.1.1 (K3) Desenvolver um modelo MBT simples (menos de 15 elementos de modelagem) para um objeto de teste com objetivos de testes predefinidos usando uma linguagem de modelagem baseada no fluxo de trabalho (consulte capítulo 9.1).

FM-2.1.2 (K3) Desenvolver um modelo MBT simples (menos de 15 elementos de modelagem) para um objeto de teste com objetivos de teste predefinidos usando uma linguagem de modelagem baseada na transição de estado (consulte capítulo 9.1).

FM-2.1.3 (K2) Classificar um modelo MBT com relação ao assunto e ao foco.

FM-2.1.4 (K2) Dar exemplos de como um modelo MBT depende dos objetivos dos testes.

2.2 Linguagem para Modelos MBT

FM-2.2.1 (K1) Lembrar exemplos de categoria de linguagem de modelagem comumente usadas para o MBT.

FM-2.2.2 (K1) Lembrar representantes típicos de categoria de linguagem de modelagem relevantes para diferentes sistemas e objetivos de projetos.

2.3 Boas práticas para as atividades de Modelagem MBT

FM-2.3.1 (K1) Lembrar as características de qualidade para modelos MBT.

FM-2.3.2 (K2) Descrever os erros clássicos e as armadilhas durante as atividades de modelagem para o MBT.

FM-2.3.3 (K2) Explicar as vantagens de vincular os requisitos e as informações relativas ao processo do modelo MBT.

FM-2.3.4 (K2) Explicar a necessidade de diretrizes para modelagem MBT.

FM-2.3.5 (K2) Fornecer os exemplos de quando é ou não é apropriada a reutilização de modelos existentes (da fase de requisitos ou fase de desenvolvimento).

FM-2.3.6 (K1) Lembrar os tipos de ferramentas que suportam as atividades específicas de modelagem MBT.

FM-2.3.7 (K2) Resumir o desenvolvimento do modelo MBT iterativo, revisão e validação.

2.1 Modelando MBT

No contexto de teste, os modelos MBT são excelentes para expressar o que deve ser testado, para comunicação entre os stakeholders e para o resumo de todas as informações relevantes do projeto do teste. Os modelos têm também algumas aplicabilidades para atividades de gerenciamento de teste.

Um modelo MBT é desenvolvido com o objetivo de derivar (ou identificar) casos de teste. O modelo MBT deve incluir as informações necessárias para geração posterior de casos de teste, por exemplo, para permitir a seleção razoável de casos de teste de acordo com os objetivos de teste do projeto, para permitir a geração de oráculos de teste e para suportar a rastreabilidade bidirecional entre requisitos e casos de teste gerados.

Projetar o modelo é uma atividade essencial e exigente no MBT. A qualidade do modelo tem forte impacto na qualidade do resultado do processo do teste baseado no MBT. Os modelos MBT geralmente são interpretados por ferramentas e devem seguir uma síntese estrita. Eles também podem suportar outras características de qualidades como as definidas nas diretrizes de modelagem.

2.1.1 Atividades de modelagem MBT

Cada modelo é expresso em uma linguagem de modelagem específica. A linguagem de modelagem define os artefatos que compõem os elementos do modelo e as regras que devem ser seguidas para construir modelos nesta linguagem. Existem linguagens de modelos para expressar estruturas, comportamentos, ou artefatos específicos de domínio, entre outros.

As seguintes questões importantes de modelagem MBT devem ser feitas:

Quais características de qualidade do objeto do teste serão modeladas?

Comumente o foco está na funcionalidade, mas fatores de performance e outros aspectos não funcionais, como segurança, também podem fazer parte do modelo MBT. Para manter os modelos gerenciáveis e o processo de teste baseado em um MBT mais eficiente, recomenda-se modelar apenas os aspectos do sistema ou seu ambiente que sejam relevantes para o teste e para os objetivos do teste.

Que linguagens de modelagem são adequadas?

Cada modelo é expresso em uma linguagem de modelagem específica. A linguagem de modelagem define os artefatos que compõem o modelo e as regras que devem ser seguidas para construir modelos nessa linguagem. Há uma grande variedade de linguagens de modelagem para representar a estrutura, o comportamento ou outros aspectos (p. ex., dados, fluxos de trabalho, protocolos de comunicação) do objeto de teste ou seu ambiente.

Qual é o nível apropriado de abstração?

A abstração é útil para dominar a complexidade e focar o modelo MBT nos objetivos do teste. Entretanto, quanto mais abstrato é o modelo, mais exigente é a adaptação do teste para sua execução. Além disso, o nível de abstração muitas vezes determina o público ou stakeholders que são capazes de discutir o projeto de teste no nível da abstração dada.

As respostas a estas perguntas e o conjunto de ferramentas MBT escolhido influenciarão as atividades MBT.

Nota: No Apêndice (A) são apresentadas duas linguagens simplificadas de modelagem gráfica. A preparação para o treinamento e exames deve abranger o uso das duas linguagens.

2.1.2 Assuntos e focos dos modelos MBT

Três tipos de assuntos são considerados neste syllabus:

Modelo de sistema

O modelo de sistema descreve um sistema como pretende ser. Os casos de teste gerados a partir deste modelo verificam se o sistema está em conformidade com este modelo. Exemplos de modelos de sistemas são diagramas de classes para sistemas orientados por objetos ou diagramas de estados que descrevem os estados e transições de estados de sistemas reativos.

Modelo de ambiente e modelo de utilização

Um modelo de ambiente descreve o ambiente do sistema. Exemplos de modelo de ambiente incluem o Modelo de Cadeia Markov descrevendo o uso esperado do sistema.

Modelo de teste

Um modelo de teste é um modelo de (um ou mais) casos de testes. Isso geralmente inclui o comportamento esperado do objeto de teste e sua avaliação. Exemplos de modelos de teste são descrições de casos de teste (abstratos) ou uma representação gráfica de um procedimento de teste.

Um modelo MBT geralmente combina vários ou todos esses assuntos. Por exemplo, ele pode representar o objeto de teste incluindo elementos de dados, e o uso do objeto de teste em um dado contexto.

O foco de um modelo MBT pode ser estrutural, comportamental ou uma combinação de ambos.

- *Modelos estruturais* descrevem a estrutura estática. Exemplos de modelo de estrutura são diagramas de classes e especificações de interface ou árvores de classificação para modelagem de dados.
- *Modelos comportamentais* descrevem as interações dinâmicas. Exemplos de modelos comportamentais são diagramas de atividades ou modelos de processos de negócios descrevendo atividades e fluxos de trabalho, e diagramas de estados descrevendo entradas e saídas de um sistema.

Um modelo MBT geralmente combina aspectos estruturais (p. ex., a descrição da interface do objeto de teste) e aspectos comportamentais (p. ex., o comportamento esperado do objeto de teste).

2.1.3 Modelos MBT dependem dos objetivos de teste

No desenvolvimento de um modelo MBT é importante considerar o objetivo do teste, pois isso determina o assunto e o foco do modelo. Exemplos de objetivos de teste juntamente com características de modelo adequadas para esse objetivo são apresentados na tabela abaixo.

Objetivo do Teste	Exemplo de Modelo	Tema	Foco
Verifique se o fluxo de trabalho do negócio foi implementado corretamente.	Um modelo de processo de negócio descrevendo o fluxo de trabalho.	Sistema	Comportamento
Verifique se o sistema fornece respostas corretas para uma solicitação quando em um estado específico.	Uma UML (Modelo de linguagem unificado) máquina de estado	Sistema	Comportamento
Verifique a disponibilidade de uma interface.	Uma descrição de estrutura da interface	Sistema	Estrutura
Ganho de confiança de que o objeto de teste será adequado para o uso esperado pelos usuários.	Um modelo de uso descrevendo o comportamento do usuário	Meio ambiente	Comportamento
Verifique a configuração do sistema.	Um modelo de dados usando uma árvore de classificação	Dados	Estrutura

2.2 Linguagem para modelos MBT

2.2.1 Principais categorias de linguagem de modelagem para MBT

Os modelos são denotados pelo uso de linguagens de modelagem. As linguagens de modelagem são definidas das seguintes maneiras:

- Pelo seu conceito (também conhecido como sintaxe abstrata e muitas vezes textualmente descrita, mas somente especificado em meta-modelos).
- Pela sua sintaxe (também conhecida como sintaxe concreta e muitas vezes definida por regras de gramática).
- Pela sua semântica (frequentemente definidos por regras semânticas estáticas e dinâmicas).

Os diagramas são representações de modelos em uma linguagem de modelagem gráfica, como diagramas de classes, diagramas de sequência ou diagramas de máquina de estados no UML.

Existem várias linguagens de modelagem que podem ser usadas pelo MBT. Para selecionar uma boa linguagem adequada para o propósito, o testador precisa conhecer as principais características que diferenciam as linguagens de modelagem para MBT:

Conceito de modelagem

As linguagens de modelagem diferem no conjunto conceitual que suportam. Dependendo do objetivo do teste, os modelos podem representar aspectos estruturais (p. ex., arquitetura, componentes, interfaces de *software*), aspecto de dados (p. ex., formatos e semântica de objetos passivos do *software*) ou aspectos comportamentais (p. ex., cenários, interações, execuções de objetos ativos do *software*).

