



Certified Tester Syllabus

Test Manager

CTAL-TM 2012

Direitos Autorais

Nota de direitos autorais:

O presente documento pode ser reproduzido total ou parcialmente mediante citação da fonte.

Copyright© Comissão Internacional para Qualificação de Teste de *Software* (doravante denominado ISTQB®)

Subgrupo de trabalho de *Advanced Level Test Manager*: Rex Black (presidente), Judy McKay (vice-presidente), Graham Bath, Debra Friedenber, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson e Tsuyoshi Yumoto (2010-2012).

Histórico de revisões

Versão	Data	Observações
ISEB v1.1	04/09/2001	ISEB Practitioner Syllabus.
ISTQB 1.2E	Setembro de 2003	ISTQB Advanced Level Syllabus do Grupo de Software da Organização Europeia para a Qualidade (EOQ-SG, na sigla em inglês).
V2007	12/10/2007	Versão de 2007 do Certified Tester Advanced Level Syllabus.
D100626	26/06/2010	Incorporação das alterações aceites em 2009 e divisão de capítulos por módulos diferentes.
D101227	27/12/2010	Aceite de alterações no formato e correções que não afetam o significado das frases.
D2011	31/10/2011	Divisão do syllabus, revisão dos objetivos de aprendizagem e alterações no texto para manter a coerência com os objetivos de aprendizagem. Acréscimo de resultados de negócios.
Alfa 2012	09/02/2012	Incorporação de todos os comentários das comissões nacionais recebidos em função do lançamento de outubro.
Beta 2012	26/03/2012	Incorporação de observações das comissões nacionais recebidas em função do lançamento da versão alfa.
Beta 2012	07/04/2012	Envio da versão beta à Assembleia Geral.
Beta 2012	08/06/2012	Envio da versão revisada às comissões nacionais.
Beta 2012	27/06/2012	Incorporação de comentários do EWG e glossário.
RC 2012	15/08/2012	Lançamento de rascunho – inclusão de edições finais das comissões nacionais.
GA 2012	19/10/2012	Edições e correções finais para lançamento por parte da Assembleia Geral.

Índice

Direitos Autorais	2
Histórico de revisões	3
Índice	4
Agradecimentos	7
0 Introdução ao syllabus	8
0.1 Objeto do documento	8
0.2 Visão Geral	8
0.3 Objetivos de aprendizagem sujeitos à avaliação	9
1 Processo de teste ^[420 min]	10
1.1 Introdução	12
1.2 Planejamento, monitoramento e controle de testes	12
1.2.1 Planejamento de testes	12
1.2.2 Monitoramento e controle de testes	14
1.3 Análise de testes	15
1.4 Modelagem de testes	18
1.5 Implementação de testes	18
1.6 Execução de testes	19
1.7 Avaliação de critérios de saída e divulgação	20
1.8 Atividades de fechamento de teste	21
2 Gerenciamento de testes ^[750 min]	23
2.1 Introdução	25
2.2 O gerenciamento de testes em contexto	25
2.2.1 Compreensão dos stakeholders do teste	25
2.2.2 Outras atividades de ciclo de vida de desenvolvimento de software e produtos de trabalho	26
2.2.3 Alinhamento de atividades de teste e outras atividades de ciclo de vida	28
2.2.4 Gerenciamento de testes não funcionais	31
2.2.5 Gerenciamento de testes baseados na experiência	32
2.3 Teste baseado em risco e outras abordagens para priorização de testes e alocação de esforços	33
2.3.1 Teste baseado em risco	33
2.3.2 Técnicas de teste baseado em risco	39
2.3.3 Outras técnicas de seleção de testes	43
2.3.4 Priorização de testes e alocação de esforços no processo de teste	44
2.4 Documentação de testes e outros produtos de trabalho	45
2.4.1 Política de teste	46