Formalismo

O grau de formalismo da linguagem de modelagem varia de formalização limitada a formalização total. Embora este último seja o mais rígido e permita uma análise completa dos modelos, o primeiro é muitas vezes mais prático. No mínimo, o idioma deve ter uma sintaxe formal (p. ex., incluindo texto estruturado ou tabelas estruturadas).

Formato de apresentação

Linguagens de modelagem podem usar diferentes formatos de apresentação variando de formatos textuais a gráficos, bem como uma combinação de ambos. Os formatos gráficos são frequentemente mais fáceis de usar e mais fáceis de ler, enquanto os formatos de texto são frequentemente mais fáceis de utilizar e eficientes para criar e manter.

As categorias de linguagem de modelagem incluem:

Linguagens para modelos estruturais

Tais linguagens suportam a especificação de elementos estruturais de *software*, tais como interfaces, componentes e hierarquias. Um exemplo é o diagrama de componentes UML.

Linguagens para modelos de dados

Tais linguagens suportam a especificação de tipos e valores dos dados. Exemplos incluem os diagramas de classe UML e especificações de valor.

Linguagens para modelos comportamentais

Tais linguagens suportam especificações de eventos, ações, reações e/ou interações de *software*. Exemplos incluem atividade UML ou diagramas de interação, máquinas de estado ou *Business Process Modeling Notation* (BPMN).

Linguagens integradas

Tipicamente, uma linguagem de modelagem não se limita a um dos aspectos, mas fornece conceitos para uma série de aspectos. Um exemplo é a própria UML, na qual os diferentes diagramas podem ser usados em combinação para denotar os diferentes aspectos do *software*.

2.2.2 Categorias de linguagens relevantes para diferentes sistemas e objetivos de projetos

A seleção da linguagem de modelagem para MBT está intimamente relacionada aos objetivos originais do projeto para os quais o *software* é desenvolvido e precisa ser testado. Ele também se relaciona com as propriedades do sistema para o qual o *software* é construído. Isso resulta em um conjunto de critérios de linguagem de modelagem. Pode ser necessário avaliar esses critérios, aceitando alguns e eliminando outros, com base na relevância para atributos específicos do sistema.

Exemplos incluem o uso de:

- Diagramas de estado para representar o comportamento esperado de um controle/comando do objeto de teste.
- Diagramas de atividade ou modelos de BPMN para representar o fluxo de trabalho para o teste de ponta a ponta de um sistema de informação.
- Tabelas de decisão e gráficos de causa e efeito para representar regras de negócios para testes de sistema.
- Modelos cronometrados para representar a capacidade de resposta de um objeto de teste para casos de teste cronometrados.

Outros exemplos para a relação de linguagens de modelagem e objetivos do projeto incluem:

- Requisitos não funcionais relativos à certificação de segurança crítica ou de *software* de segurança crítica, onde a certificação requer a vinculação dos requisitos de *software* aos códigos e casos de teste.
- Requisitos não funcionais relativos à documentação de processos para propósitos de auditoria ou certificação onde os modelos são equipados com anotações para documentação.

2.3 Boas práticas para atividades de modelagem MBT

2.3.1 Características de qualidade para modelos MBT

O modelo de qualidade afeta diretamente a saída gerada. As características de qualidades para modelos são:

Qualidade sintática (correção)

O modelo MBT é consistente com as regras referentes a sua descrição formal (linguagem de modelagem, guia de modelagem) para que a geração de casos de teste e dados de teste possam produzir artefatos sem se deparar com problemas devido à sintaxe incorreta.

Qualidade semântica (validade)

O conteúdo do modelo está correto em relação ao que deve descrever; artefatos derivados são “utilizáveis” (scripts de teste podem ser executados, os casos de teste manual podem ser executados pelo testador sem produzir falhas devido a casos de teste incorretos).

Qualidade pragmática (adequação)

O modelo MBT é adequado para o objetivo de teste dado, e para o gerador de teste, de modo que os artefatos derivados satisfaçam as expectativas.

As ferramentas podem verificar a sintaxe e, pelo menos em parte, a semântica de um modelo. Comentários conferem qualidade semântica e pragmática. Os simuladores de modelo completam essas técnicas estáticas, fornecendo verificações dinâmicas da qualidade do modelo.

Recomenda-se desenvolver incrementalmente o modelo MBT e gerar (e executar) os testes derivados repetidamente, verificando assim o modelo, os artefatos derivados e o objeto de teste antecipadamente e regularmente.

2.3.2 Erros e armadilhas típicas na modelagem do modelo MBT

Os recém-chegados ao MBT tendem a cometer uma série de erros comuns na modelagem, incluindo:

- Colocar demasiados ou poucos detalhes no modelo MBT (nível de abstração errado para determinados objetivos de teste).
- Tentar escrever um modelo que cubra tudo.

Os modelos de MBT devem focar nos aspectos que são relevantes para cumprir os objetivos do teste. É melhor escrever dois modelos que enfatizem aspectos específicos (p. ex., um diagrama de estado para verificar a implementação do sistema e um diagrama de atividades para validar o fluxo de trabalho) em vez de um modelo que abranja todos os aspectos.

2.3.3 Vinculando requisitos e informações relacionadas ao processo do modelo MBT

Estabelecer a rastreabilidade entre requisitos, modelos MBT e testes gerados é uma boa prática em testes baseados em modelos. Ao vincular os elementos do modelo com os requisitos, os seguintes benefícios podem ser alcançados:

- A revisão do modelo MBT é facilitada.
- Testes gerados para o modelo MBT podem ser automaticamente vinculados aos requisitos.
- É possível gerar testes com base na seleção de requisitos e decidir quais testes executar primeiro com base na prioridade dos requisitos selecionados.
- Isso dá a oportunidade de medir a cobertura de requisitos por testes gerados pelo MBT.
- Permite que todos os stakeholders (p. ex., testadores e gerentes de teste) analisem o impacto das alterações de requisitos e determinem o escopo necessário dos testes de regressão.
- Geradores de caso de teste podem gerar automaticamente documentação de rastreabilidade (matriz de rastreabilidade).

Há várias opções de implementação para vincular os requisitos aos elementos do modelo. Por exemplo, um símbolo gráfico específico ou palavra-chave textual pode representar o requisito no modelo.

O testador pode adicionar outras informações complementares como novos elementos de modelo com significado específico ou como atributos para elementos de modelo existentes. Exemplos de informações adicionais nos modelos MBT incluem:

- Conteúdo detalhado para especificação do procedimento de teste (sequência de ações para execução de teste).
- Segmentos de *script* de testes (chamada de funções ou palavras-chave).
- Riscos/perigos (relacionados a atributos de qualidade funcionais e não funcionais ou à gestão de projetos).
- Prioridades (de funcionalidade ou testes).
- Duração (estimativa de tempo de execução).
- Equipamento de teste requerido ou regra para testar as configurações do produto.

As informações adicionais ajudam a integrar testes baseados em modelos em todo o processo de teste e suportam o gerenciamento do teste de várias maneiras:

- Informações sobre risco ou prioridade podem ser vinculadas ao modelo MBT e vinculadas a casos de teste gerados, que podem ser usados (p. ex., para priorizar a execução do teste).
- Quaisquer outros constrangimentos e objetivos do projeto podem ser refletidos no modelo MBT e, portanto, podem ajudar a adaptar o planejamento do teste.
- Quando as interfaces (p. ex., interface gráfica do usuário) ainda não estão estabilizadas na especificação, o modelo MBT pode abranger os requisitos funcionais definidos. Em caso afirmativo, os scripts funcionais orientados por palavras-chave podem ser gerados anteriormente e as fases de automação podem começar mais cedo. Uma vez especificadas as interfaces, os scripts de teste automatizados estão prontos e somente a camada de adaptação de teste ainda deve ser implementada.

2.3.4 Diretrizes de modelagem MBT

As diretrizes de modelagem são instruções documentadas sobre como modelar, escrever e ler modelos MBT. Essas diretrizes:

- Apoiam a compreensão e a revisão dos modelos MBT por todos os envolvidos.
- Apoiam a conformidade com os requisitos sintáticos provenientes do processo, do domínio, da organização ou da ferramenta específica.
- Limitam/estendem o escopo da notação selecionada (p. ex., definir um subconjunto de UML ou significados diferentes de elementos).
- Promovem uma sintaxe e semântica semelhantes de modelos MBT de vários autores.
- Ensinam boas práticas.
- Dão suporte para manutenção e reutilização de modelos MBT.

Projetos MBT sem diretrizes de modelagem assumem riscos adicionais em comparação a projetos MBT com diretrizes de modelagem, como testadores podem cometer os erros clássicos de modelagem mais facilmente.