ISTQB® Certified Tester Syllabus

Test Manager



2.4.2	Estratégia de teste	47
2.4.3	Plano-mestre de teste.....	49
2.4.4	Plano de teste de nível.....	51
2.4.5	Gerenciamento de riscos de projeto	51
2.4.6	Outros produtos de trabalho do teste	52
2.5	Estimativa de testes.....	53
2.6	Definição e utilização de métricas de teste.....	55
2.7	Valor de negócio dos testes	61
2.8	Teste distribuído, terceirizado e internalizado	62
2.9	Gerenciamento da aplicação de normas do setor	64
3	Revisões ^[180 min]	66
3.1	Introdução	67
3.2	Revisões de gerenciamento e auditorias.....	68
3.3	Gerenciamento de revisões.....	69
3.4	Métricas de revisões.....	71
3.5	Gerenciamento de revisões formais	73
4	Gerenciamento de defeitos ^[150 min]	74
4.1	Introdução	75
4.2	O ciclo de vida dos defeitos e o ciclo de vida do desenvolvimento de software	75
4.2.1	Fluxo de trabalho e estados de defeitos	76
4.2.2	Gerenciamento de relatórios de defeitos inválidos e em duplicidade	77
4.2.3	Gerenciamento de defeitos interfuncionais.....	77
4.3	Informações de relatórios de defeitos.....	78
4.4	Avaliação de recursos de processos com informações de relatórios de defeitos.....	80
5	Aprimoramento de processos de teste ^[135 min]	82
5.1	Introdução	83
5.2	Processo de aprimoramento de testes.....	83
5.2.1	Introdução ao aprimoramento de processos.....	83
5.2.2	Tipos de aprimoramento de processos.....	84
5.3	Aprimoramento de processos de teste	84
5.4	Aprimoramento de processos de teste com TMMi.....	86
5.5	Aprimoramento de processos de teste com TPI Next.....	87
5.6	Aprimoramento de processos de teste com CTP.....	87
5.7	Aprimoramento de processos de teste com STEP	87
6	Ferramentas e automação de testes ^[135 min]	89
6.1	Introdução	90

ISTQB® Certified Tester Syllabus

Test Manager



6.2	Seleção de ferramentas	90
6.2.1	Ferramentas de código aberto	90
6.2.2	Ferramentas personalizadas.....	91
6.2.3	Retorno sobre o investimento (ROI)	92
6.2.4	Processo seletivo	93
6.3	Ciclo de vida das ferramentas.....	95
6.4	Métricas das ferramentas.....	96
7	Habilidades interpessoais – formação de equipes ^[210 min]	97
7.1	Introdução	98
7.2	Habilidades individuais	98
7.3	Dinâmica da equipe de teste.....	100
7.4	Adequação de testes à organização.....	102
7.5	Motivação.....	104
7.6	Comunicação	104
	Referências	107

Agradecimentos

O presente documento foi elaborado pelo grupo principal do subgrupo de trabalho de nível avançado da International Software Testing Qualifications Board – *Advanced Test Manager (TM)* composto por Rex Black (presidente), Judy McKay (vice-presidenta), Graham Bath, Debra Friedenber, Bernard Homès, Paul Jorgensen, Kenji Onishi, Mike Smith, Geoff Thompson, Erik van Veenendaal e Tsuyoshi Yumoto.

O grupo principal gostaria de agradecer à equipe de revisão e às comissões nacionais por suas sugestões e contribuições.

Quando o *Advanced Level Syllabus* foi concluído, o grupo de trabalho de nível avançado contava com os seguintes integrantes (em ordem alfabética):

Graham Bath, Rex Black, Maria Clara Choucair, Debra Friedenber, Bernard Homès (vice-presidente), Paul Jorgensen, Judy McKay, Jamie Mitchell, Thomas Mueller, Klaus Olsen, Kenji Onishi, Meile Posthuma, Eric Riou du Cosquer, Jan Sabak, Hans Schaefer, Mike Smith (presidente), Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

As seguintes pessoas revisaram, comentaram e escolheram este *syllabus*:

Chris van Bael, Graham Bath, Kimmo Hakala, Rob Hendriks, Marcel Kwakernaak, Rik Marselis, Don Mills, Gary Mogyorodi, Thomas Mueller, Ingvar Nordstrom, Katja Piroué, Miele Posthuma, Nathalie Rooseboom de Vries, Geoff Thompson, Jamil Wahbeh e Hans Weiberg.

A Assembleia Geral do ISTQB® lançou este documento formalmente no dia 19 de outubro de 2012.

0 Introdução ao syllabus

0.1 Objeto do documento

O presente *syllabus* forma a base para a certificação internacional de nível avançado *CTAL-TM Test Manager*. O ISTQB® fornece o *syllabus*:

1. Aos Conselhos Nacionais a fim de traduzi-lo para o idioma local e com a finalidade de credenciar os provedores de treinamento. Os Conselhos Nacionais podem adaptar o *syllabus* às suas necessidades linguísticas específicas e modificar as referências para adaptá-las às publicações locais;
2. Aos Provedores de Exames para a elaboração de perguntas no idioma local adaptadas aos objetivos de aprendizagem de cada *syllabus*;
3. Aos Provedores de Treinamento para produzirem os materiais dos cursos e definirem os métodos adequados de ensino;
4. Aos Candidatos à certificação para prepará-los para a avaliação (em função de um curso de treinamento ou independentemente disso);
5. À Comunidade Internacional de Teste e Engenharia de Software e Sistemas para fomentar o desenvolvimento da profissão de teste de *software* e sistemas e servir de base para livros e artigos.

O ISTQB® pode permitir que outras entidades utilizem este *syllabus* para outras finalidades, contanto que procurem e obtenham uma autorização prévia por escrito.

0.2 Visão Geral

O nível avançado é composto por três *syllabi* diferentes:

- Test Manager (TM);
- Test Analyst (TA);
- Technical Test Analyst (TTA).

O documento Advanced Level Modules Overview dá uma visão geral do nível avançado [ISTQB_AL_OVIEW] contém as seguintes informações:

- Os resultados de negócios de cada *syllabus*;
- Um resumo de cada *syllabus*;
- Os relacionamentos entre os *syllabi*;
- Uma descrição dos níveis cognitivos (níveis K);
- Os anexos.

0.3 Objetivos de aprendizagem sujeitos à avaliação

Os objetivos de aprendizagem fundamentam os resultados comerciais e são utilizados na criação de avaliações para a obtenção da certificação *Advanced Test Manager (TM)*. Em geral, todas as partes deste *syllabus* estão sujeitas a uma avaliação no nível K1. Isto é, o candidato reconhecerá e se lembrará de um termo ou de um conceito. Os objetivos de aprendizagem nos níveis K2, K3 e K4 aparecem no começo do capítulo pertinente.

1 Processo de teste [420 min]

Palavras-chave

caso de teste, condição de teste, controle de testes, critério de saída, execução de testes, fechamento de teste, implementação de testes, modelagem de testes, planejamento de testes, procedimento de teste, registro de teste, relatório de resumo de teste, *script* de teste.

Objetivos de aprendizagem

1.2 Planejamento, monitoramento e controle de testes

TM-1.2.1 (K4) Analisar as necessidades de teste de um sistema para planejar atividades de teste e produtos de trabalho que alcançarão os objetivos do teste.

1.3 Análise de testes

TM-1.3.1 (K3) Utilizar a rastreabilidade para a verificar a integridade e a coerência das condições de teste definidas em relação aos objetivos, à estratégia e ao plano de teste;

TM-1.3.2 (K2) Explicar os fatores que podem afetar o nível de detalhamento no qual as condições de teste podem ser especificadas e os prós e os contras do detalhamento.

1.4 Modelagem de testes

TM-1.4.1 (K3) Utilizar a rastreabilidade para a verificação da integridade e da coerência dos casos de teste modelados em relação às condições de teste definidas.

1.5 Implementação de testes

TM-1.5.1 (K3) Utilizar os riscos, a priorização, o ambiente de teste, as dependências de dados e as restrições para o desenvolvimento de um cronograma completo e coerente de execução de testes em relação aos objetivos do teste, à estratégia de teste e ao plano de teste.