2.3.5 Reutilização de modelos de requisitos ou modelos de sistemas existentes

Assim como no desenvolvimento de modelos MBT a partir de documentos de teste "clássicos", é possível reutilizar modelos de sistemas ou requisitos existentes (p. ex., modelos de processos de negócios ou diagramas de atividades) diretamente como parte de um modelo de MBT, mas essa abordagem tem seus limites. O grau de reutilização depende do modelo dado e dos objetivos do teste. Por exemplo, um diagrama de máquina de estado do projeto do sistema pode servir para testar um sistema de comando e controle, mas os diagramas de classe são inadequados para testes de sistema baseados no fluxo de trabalho.

A reutilização de modelos de projeto de sistema existentes cria um problema de fonte única. Qualquer erro no modelo de modelagem do sistema original se propaga para o modelo MBT. Os testes não são tão independentes do desenvolvimento e implementação do sistema como precisam ser. Portanto, recomenda-se escrever modelos independentes por diferentes autores: um para modelagem do sistema e outro para fins MBT. Isso cumpre a separação de responsabilidades e promove a independência.

Se reutilizados sem quaisquer modificações, os testes derivados dos modelos de modelagem do sistema são apenas testes de verificação (oposição à validação). Eles verificam se a implementação do sistema corresponde à especificação descrita pelo modelo de projeto do sistema.

Para fins de validação, o testador pode iniciar a partir de um modelo existente (p. ex., a partir da elicitação de requisitos) e enriquecê-lo com os aspectos de teste. Uma vez transformado em um modelo MBT, o modelo de projeto inicial do sistema tende a crescer. O testador irá adicionar elementos do modelo (p. ex., transições imprevistas) para testar cenários incomuns ou situações de erro. No final, o modelo MBT resultante reflete a mentalidade do testador em vez do projeto inicial do sistema.

Considere o caso em que o autor do modelo existente não era um testador. Como resultado, os modelos de fases de desenvolvimento anteriores não respeitam necessariamente os requisitos e diretrizes relacionados ao MBT. Um modelo reutilizado não segue necessariamente as práticas recomendadas (dependentes da ferramenta) e pode levar a uma explosão de casos de teste.

2.3.6 Ferramentas de suporte para atividades de modelagem

Os editores de modelos são usados para escrever modelos. Isso pode ser uma ferramenta de modelagem específica ou qualquer outro editor de fluxograma (para modelos gráficos) ou um editor de texto/*script* (para modelos textuais). Em princípio, é possível escrever modelos MBT com lápis e papel.

Enquanto ferramentas de desenho simples podem ser suficientes para fins de documentação, eles não suportam processamento adicional do modelo MBT.

Um editor de modelo MBT pode fornecer elementos de modelo predefinidos e possivelmente algumas verificações sintáticas e semânticas. As ferramentas que executam caminhos diferentes através do modelo, validando assim o modelo MBT, são chamadas de simuladores de modelo.

Para obter casos de teste ou dados de teste do modelo MBT automaticamente, é necessário um gerador de caso de teste (ou dados de teste).

2.3.7 Desenvolvimento, revisão e validação de modelos iterativos

A modelagem de comportamentos mais complexos em vários diagramas pode facilmente levar a situações em que uma revisão do modelo MBT sozinho pode não ser suficiente. Um testador precisa ter certeza de que os testes gerados a partir do modelo MBT irão satisfazer as expectativas. O desenvolvimento iterativo, revisões de modelos, bem como revisões frequentes de artefatos de teste gerados, primeiro pelo testador, mas também por colegas e outros stakeholders envolvidos no ciclo de desenvolvimento de *software*, são formas de gerenciar essa situação.

A aplicação dessas melhores práticas permite:

- Testadores para validar seu ponto de vista sobre os aspectos a serem testados com outros stakeholders.
- Testadores para detectar e corrigir erros, bem como detectar partes que faltam no modelo no início do desenvolvimento do modelo MBT (antes da execução do teste).
- Testadores para identificar e comunicar requisitos incompletos ou inconsistentes.
- Gerentes de testes para gerenciar riscos associados ao projeto.
- A equipe para reduzir o tempo total necessário para completar a atividade de modelagem MBT no processo de teste.

O desenvolvimento do modelo MBT iterativo também está relacionado à geração de testes iterativos. Isso significa que cada vez que alguns aspectos são adicionados ao modelo MBT, a geração de teste também é atualizada. Em essência, a geração de teste constitui uma simulação do modelo MBT, que fornece *feedback* sobre os aspectos comportamentais a serem testados.

3 Critérios de seleção para geração de caso de teste [205 min]

Conceitos-chave

cobertura do modelo, critérios de seleção do teste, Item de cobertura, explosão do caso de teste.

Objetivos de Aprendizagem

3.1 Classificação dos critérios de seleção do teste MBT

FM-3.1.1 (K2) Classificar as várias famílias dos critérios de seleção de testes utilizados para a geração de testes a partir de modelos.

FM-3.1.2 (K3) Gerar casos de teste a partir de um modelo de MBT para alcançar determinados objetivos de teste em um dado contexto.

FM-3.1.3 (K2) Fornecer os exemplos de critérios de cobertura de modelos, de dados, padrões e cenários, e baseados em projetos.

FM-3.1.4 (K2) Reconhecer como os critérios de seleção do teste MBT se relacionam com as técnicas de projeto de teste do *ISTQB® Foundation Level*.

3.2 Aplicando Critérios de Seleção de Teste

FM-3.2.1 (K1) Recordar graus de automação de geração de artefatos de teste.

FM-3.2.2 (K3) Aplicar critérios de seleção de teste dados a um determinado modelo MBT.

FM-3.2.3 (K2) Descrever boas práticas dos critérios de seleção do teste MBT.

3.1 Classificação dos critérios de seleção do teste MBT

A partir do mesmo modelo MBT, vários conjuntos de testes podem ser gerados. O uso de critérios de seleção de teste ajuda o testador a selecionar um subconjunto significativo de casos de teste que melhor se adapta aos objetivos de teste direcionados.

Os geradores de casos de teste desempenham um papel essencial na obtenção de melhorias de eficiência com MBT.

Os critérios de seleção de teste para MBT têm sido amplamente estudados na literatura (ver [Utt07], [Wei09], [Zan11]). Este syllabus apresenta seis famílias de critérios de seleção de teste. Alguns deles se concentram na cobertura de itens específicos, outros suportam aspectos de gerenciamento de projetos, testes de cenários específicos ou aspectos aleatórios.

3.1.1 Critérios de seleção dos testes

Os critérios de seleção de testes baseados na cobertura relacionam a geração de teste com os itens de cobertura do modelo MBT. Os itens de cobertura podem ser:

Requisitos ligados ao modelo MBT

Este critério requer que os elementos do modelo MBT sejam vinculados a requisitos selecionados. A cobertura completa dos requisitos corresponde a um conjunto de casos de teste que cobre completamente um conjunto selecionado de requisitos (cada requisito é coberto por pelo menos um caso de teste).

Elementos do modelo MBT

A cobertura do modelo baseia-se na estrutura interna do modelo MBT. O testador, o gerente de teste ou qualquer outro papel envolvido define itens de cobertura e seleciona o conjunto de casos de teste que cobre uma quantidade desejada desses itens. Os itens de cobertura possíveis podem ser elementos do modelo, tais como:

- Estados, transições e decisões em diagramas de estado (ver "teste de transição de estado" em [ISTQB_FL_SYL]).
- Atividades e *gateways* em modelos de processos de negócios.
- Condições e ações em tabelas de decisão.
- Declarações e condições em modelos textuais.

Critérios de seleção de testes relacionados com dados

Esses critérios se relacionam com técnicas de projeto de teste (ver [ISQB_FL_SYL]), tais como:

- Partição de equivalência.
- Análise do valor de limite.

Eles também incluem heurísticas tais como por pares, ou, mais geralmente, geração combinatória de múltiplos (*n-wise*) casos de teste (para obter mais informações, consulte [ISTQB_ALTA_SYL]).

Outros critérios de seleção de teste incluem:

Aleatória

Este critério de seleção é comparável a um passo-a-passo (mais ou menos aleatório) do modelo. Na seleção aleatória de casos de teste, as diferentes alternativas são todas igualmente prováveis. A seleção de casos de teste estocástico leva em consideração que diferentes alternativas podem ter probabilidades diferentes.

Baseado em cenários e padrões

A seleção do caso de teste é baseada em cenários ou padrões predefinidos. Um cenário pode ser um caso de uso explicitamente definido (veja também "teste de caso de uso" em [ISTQB_FL_SYL]) ou uma história de usuário explicitamente definida (consulte [ISTQB_FLAT_SYL]). Um padrão é um cenário parcialmente definido, que pode ser aplicado ao modelo MBT para produzir vários ou múltiplos testes.