1.6 Execução de testes

TM-1.6.1 (K3) Utilizar a rastreabilidade para a o monitorar a integridade e a coerência do andamento do teste em relação aos objetivos do teste, à estratégia de teste e ao plano de teste.

1.7 Avaliação de critérios de saída e divulgação

TM-1.7.1 (K2): Explicar a importância da coleta de informações durante o processo de teste para reforçar a exatidão da divulgação e avaliação em relação aos critérios de saída.

1.8 Atividades de fechamento de teste

TM-1.8.1 (K2) Resumir os quatro grupos de atividades de fechamento de teste;

TM-1.8.2 (K3) Implementar uma retrospectiva do projeto para a avaliação de processos e a descoberta de pontos de melhoria.

1.1 Introdução

O *syllabus* de nível fundamental do ISTQB® descreve o processo básico de teste, que contém as seguintes atividades:

- Planejamento e controle;
- Análise e modelagem;
- Implementação e execução;
- Avaliação de critérios de saída e divulgação;
- Atividades de fechamento de teste.

O *syllabus* do nível fundamental afirma que, embora sigam uma sequência lógica, as atividades do processo podem se sobrepor umas às outras ou acontecer ao mesmo tempo. Normalmente, é preciso ajustar as principais atividades ao contexto do sistema e ao projeto.

Nos *syllabus* de nível avançado, algumas destas atividades são consideradas à parte para refinar e otimizar mais os processos, se adequar melhor ao ciclo de vida de desenvolvimento de *software* e facilitar o monitoramento e o controle eficazes de testes. As atividades ora consideradas são as seguintes:

- Planejamento, monitoramento e controle;
- Análise;
- Modelagem;
- Implementação;
- Execução;
- Avaliação de critérios de saída e divulgação;
- Atividades de fechamento de teste.

1.2 Planejamento, monitoramento e controle de testes

Esta parte trata dos processos de planejamento, monitoramento e controle de testes. Como foi discutido no nível fundamental, as atividades são funções de gerenciamento de testes.

1.2.1 Planejamento de testes

Em cada nível de teste, o planejamento de testes começa no início do processo de teste do nível e continua ao longo do projeto até a conclusão das atividades de fechamento do nível. Isto implica a identificação das atividades e dos recursos necessários para cumprir a missão e alcançar os objetivos identificados na estratégia de teste. O planejamento de testes também inclui a identificação dos métodos de coleta e rastreamento das métricas que serão utilizadas na orientação do projeto, na adesão ao plano e na avaliação da realização dos objetivos. Ao

determinar as métricas úteis durante as etapas de planejamento, as ferramentas podem ser selecionadas, o treinamento pode ser agendado e as diretrizes de documentação podem ser definidas.

A estratégia (ou as estratégias) selecionada para o projeto de teste contribui para a determinação das tarefas que devem ocorrer durante as etapas de planejamento. Por exemplo, ao utilizar a estratégia de teste baseado em risco (ver o capítulo 2), a análise de risco é utilizada para orientar o processo de planejamento de testes em relação às atividades necessárias de mitigação para a redução dos riscos de produto identificados e a facilitação do planejamento de contingências. Se vários defeitos prováveis e possivelmente graves relacionados à segurança forem identificados, o desenvolvimento e a execução de testes de segurança devem ser muito enfatizados. Igualmente, se defeitos graves forem detectados, o que normalmente acontece na especificação da modelagem, o processo de planejamento de testes pode levar a outros testes (revisões) estáticos com as especificações de modelagem.

As informações sobre os riscos também podem ser utilizadas para a determinação das prioridades das diversas atividades de teste. Por exemplo, quando a performance do sistema for de alto risco, o teste de performance poderá ser realizado assim que o código integrado estiver disponível.

Igualmente, se uma estratégia reativa for empregada, talvez haja necessidade de planejamento na criação de cartas de teste e ferramentas para técnicas de teste dinâmico, como testes exploratórios.

Além disso, o planejamento de testes é a etapa em que a abordagem ao teste é claramente definida pelo Gerente de Teste (TM), inclusive os níveis de teste que serão empregados, as metas e os objetivos de cada nível e as técnicas de teste que serão utilizadas em cada nível de teste. Por exemplo, nos testes baseados em risco de certos sistemas de aviação, a avaliação de riscos prescreve o nível de cobertura de código necessário e, portanto, as técnicas de teste que devem ser utilizadas.

Podem existir relacionamentos complexos entre a base de teste (p. ex., requisitos ou riscos específicos), as condições de teste e os testes que as cobrem. Frequentemente, muitos relacionamentos existem entre tais produtos de trabalho. Eles precisam ser assimilados para possibilitar a implementação eficaz do planejamento, do monitoramento e do controle de testes. As decisões sobre as ferramentas também poderão depender da assimilação dos relacionamentos entre os produtos de trabalho.

Também podem existir relacionamentos entre os produtos de trabalho produzidos pelas equipes de desenvolvimento e de teste. Por exemplo, talvez a matriz de rastreabilidade precise rastrear os relacionamentos entre os elementos detalhados de especificação de modelagem dos *designers* do sistema, os requisitos de negócios dos analistas de negócios e os produtos de trabalho definidos pela equipe de teste. Se casos de teste de baixo nível forem modelados e utilizados, talvez se defina, nas etapas de planejamento, o requisito de que os documentos de modelagem detalhados

da equipe de desenvolvimento sejam aprovados antes do início da criação do caso de teste. Ao seguir um ciclo de vida ágil, as sessões informais de transferência de informações podem ser utilizadas para transmitir informações entre equipes antes do início dos testes.

Além disso, o plano de teste elenca as características específicas do *software* que estão na abrangência (com base na análise de risco, se for o caso) e identifica explicitamente as características que não estão na abrangência. Dependendo dos níveis adequados de formalidade e documentação do projeto, cada característica que aparecer na abrangência pode ser relacionada a uma especificação de modelagem de testes correspondente.

Nesta etapa, talvez o Gerente de Teste (TM) também precise trabalhar com os arquitetos do projeto para definir as especificações iniciais do ambiente de teste, confirmar a disponibilidade dos recursos necessários, garantir que as pessoas que configurarão o ambiente se comprometam a fazê-lo e entender os custos ou prazos de entrega e o trabalho necessário para concluir e entregar o ambiente de teste.

Enfim, todas as dependências externas e todos os acordos de nível de serviço (ANS) relacionados devem ser identificados e, se necessário, o contato inicial deve ser realizado. Exemplos de dependências são solicitações de recursos a grupos externos, a dependências de outros projetos (se o trabalho for realizado em um programa), a fornecedores externos, a parceiros de desenvolvimento, à equipe de implantação e aos administradores de bancos de dados.

1.2.2 Monitoramento e controle de testes

Para que o Gerente de Teste controle os testes de maneira eficiente, um cronograma de testes e uma estrutura de monitoramento precisam ser definidos para possibilitar o rastreamento dos produtos de trabalho e dos recursos do teste em comparação com o plano. A estrutura deve incluir as medidas e os alvos detalhados que são necessários para relacionar a situação dos produtos de trabalho e das atividades do teste ao plano e aos objetivos estratégicos.

Em relação a projetos menores e menos complexos, talvez seja relativamente fácil relacionar os produtos de trabalho e as atividades do teste ao plano e aos objetivos estratégicos, mas, no geral, é preciso definir mais objetivos detalhados para conseguir isto. Entre eles podem estar as medidas e os alvos para alcançar os objetivos do teste e a cobertura da base de teste.

A necessidade de relacionar a situação dos produtos de trabalho e as atividades do teste à base do teste de maneira compreensível e relevante para os *stakeholders* do projeto e da empresa é particularmente importante.

A definição de alvos e a medição do andamento com base nas condições do teste e em grupos de condições de teste podem ser utilizadas como meio de realizar isto ao relacionar outros produtos de trabalho do teste à base de teste através das condições de teste. A rastreabilidade adequadamente configurada, inclusive a capacidade de divulgar a situação da rastreabilidade,

deixa os complexos relacionamentos existentes entre os produtos de trabalho em desenvolvimento, a base de teste e os produtos de trabalho do teste mais transparentes e compreensíveis.