Projetos

A seleção de casos de teste baseada em projetos é baseada em informações adicionais relacionadas ao projeto, que foram adicionadas ao modelo para suportar o gerenciamento de testes, ou para atingir objetivos de teste específicos para o projeto. As informações relacionadas ao projeto incluem riscos, prioridades, equipamentos necessários de teste ou qualquer outro aspecto de interesse no projeto específico. Selecionar todos os casos de teste que requerem equipamento específico de teste corresponde à aplicação de um critério de seleção orientado por projeto.

As ferramentas MBT suportam pelo menos um, mas geralmente não todos os critérios de seleção de teste mencionados acima.

3.1.2 Seleção de casos de teste na prática

Na prática, o testador deve considerar a combinação de vários critérios de seleção de teste para obter um subconjunto de casos de teste que melhor se adapte aos objetivos de teste direcionados. Exemplos para essas combinações são cobertura de requisitos e cobertura de modelos ou critérios de seleção de teste de cenário/padrão e dados.

Para além dos objetivos visados, a aplicação dos critérios de seleção dos testes também depende:

- Dos mecanismos fornecidos pela ferramenta MBT.
- Do desenho do modelo.
- Da experiência do testador com os critérios.

É possível que o modelo MBT esteja formalmente correto, mas produz uma explosão de caso de teste devido a algoritmos de geração específicos da ferramenta.

3.1.3 Exemplos de critérios de seleção de testes

Alguns exemplos de critérios de seleção de testes em relação a algumas linguagens de modelagem são (ver [Utt07] - capítulo 4):

Em diagramas de atividades ou modelos de processos de negócios:

- Cobertura da atividade (100% = cobertura de cada atividade no diagrama).
- Cobertura decisão/*gateway* (100% = cobertura de cada ponto de decisão no diagrama).
- Cobertura do caminho (100% = cobertura de todos os cenários de negócios no diagrama), com ou sem *loops*.

Em diagramas de estado:

- Cobertura de transição de estado (100% = cobertura de cada estado ou transição).
- Cobertura de par de transições (100% = cobertura de cada par consecutivo de transições no diagrama).
- Cobertura de caminho (100% = cobertura de todos os caminhos, normalmente sem *loops*, no diagrama).

Nas tabelas de decisão:

- Toda condição.
- Toda ação.
- Toda regra.

Nos domínios de dados definidos na parte estrutural do modelo:

- Cobertura de partição de equivalência (p. ex., com um representante por classe).
- Cobertura de valores limites (p. ex., considerando limites em intervalos de dados numéricos).
- Teste de pares em domínios definidos (garantindo que todas as possíveis combinações discretas de cada par de parâmetros de entrada são produzidas).

Em um modelo textual:

- Cobertura de declaração (100% = cada declaração executável está coberta).
- Cobertura de decisão (100% = cada decisão é coberta).
- Cobertura da condição de decisão (100% = todos os resultados da condição e todos os resultados da decisão foram exercidos pelo conjunto de testes).
- Cobertura de múltiplas condições (todas as combinações de condições em cada decisão são cobertas - leva a um elevado número de casos de teste gerados).

3.1.4 Relação com as técnicas de modelagem de testes do nível de fundamental

O MBT suporta as técnicas de teste do *ISTQB® Foundation Level*, tais como testes de transição de estado, particionamento de equivalência, análise de valores limite, testes de tabelas de decisão e testes de casos de uso. O próprio modelo MBT pode conter itens de cobertura dessas técnicas de concepção de teste (p. ex., partições de equivalência ou valores de limite descritos no modelo MBT). Alternativamente, o modelo MBT pode ser combinado com outras representações gráficas ou textuais contendo esses itens (p. ex., tabelas de decisão ou diagramas adicionais).

3.2 Aplicando critérios de seleção de teste

3.2.1 Grau de automação na geração de teste

O MBT é geralmente associado à geração automatizada de casos de teste, mas o processo de geração não é necessariamente baseado em ferramentas. Os métodos a seguir podem ser usados para geração de casos de teste:

- **Geração de teste manual:** Derivando casos de teste manualmente a partir do modelo MBT, seguindo os caminhos e escrevendo os casos de teste correspondentes "manualmente". No entanto, esta abordagem MBT tem um baixo nível de maturidade e não fornece os benefícios em relação à eficiência e eficácia da geração automatizada de casos de teste mencionados no capítulo 1.1.
- **Geração de teste automatizado:** Artefatos de saída, como casos de teste, são gerados automaticamente por uma ferramenta MBT e podem ser usados como estão sem mais pós-processamento.
- **Geração de teste semi-automatizado:** Soluções intermediárias também existem onde uma ferramenta é usada, mas etapas manuais são necessárias no meio ou no final, por exemplo para selecionar casos de teste específicos.

3.2.2 Prós e contras de critérios específicos de seleção de testes

Cada critério de seleção do teste tem vantagens e desvantagens. A escolha depende dos objetivos do teste. Estes são exemplos de prós e contras para vários critérios de seleção de teste:

Cobertura de requisitos

Pró: Frequentemente obrigatório devido a requisitos regulatórios.

Contra: É necessária uma definição precisa (p. ex., se uma exigência está ligada a vários comportamentos no modelo, a cobertura de requisitos é alcançada ao cobrir todos os comportamentos ou apenas um).

Cobertura do modelo

Pró: Útil para entender onde exatamente um modelo não poderia ser coberto e para se ter uma ideia de completude da cobertura do teste em termos do modelo.

Contra: Alguns critérios de seleção de testes estruturais podem produzir a explosão de casos de testes durante a geração de teste (p. ex., toda a cobertura de trajetória em um diagrama de atividade).

Critérios de seleção de testes relacionados com dados

Pró: Frequentemente obrigatório para cobrir partições de equivalência dos domínios.

Contra: Pode ser fortemente combinatória.

Aleatória

Pró: Útil para selecionar casos de teste inesperados.

Contra: Algoritmos aleatórios podem gerar testes longos que não têm significado comercial.

Baseados em cenários / padrões

Pró: Suporta a seleção de casos de uso (p. ex., para teste de regressão).

Contra: Exige esforço extra para definir e manter cenários e padrões.

Projetos

Pró: Útil para a gestão de testes.

Contra: Requer esforço extra para vincular informações específicas ao modelo.

3.2.3 Boas práticas de seleção de testes MBT

Muitas vezes, um critério por si só não é suficiente para cobrir todos os aspectos de teste necessários para alcançar o objetivo do teste alvo. Nesse caso, o testador tem de ter a necessidade de combinar vários critérios.

Existem duas abordagens para combinar critérios:

- **Composição dos critérios:** O conjunto de testes gerado contém apenas os casos de teste que cumprem todos os critérios que foram aplicados (interseção).
- **Adição de critérios:** O conjunto de testes gerado contém todos os casos de teste que satisfazem pelo menos um critério (união).

Os critérios de seleção podem influenciar o conceito de modelagem MBT. Além disso, deve ser possível reproduzir a seleção, ou seja, obter o mesmo conjunto de casos de teste do mesmo modelo ao aplicar os mesmos critérios. Portanto, é essencial planejar as atividades de seleção de teste e documentar as escolhas e as razões subjacentes.

Para alguns critérios de cobertura do modelo, existem vários conjuntos de caminhos que cumprem a cobertura necessária. Pequenas mudanças no modelo MBT podem levar a uma seleção completamente diferente, um fato que requer uma nova maneira de pensar. O modelo é o mestre, os casos de teste gerados são artefatos derivados.

Os critérios de seleção do teste também são uma maneira no MBT para dominar a explosão do caso de teste; usando um único critério ou uma combinação de critérios, o testador pode ajustar precisamente a geração de teste para satisfazer os objetivos do teste e evitar a explosão do caso de teste.

4 Implementação e execução do teste MBT [120 min]

Conceitos-chave

MBT *offline*, MBT *online*, camada de adaptação de teste.

Objetivos de aprendizagem

4.1 Especificações da implementação e execução do teste MBT

FM-4.1.1 (K2) Explicar a diferença entre casos de teste abstratos e concretos no contexto MBT.

FM-4.1.2 (K2) Explicar os diferentes tipos de execução de teste no contexto MBT.

FM-4.1.3 (K3) Executar atualizações de um modelo MBT e geração de testes causados por alterações nos requisitos, objetos de teste ou objetivos de teste.

4.2 Atividades de adaptação aos testes em MBT

FM-4.2.1 (K2) Explicar que tipo de adaptação do teste pode ser necessária para a execução do teste MBT.

4.1 Especificações da implementação e execução do teste MBT

Uma vez que o conjunto de testes foi gerado, o próximo passo é a execução dos casos de teste. A execução do teste pode ser manual ou pode ser automatizada usando um ambiente de execução de teste que fornece facilidades para executar automaticamente os testes e registrar os resultados do teste. Em ambos os casos, deve haver uma conexão entre o modelo MBT e os artefatos de execução de teste.