Às vezes, as medidas e os alvos detalhados que os *stakeholders* querem que sejam monitorados não dizem respeito diretamente à funcionalidade do sistema ou a uma especificação, especialmente se houver poucos documentos ou se não houver documento nenhum. Por exemplo, o *stakeholder* de uma empresa pode ter mais interesse na definição da cobertura do que no ciclo operacional de negócios, embora a especificação seja definida em termos da funcionalidade do sistema.

O envolvimento dos *stakeholders* das empresas na etapa inicial de um projeto pode contribuir com a definição de tais medidas e alvos, que podem ser utilizados não só para melhorar o controle durante o projeto, mas também para motivar e influenciar as atividades de teste ao longo do projeto.

Por exemplo, as medidas e os alvos dos *stakeholders* podem levar a produtos de trabalho de modelagem e implementação de testes e / ou a cronogramas de execução de testes para facilitar o monitoramento preciso do andamento do teste em comparação com tais medidas. Além disso, os alvos também possibilitam a rastreabilidade em um nível específico do teste e podem facilitar a rastreabilidade das informações em níveis de teste diferentes.

O controle do teste é uma atividade em andamento. Envolve a comparação do progresso real com o plano e a implementação de ações corretivas, quando necessárias. O controle do teste o orienta para o cumprimento da missão, das estratégias e dos objetivos, inclusive a revisão das atividades de planejamento de testes, quando necessárias. As reações adequadas aos dados de controle dependem de informações detalhadas de planejamento.

O capítulo 2 trata dos documentos de planejamento de testes e das atividades de controle de testes.

1.3 Análise de testes

Em vez de considerar a análise e a modelagem de testes em conjunto, como se descreveu no *syllabus* no nível fundamental, os *syllabus* de nível avançado os consideram atividades diferentes, embora reconheçam que podem ser implementadas como atividades paralelas, integradas ou iterativas para facilitar a criação de produtos de trabalho de modelagem de testes.

A análise de testes é a atividade que define “o que” será testado na forma de condições de teste. As condições de teste podem ser identificadas pela análise da base de teste, dos objetivos de teste e dos riscos de produto. Podem ser consideradas medidas e alvos detalhados para o sucesso (p. ex., nos termos dos critérios de saída) e devem proceder da base de teste e de objetivos estratégicos definidos, inclusive dos objetivos de teste e de outros critérios de sucesso de projetos

ou *stakeholders*. As condições de teste também devem conduzir a modelagens de testes e outros produtos de trabalho de testes à medida que os produtos de trabalho são criados.

A análise de testes de determinado nível de testes pode ser realizada assim que a base de teste for definida em tal nível. As técnicas formais de teste e outras técnicas analíticas gerais (p. ex., estratégias analíticas baseadas em risco e estratégias analíticas baseadas em requisitos) podem ser utilizadas na identificação das condições de teste. As condições de teste podem especificar valores ou variáveis dependendo do nível do teste, das informações disponíveis no momento da análise e do nível escolhido de detalhamento (isto é, o grau de granularidade da documentação).

Há uma série de fatores que é preciso levar em consideração na hora de decidir o nível de detalhamento no qual as condições de teste serão especificadas, inclusive:

- O nível de teste;
- O nível de detalhamento e a qualidade da base de teste;
- A complexidade do sistema / do *software*;
- O risco de projeto e de produto;
- O relacionamento entre a base de teste, o objeto do teste e a maneira pela qual o objeto será testado;
- O ciclo de vida de desenvolvimento de *software* utilizado;
- A ferramenta de gerenciamento de testes utilizada;
- O nível no qual a modelagem de testes e outros produtos de trabalho de teste são especificados e documentados;
- As habilidades e os conhecimentos do Analista de Teste (TA);
- O nível de maturidade do processo de teste e a própria organização (repare que uma maturidade maior pode exigir um nível maior de detalhamento ou permitir um nível menor de detalhamento);
- A disponibilidade de outros *stakeholders* do projeto para consulta.