4.1.1 Casos de teste abstratos e concretos no contexto MBT

A geração do teste MBT pode produzir casos de teste abstratos (casos de teste de alto nível) ou casos de teste concretos (casos de teste de baixo nível):

- Os casos de teste abstratos são casos de teste sem valores concretos (nível de implementação) para dados de entrada e resultados esperados. Eles também não especificam as etapas do teste em detalhes.
- Os casos de teste concretos são casos de teste com valores concretos (nível de implementação) para dados de entrada e resultados esperados e com descrições muito detalhadas das etapas do teste.

Os modelos MBT podem conter diferentes níveis de abstração abordando diferentes artefatos de saída a serem gerados, e abordando as diferentes stakeholders. Por exemplo:

- Os casos de teste abstratos podem ser direcionados para revisão por analistas de negócios.
- Os casos de teste concretos podem ser executados diretamente por testadores.

Os casos de teste abstratos gerados pelo MBT fornecem informações sobre condições de teste, dados de entrada e resultados esperados sem valores concretos. A informação do teste é expressa em um nível abstrato, por exemplo:

- Pode definir os procedimentos de teste com sequências de ações de teste de alto nível em vez das ações de teste detalhadas e completamente definidas.
- Pode fornecer partições de equivalência em vez de representantes concretos das partições.

Passando dos casos de teste abstratos para casos de teste concretos prontos para execução (manual ou automaticamente), é necessário complementar os casos de teste abstratos com o seguinte:

- Ações de teste completamente definidas.
- Valores concretos e completos de dados de entrada.
- Valores concretos para resultados esperados

Esta informação complementar pode ser definida no modelo MBT (p. ex., como documentação dos artefatos do modelo, como ações/verificações e dados) ou fora do modelo MBT (p. ex., usando uma tabela de dados para mapear valores de dados abstratos e concretos).

4.1.2 Diferentes tipos de execução de teste

Os casos de teste MBT gerados podem ser executados de forma manual ou automatizada.

Para a execução de teste manual, os testadores executam casos de teste que foram gerados pela ferramenta MBT. Esses casos de teste devem ser gerados em um formato utilizável para a execução de teste manual. Também pode ser benéfico ser capaz de exportar os casos para uma ferramenta de gerenciamento de teste. A execução do teste e o gerenciamento de defeitos são então realizados de acordo com o processo fundamental de teste do ISTQB.

Para a execução do teste automatizado, os casos de teste devem ser gerados de uma forma que seja executável. Há três abordagens principais que são usadas para gerenciar a execução automatizada a partir de casos de teste abstratos (ver [Utt07] - capítulo 8):

- **Abordagem de adaptação:** Com esta abordagem, o código da camada de adaptação de teste é escrito para preencher a lacuna de abstração. Este código é essencialmente um invólucro em torno do objeto de teste (semelhante à abordagem de teste baseada em palavra-chave)
- **Abordagem de transformação:** Com esta abordagem, os casos de teste gerados pelo MBT são convertidos diretamente em scripts de teste automatizados (nenhuma camada de adaptação de teste é necessária)
- **Abordagem mista:** Esta é uma combinação das duas abordagens anteriormente mencionadas nas quais os casos de teste abstratos são transformados em casos de teste concretos, que são envolvidos com o código da camada de adaptação em torno do objeto de teste.

Uma vez gerados, os scripts de teste automatizados são executados por uma ferramenta de execução de teste. As ferramentas MBT podem converter casos de teste gerados na linguagem de teste de automação de scripts da ferramenta de execução de teste. A ferramenta MBT também pode publicar os scripts de teste automatizados gerados em uma ferramenta de gerenciamento de teste.

Existem dois métodos para execução de teste automatizado no contexto MBT:

- **MBT offline:** os scripts de teste automatizados são gerados primeiro (incluindo os resultados esperados) e executados posteriormente.
- **MBT online:** (também chamado "*on-the-fly*") a geração de teste e sua execução são realizadas simultaneamente.

Portanto, cada etapa de teste é gerada após a execução da etapa anterior durante o teste.

O resultado da execução pode influenciar o caminho tomado através do modelo.

4.1.3 O impacto das mudanças nos artefatos do MBT

As mudanças são inevitáveis em um projeto de *software*. As seguintes mudanças podem ocorrer e devem ser esperadas:

- a) Mudança nos requisitos, no objeto de teste ou seu ambiente, que podem afetar:
 - O modelo MBT:
 - Ação;
 - Condições/resultados esperados;
 - Dados.
 - Seleção de critérios do teste MBT;
 - A especificação da camada de adaptação se existir (e o código se existir).

- b) Mudança na interface do objeto de teste, mas não nos requisitos funcionais (p. ex., uma pequena mudança na interface gráfica do usuário sem impacto no comportamento funcional), que impacta:
- Apenas a especificação da camada de adaptação (e o código se existir).
- c) Mudança no objetivo do teste ou condições de teste que podem afetar:
- Modelo MBT;
 - Seleção de critérios de teste MBT;
 - A especificação da camada de adaptação se existir (e o código se existir).

O gerenciamento de mudanças em artefatos MBT deve ser baseado em um processo que inclui análise de impacto, exploração de opções de mudança, aplicação de mudanças em artefatos e atividades de revisão.

4.1.4 Atividades de Adaptação ao Teste em MBT

No caso de execução de teste manual, a adaptação do teste está relacionada com a documentação dos testes gerados para preencher a lacuna entre as abstrações feitas no modelo MBT e as interfaces concretas e dados de teste do sistema em teste. Por exemplo, valores concretos de dados podem ser fornecidos como representantes dos valores limite específicos. Este processo de adaptação fornece scripts de teste manuais que estão completos e suficientemente documentados para que possam ser usados diretamente para testes manuais.

No caso da execução do teste automatizado, a adaptação do teste é o processo de integrar os artefatos gerados a partir de um modelo MBT na estrutura de execução com base na especificação da camada de adaptação. Este processo suporta as práticas de testes orientados por palavras-chave (*keyword-driven*) ou testes orientados por dados (*data-driven*).

No que diz respeito ao teste baseado em palavras-chave, as palavras-chave são definidas no modelo MBT e utilizadas nos casos de teste gerados. Para obter scripts de teste totalmente automatizados, são necessários os seguintes passos e atividades:

- Exportar os casos de teste como scripts na linguagem da ferramenta de execução de teste. Isso pode ser feito manual ou automaticamente com um exportador (pode ser fornecido pela ferramenta MBT);
- Implementar as palavras-chave utilizando a especificação da camada de adaptação na linguagem da ferramenta de execução de teste. O testador responsável pela automação de teste ou o desenvolvedor do objeto de teste pode fazer isso.

No que se refere aos testes orientados por dados, o modelo MBT descreve os dados de entrada abstratos e os resultados esperados (p. ex., com base na partição de equivalência). Os casos de teste gerados ou scripts referem-se a estes dados de entrada abstratos e resultados esperados.

Para obter scripts de teste totalmente automatizados, são necessários os seguintes passos e atividades:

- Fornecer os dados de entrada concretos e os resultados esperados necessários na especificação da camada de adaptação. Esses dados podem ser armazenados em uma tabela ou em uma planilha;

- Vincular os dados de teste formalizados no modelo MBT aos dados concretos de teste na ferramenta de execução de teste.

Cada script de teste assume pré-condições iniciais específicas. Para poder encadear a execução dos scripts de teste automatizados, é necessário garantir que as pré-condições sejam definidas corretamente antes de cada execução do script de teste. Isso pode ser feito das seguintes maneiras:

- Ao fornecer pós-condições dentro dos scripts de teste (o que nem sempre é possível) para ser usado como pré-condições para o próximo;
- Ao configurar a pré-condição no início de cada script de teste.

A adaptação do teste deve ser preparada durante as atividades de modelagem do MBT, por exemplo, desenvolvendo a especificação de adaptação do teste ao mesmo tempo em que os elementos do modelo são desenvolvidos.

5 Avaliação e implantação de uma abordagem MBT [60 min]

Conceitos-chave

Nenhuma

Objetivos de Aprendizagem

5.1. Avaliar uma implantação MBT

FM-5.1.1 (K2) Descrever os fatores de ROI para a introdução do MBT.

FM-5.1.2 (K2) Explicar como os objetivos do projeto estão relacionados às características da abordagem MBT.

FM-5.1.3 (K1) Recuperar métricas selecionadas e principais indicadores de performance para medir o progresso e os resultados das atividades de MBT.

5.2. Gerenciar e monitorar a implantação de uma abordagem MBT

FM-5.2.1 (K1) Recuperar boas práticas para gerenciamento de testes, gerenciamento de mudanças e trabalho colaborativo ao implantar MBT.

FM-5.2.2 (K1) Fatores de custo de recarga do MBT.

FM-5.2.3 (K2) Dar exemplos da integração da ferramenta MBT com ferramentas de gerenciamento de configuração, gerenciamento de requisitos, gerenciamento de teste e automação de teste.

5.1 Avaliar uma implantação MBT

A implantação de uma abordagem MBT em uma organização geralmente segue um processo clássico de adoção de produtos:

- 1) **Conscientização:** estabelecer os objetivos de melhoria no processo de teste e identificação de MBT como uma possível tecnologia para abordar todos ou alguns desses objetivos. Essa identificação de possíveis melhorias é um fator importante para a motivação para mudar a maneira de trabalhar nas equipes de teste.
- 2) **Interesse:** aprender mais sobre MBT.
- 3) **Avaliação:** analisar os principais princípios MBT e abordagens MBT existentes no que diz respeito à sua aplicabilidade dado o contexto do projeto.
- 4) **Utilização de testes:** definir os indicadores-chave de performance (KPI) e criar um projeto-piloto para avaliar melhorias.
- 5) **Adoção:** liderar e reforçar a mudança com melhorias de competência e mudança de comportamento na organização.

Portanto, a estimativa do ROI (Retorno de Investimento) e a capacidade de avaliar o impacto do MBT na performance do projeto de teste (com base em KPIs) faz parte do ciclo de adoção.

5.1.1 Fatores de ROI para implantação ao MBT

O MBT gera custos tanto durante a implantação como na fase de operação. Para uma implantação bem-sucedida do MBT, os custos têm de ser contrabalançados pelos benefícios em relação a:

- Balanço financeiro líquido tendo em conta o processo de teste completo em uma escala de longo prazo;
- Qualidade do projeto de teste e impacto positivo no processo completo de desenvolvimento.

Custo, economia e melhorias no processo de teste e na qualidade do teste devem ser levados em conta.

Custos (ver capítulo 5.2.2):

- Introdução ao MBT;
- Custo usando MBT no processo de teste.

Economias:

- *A validação precoce de requisitos, evitando assim correções de bugs dispendiosas em fases de desenvolvimento posteriores:* a abordagem *top-down* oferecida pelo MBT permite a modelagem do MBT nas etapas do projeto, quando os requisitos ainda não foram corrigidos ou apenas definidos de forma aproximada, permitindo também a validação de requisitos na maturidade dos requisitos atuais;
- *Esforço reduzido na modelagem do teste pela reutilização dos artefatos do projeto de teste, e para evitar redundâncias:* em um modelo MBT, todos os caminhos possíveis são integrados sem redundância em um artefato, em comparação com as redundantes suítes de sequências de teste implementadas com teste manual ou *scripts* de teste automatizados;

- *Esforço reduzido na implementação do teste pela geração automatizada de artefatos de teste a partir dos modelos MBT:* uma vez desenvolvido o modelo MBT, os casos de teste são gerados automaticamente reduzindo o esforço para as atividades de manutenção;
- *Identificação antecipada de defeitos ou falhas e melhoria no tempo de implantação em produção:* isto é fomentado pela capacidade de MBT para começar cedo no projeto de teste e a eficiência avançada fornecida por MBT;
- *Suporte ao gerenciamento de teste:* usando as diferentes estratégias de geração de teste MBT e mecanismos de filtragem, o gerente de teste pode selecionar o conjunto ideal de casos de teste que melhor se ajustam ao objetivo do teste;
- *Redução potencial do número de casos de teste definidos usando MBT:* (em comparação com técnicas de design de caso de teste sem MBT) devido algoritmos otimizados, reduzindo assim o número de casos de teste que serão executados. Os casos de teste podem ser selecionados e reduzidos em número sistematicamente com base em estratégias de geração uniformes e sistemáticas.

Benefícios para a qualidade do teste:

- Melhoria dos métodos de modelagem dos testes e consistência da modelagem do teste;
- Maior conhecimento sobre a cobertura do teste e, portanto, sobre a qualidade do projeto do teste;
- Suporte de gerenciamento de teste usando saídas MBT incluindo priorização e garantia de qualidade do projeto de teste;
- Melhorias na rastreabilidade através da melhoria do conteúdo e da organização do desenho do teste.

5.1.2 Objetivos organizacionais e sua relação com as características do MBT

O MBT pode melhorar o processo de teste em relação à qualidade do teste, redução do esforço e comunicação. A organização deve priorizar as melhorias necessárias. O escopo das melhorias e seu impacto qualitativo e quantitativo na organização depende muito das características do método MBT. Assim, a abordagem MBT deve ser definida com base no escopo de melhorias priorizadas.

A tabela a seguir mostra alguns exemplos de objetivos organizacionais e como o MBT pode ser usado para atendê-los.

Objetivos Organizacionais	MBT Melhoria Foco	Alcançado por
Aprimoramento da Qualidade dos testes	<ul style="list-style-type: none"> • Métodos do projeto do teste para maior cobertura do teste. • Rastreabilidade (permitindo métricas de cobertura e melhor capacidade para análise de impacto). 	<ul style="list-style-type: none"> • Modelos separados para o desenvolvimento e atividades MBT (permitindo a mentalidade do testador e incentivando a independência) • Nível de abstração bem definido com base no plano de teste • Alto grau de automação de processos, incluindo a geração de artefatos de teste e a execução de testes para reduzir erros humanos

Objetivos Organizacionais	MBT Melhoria Foco	Alcançado por
	<ul style="list-style-type: none"> Automatização de processos para evitar erros humanos. 	<ul style="list-style-type: none"> Critérios de seleção de testes bem definidos adaptáveis a objetivos específicos de teste
Redução de esforço	<ul style="list-style-type: none"> Automação de processos Rastreabilidade (suporte à automação de processos). 	<ul style="list-style-type: none"> Modelos MBT compartilhados (reutilização de modelos MBT) Alto grau de automatização de processos e artefatos de teste gerados automaticamente Critérios de seleção de testes bem definidos para gerar o conjunto mais eficiente de casos de teste a serem executados
Comunicação aperfeiçoada	<ul style="list-style-type: none"> Abordagem MBT adequada, compreensível e utilizável, refletindo o nível de abstração do pensamento dos stakeholders envolvidos 	<ul style="list-style-type: none"> Nível de abstração adequado para todos os stakeholders envolvidos (p. ex., alto nível e orientado para os analistas de negócios ou detalhado e testado para testadores)

Uma combinação de objetivos organizacionais pode levar a requisitos contraditórios para a abordagem MBT. Isso pode ser resolvido desenvolvendo diferentes modelos MBT para diferentes objetivos de teste (ver capítulo 2.1.3).

5.1.3 Métricas e principais indicadores de performance

A introdução do MBT em uma organização deve ser baseada em objetivos, métricas e KPIs claros que podem ser usados para medir o progresso e os resultados das atividades MBT.

Métricas e KPIs possíveis de serem monitoradas incluem:

- O número de requisitos gerenciados e rastreados no modelo MBT e a cobertura de requisitos (porcentagem) por casos de teste gerados;
- O tamanho e a complexidade do modelo MBT;
- O número de casos ou *scripts* de teste gerados e o número de casos ou *scripts* de teste gerados por pessoa/dia;
- O número de defeitos encontrados nos requisitos durante as atividades de modelagem MBT;
- O nível de reutilização dos elementos do modelo MBT de um projeto para outro;
- O nível de utilização dos modelos MBT pelos stakeholders do projeto (negócios, desenvolvimento, teste);
- A porcentagem de ganho de eficiência em relação às abordagens anteriores para o modelo do teste em termos de produtividade (testes mais baratos);
- A porcentagem de ganho de eficiência em relação às abordagens anteriores para a concepção do teste em termos de detecção de defeitos (melhores testes).

Definir e monitorar métricas e KPIs é parte das melhores práticas de gerenciamento de projetos ao implementar uma abordagem MBT.

5.2 Gerenciar e monitorar a implantação de uma abordagem MBT

5.2.1 Boas práticas na implantação do MBT

O processo fundamental de teste baseado em MBT, juntamente com seus artefatos e ferramentas, deve ser estreitamente integrado ao processo de desenvolvimento, ao processo de teste e à cadeia de ferramentas existentes. Isso também é importante para a integração na cadeia de ferramentas do ciclo de vida do gerenciamento de aplicativos, por exemplo, a integração do MBT com os processos e ferramental da engenharia de requisitos.

Uma integração perfeita é um fator chave para o sucesso na introdução MBT. Boas práticas aqui incluem:

- *Gerenciamento de configuração de todos os artefatos MBT*, incluindo:
 - Base de teste;
 - Modelos MBT e critérios de seleção de testes;
 - Casos de teste e scripts de teste;
 - Especificação e código da camada de adaptação.

Em um processo de desenvolvimento, artefatos, como os de lançamento ou implantação, são versionados. Isso é obrigatório para estabelecer um processo de desenvolvimento e implantação de trabalho. Para integrar e relacionar artefatos MBT em tal processo, o gerenciamento de configuração de artefatos MBT também é obrigatório.

- *Integração do processo de geração de teste MBT com integração contínua*. Fazer um autoteste de compilação é um dos valores-chave da integração contínua. Uma vez que o código é construído, o servidor de integração contínua chama as ferramentas de teste para verificar o novo conteúdo. Não só ferramentas de teste de unidade, mas também ferramentas MBT devem estar integradas aqui, especialmente quando MBT é usado para testes de regressão contínua.
- *Integração com engenharia de requisitos e práticas de gerenciamento de backlog*. Requisitos e itens de *backlog* (recursos planejados do produto) devem ser testados antes de serem concluídos (de acordo com a definição de realizado [ISTQB_FLAT_SYL]). Por exemplo, em um processo ágil, o processo de teste deve ser refletido no processo de gerenciamento de *backlog*. Se o MBT é usado para o teste de itens de *backlog*, os artefatos MBT correspondentes devem ser rastreáveis na ferramenta de gerenciamento de *backlog*, isto é, "quais casos de teste, de qual modelo versionados, com qual objetivo de teste, foram usados para testar este item".

5.2.2 Fatores de custo de MBT

As tabelas a seguir relacionam os custos iniciais e os custos correntes do MBT com as atividades de teste do processo fundamental de teste do ISTQB.

Custos iniciais (para a organização e para o projeto):

Custos iniciais para a organização	Custos iniciais para o projeto
<ul style="list-style-type: none"> • Verificar os recursos e conhecimentos existentes para a introdução do MBT; • Avaliação de abordagens e ferramentas MBT; • Definição e implementação de métodos e processos MBT; • Integração com gerenciamento de requisitos, gerenciamento de teste, integração contínua; • Automatização e integração dos relatórios MBT; • Estabelecimento de meios para arquivamento de artefatos MBT; • Criando diretrizes gerais de modelagem e processo do MBT; • Treinamento e coaching MBT; • Licenciamento de ferramentas MBT. 	<ul style="list-style-type: none"> • Criação de modelos específicos de MBT e diretrizes de processo; • Criando o modelo MBT inicial; • Transformar ativos (p. ex., de casos de teste de texto para modelos MBT); • Migração de modelos MBT.

Custos atuais:

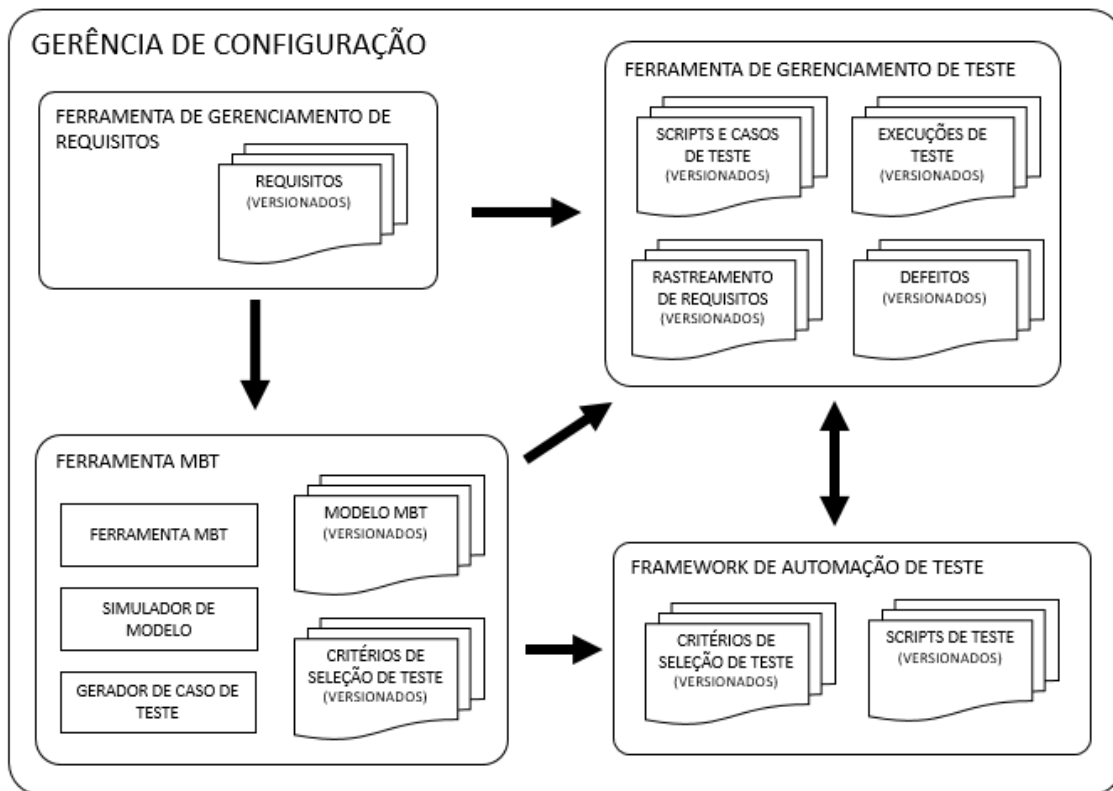
Atividades de Teste	Custos atuais
Geral	<ul style="list-style-type: none"> • Licenciamento de ferramentas (dependendo do modelo de licença) e custos de manutenção. • Treinamento e coaching de novos membros da equipe.
Planejamento e Controle	<ul style="list-style-type: none"> • Analisando a base de testes com relação à testabilidade MBT. • Planejamento do desenvolvimento/enriquecimento/derivação de modelos MBT. • Verificação contínua da qualidade do modelo MBT.
Análise e Design	<ul style="list-style-type: none"> • Modelagem MBT. • Refatoração de modelos MBT. • Validação e verificação de modelos.
Implementação e Execução	<ul style="list-style-type: none"> • Escolhendo critérios de seleção de teste adequados. • Geração de casos de teste executáveis. • Desenvolver a camada de adaptação de teste (no caso de execução automática de teste). • Executar casos de teste (manual ou automaticamente).
Avaliação de Critérios de Saída e Relatórios	<ul style="list-style-type: none"> • Garantia de rastreabilidade de defeitos. • Documentação dos critérios de conclusão do teste. • Combinação de avaliação MBT e outras avaliações de teste em um relatório comum.
Encerramento	<ul style="list-style-type: none"> • Arquivamento de artefatos MBT. • Documentar o conhecimento adquirido em relação ao MBT. • Transição de artefatos e processos MBT para manutenção.

5.2.3 Integração da ferramenta MBT

A introdução de uma modelagem MBT em uma organização segue os mesmos princípios que se aplicam à introdução de qualquer outra ferramenta de teste, consulte [ISTQB_FL_SYL], capítulo 6.3. Uma vez que MBT normalmente não é tão prevalente como abordagens de teste tradicionais, a avaliação e introdução da ferramenta deve ser particularmente sólida e não deve ser subestimada.

Um aspecto importante desta avaliação está relacionado com a integração da ferramenta MBT com o gerenciamento de configuração, gerenciamento de requisitos, gerenciamento de teste e ferramentas de automação de teste.

Uma cadeia de ferramentas de teste típica que encaixa MBT é ilustrada no diagrama abaixo.



Esta cadeia de ferramentas típicas suportam as principais atividades do processo fundamental de teste do MBT, incluindo:

- Desenvolvimento, revisão e validação do modelo MBT iterativo utilizando a ferramenta de modelagem, o simulador de modelo e a rastreabilidade com os requisitos;
- Geração de teste aplicando critérios de seleção de teste ao modelo MBT suportado pelo gerador de casos de teste;
- Casos de teste e *scripts* de teste gerados, bem como *links* de rastreabilidade que podem ser exportados para a ferramenta de gerenciamento de teste e a estrutura de automação de teste (no caso de automação de execução de teste);
- Gerenciamento de configuração para o *testware* MBT, como o modelo MBT e a camada de adaptação.

Abreviaturas

BPMN Business Process Modeling Notation

BSTQB Brazilian Software Testing Qualifications Board

CTFL Certified Tester Foundation Level

ISTQB International Software Testing Qualifications Board

KPI Key Performance Indicator

MBT Model-Based Testing

ROI Retorno sobre o Investimento (Return On Investment)

UML Unified Modeling Language

Marcas Registradas

UML® Marca registrada pertencente à Object Management Group, Inc.

Referências

Padrões

[ISO25000] ISO/IEC 25000:2005, *Software Engineering - Software Product Quality 4 - Requirements and Evaluation (SQuaRE)*

[ETSI_MBT] ETSI ES 202 951, *Methods for Testing and Specification (MTS) - Model-Based Testing (MBT) - Requirements for Modeling Notations, Version 1.1.1 (2011-07)*

Documentos BSTQB

[ISTQB_FLAT_SYL] Syllabus CTFL Agile Tester, Version 2014.

[ISTQB_ALTA_SYL] Syllabus CTAL Test Analyst, Version 2012.

[ISTQB_FL_SYL] Syllabus CTFL, Version 2011.

[ISTQB_GLOSSARY] Glossario de Termos de Teste de software, Version 3.0, 2015.

[ISTQB_FLMBT_OVW] Overview CTFL Model-Based Tester, Version 2015.

Referências neste Syllabus

[Utt07] Mark Utting and Bruno Legeard, "Practical Model-Based Testing - A tools approach," Morgan&Kauffmann, 2007.

[Wei09] Stephan Weißleder. "Test Models and Coverage Criteria for Automatic Model-Based Test Generation with UML State Machines," PhD Thesis, Humboldt-Universität zu Berlin, 12/2009.

[Zan11a] Justyna Zander, Ina Schieferdecker and Pieter J. Mosterman, "A Taxonomy of Model-Based Testing for Embedded Systems from Multiple Industry Domains," In *Model-based testing for embedded systems*. CRC Press.

Outras Referências

[Bak08] Paul Baker, Zhen Ru Dai, Jens Grabowski, Øystein Haugen, Ina Schieferdecker and Clay Williams, "Model-Driven Testing," Springer, 2008.

[Jac07] Jonathan Jacky, Margus Veanes, Colin Campbell and Wolfram Schulte, "Model-Based *Software* Testing and Analysis with C#," Cambridge University Press, 2007.

[Sch12] Ina Schieferdecker: Model-Based Testing. *IEEE Software* 29(1): 14-18, 2012.

[Utt12] Mark Utting, Alexander Pretschner and Bruno Legeard, "A Taxonomy of Model-Based Testing Approaches," *Softw. Test. Verif. Reliab.* 22 (5), 297-312, 2012.

[Zan11b] Justyna Zander (Editor), Ina Schieferdecker (Editor) and Pieter J. Mosterman (Editor), "ModelBased Testing for Embedded Systems," CRC Press, 2011.

Apêndice A – Linguagem de Modelagem Simples

Para o propósito dos objetivos de aprendizado do nível K3, duas linguagens de modelagem gráfica simples são discutidas aqui:

- A primeira é uma subparte de diagramas de atividade UML
- A segunda é uma subparte de máquinas de estado UML

Nos dois capítulos seguintes, essas partes da UML, que podem ser usadas para praticar a modelagem MBT, são definidas por exemplos.

Uma linguagem de modelagem gráfica simples para fluxos de trabalho

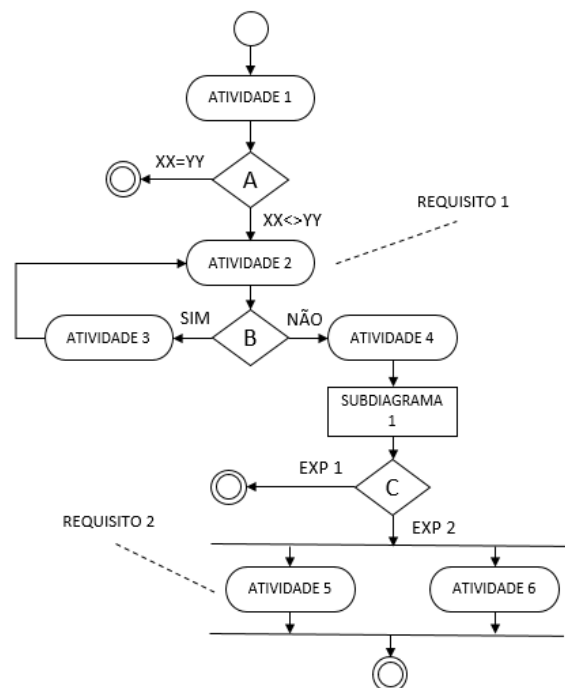
Esta linguagem de modelagem pode ser usada para construir modelos MBT representando fluxos de trabalho ou diagramas de atividade. A ideia é modelar um fluxo de atividades controladas por decisões intermediárias. Os modelos MBT desenvolvidos com esta linguagem podem ser usados para gerar casos de teste com base em critérios de seleção de teste (ver capítulo 3).

Esta linguagem de modelagem gráfica é uma parte do diagrama de atividades UML, sendo composto pelos seguintes elementos:

- Um círculo preto representando o estado inicial do fluxo de trabalho e um círculo preto circundado que representa o estado final.
- Retângulos arredondados representando as ações.
- Losangos representando “nós de decisão” e de mesclagem com etiquetas (texto).
- Setas que representam fluxos, possivelmente com expressões como texto ou expressões lógicas (incluindo operadores aritméticos e booleanos).
- Barras que representam o início ou o fim de atividades paralelas simultâneas o processo prossegue após todas as atividades paralelas chegarem à junção final.
- Identificadores de requisitos vinculados a atividades usando linhas tracejadas.
- Sub diagramas indicados por retângulos.

Tal descrição de uma linguagem de modelagem é bastante informal e simples, mas contém informações suficientes para ser usado para criar modelos MBT no contexto deste syllabus.

A figura ao lado representa um exemplo abstrato de um modelo MBT desenvolvido com esta linguagem de modelagem. Ele mostra o comportamento que sempre começa com a Atividade 1. O processo termina se $xx=yy$ senão a Atividade 2 é executada. As Atividades 2 e 3 são executadas repetidamente até que o



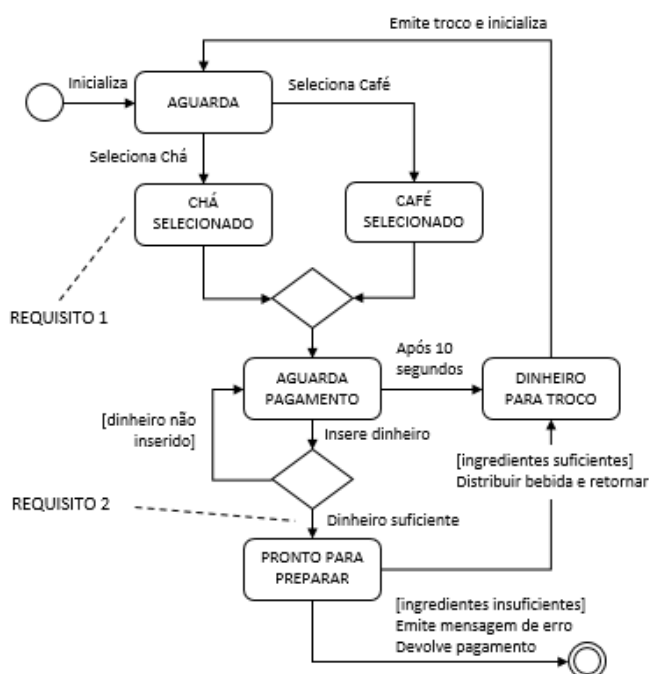
resultado da decisão B seja "Não". Neste caso, a Atividade 4 e o comportamento do Sub diagrama 1 são executados. O comportamento para se o resultado da Decisão C é igual a Exp 1, senão as Atividades 5 e 6 são executadas em paralelo, então o comportamento para. A notação mostra que a Atividade 2 está relacionada com o Requisito 1 e a Atividade 5 está relacionada com o Requisito 2.

Uma linguagem de modelagem gráfica simples para diagramas de transição de estados

Esta linguagem de modelagem gráfica é uma parte de máquinas de estado UML. É composto dos seguintes elementos:

- Um círculo preto representando o início (estado inicial) do fluxo de trabalho e um círculo preto circundado que representa o fim (estado final).
- Retângulos arredondados rerepresentando estados.
- Setas representando transições com anotação textual "evento [condição] / ação" (o gatilho se refere a qualquer evento, o guarda contendo operadores aritméticos e booleanos que formam condições, o efeito que chama qualquer ação).
- Diamantes, representando nós de decisão e de mesclagem.
- Identificadores de requisitos ligando a qualquer tipo de elemento usando linhas tracejadas.
- Sub diagramas são indicados por retângulos.

Tal descrição de uma linguagem de modelagem é bastante informal e simples, mas contém informações suficientes para ser usado para criar modelos MBT no contexto deste syllabus.



A figura ao lado (Figura 3) representa um exemplo de um modelo MBT desenvolvido com esta linguagem de modelagem. Mostra o comportamento de um distribuidor de bebidas contendo chá e café. Se o equipamento é ativado, a quantidade depositada é inicializada e está aguardando entradas do usuário. Estas entradas do utilizador podem ser "selecionar chá" ou "selecionar café" indicando a bebida desejada.

Como reação à seleção, executam-se as preparações correspondentes (p. ex., estabelecendo preços diferentes para o chá para o café). Depois, o usuário tem que pagar pela bebida. O modelo contém um loop que garante que o equipamento entre no estado "pronto para preparar" somente depois que o dinheiro

suficiente foi inserido. Se o usuário parar de inserir moedas antes de inserir a quantidade necessária, o equipamento aguarda 10 segundos antes de devolver o dinheiro inserido e começar tudo de novo. Se tiver sido introduzida uma quantidade suficiente de dinheiro, a bebida é distribuída.

Como única exceção, o equipamento desliga e retorna o dinheiro inserido se não houver ingredientes suficientes (água, chá, café) disponíveis.

Pode-se também ver que o Requisito 1 está ligada ao estado "Chá Selecionado", indicando que um teste que atinge este estado também cobre esta exigência. O Requisito 2 está ligado à transição para "pronto para preparar" para afirmar que o pagamento da bebida é um aspecto importante.