A especificação detalhada das condições de teste tende a resultar em um número maior de condições de teste. Por exemplo, talvez haja apenas uma condição geral de teste, a verificação de testes, em um aplicativo de comércio eletrônico. No entanto, em um documento detalhado de condições de teste, podem ser divididas em várias condições de teste, sendo que há uma condição para cada método de pagamento suportado, uma condição para cada país de destino possível e assim por diante.

Entre as vantagens de especificar as condições de teste detalhadamente estão:

- Maior flexibilidade ao relacionar outros produtos de trabalho de teste (p. ex., casos de teste) à base de teste e aos objetivos de teste, proporcionando um monitoramento e um controle melhores e mais detalhados ao Gerente de Teste;
- A contribuição com a prevenção de defeitos, como já foi discutido no nível fundamental, por acontecer nos primeiros momentos do projeto em altos níveis de teste assim que a

base de teste for definida e, possivelmente, antes da disponibilidade da arquitetura do sistema e do detalhamento da modelagem;

- A relação dos produtos de trabalho de teste aos *stakeholders* em termos que consigam entender (frequentemente, os casos de teste e outros produtos de trabalho de teste não significam nada para os *stakeholders* das empresas e métricas simples como o número de casos de teste executados não querem dizer nada para os requisitos de cobertura dos *stakeholders*);
- Maior influência e orientação não só de outras atividades de teste, mas também de outras atividades de desenvolvimento;
- Possibilidade de modelagem, implementação e execução de testes com a otimização dos produtos de trabalho resultantes através de uma cobertura mais eficiente das medidas e dos alvos detalhados;
- Base para uma rastreabilidade horizontal mais clara no nível do teste.

Entre as desvantagens de especificar as condições de teste detalhadamente estão:

- Pode levar muito tempo;
- A manutenibilidade pode ficar difícil em um ambiente que muda bastante;
- O nível de formalidade precisa ser definido e implementado na equipe toda.

A especificação das condições de teste detalhadas pode ser particularmente eficaz nas seguintes situações:

- Métodos leves de documentação de modelagem de testes, como *checklists*, estão sendo utilizados para contemplarem o ciclo de vida de desenvolvimento, os custos e / ou as restrições de tempo ou outros fatores;
- Poucos requisitos formais ou a ausência de requisitos formais ou outros produtos de trabalho de desenvolvimento estão disponíveis como base de teste;
- O projeto é grande, complexo ou de alto risco e exige um nível de monitoramento e controle que não pode ser realizado através da mera associação de casos de teste com produtos de trabalho de desenvolvimento.

As condições de teste podem ser especificadas com menos detalhes quando a base de teste puder ser relacionada direta e facilmente aos produtos de trabalho de modelagem de testes. Isto acontecerá mais provavelmente com os seguintes:

- Teste de nível de componente;
- Projetos menos complexos em que existem relações hierárquicas simples entre o objeto do teste e o método do teste;
- Teste de aceite em que os casos de uso podem ser utilizados para ajudar a definir os testes.

1.4 Modelagem de testes

A modelagem de testes é a atividade que define *como* algo será testado. Envolve a identificação dos casos de teste pela elaboração por etapas das condições de teste identificadas ou da base de teste com técnicas de teste identificadas na estratégia de teste / ou no plano de teste.

Dependendo das abordagens utilizadas para o monitoramento de testes, o controle de testes e a rastreabilidade, os casos de teste podem manter relação direta (ou indireta através das condições de teste) com a base de teste e os objetivos definidos. Entre tais objetivos estão os objetivos estratégicos, os objetivos do teste e outros critérios do projeto ou dos *stakeholders* para o sucesso.

A modelagem de testes em determinado nível de teste pode ser realizada assim que as condições de teste forem identificadas e informações suficientes estiverem disponíveis para possibilitar a produção de casos de teste de baixo ou alto nível, de acordo com a abordagem à modelagem de testes. Em níveis superiores de teste, é mais provável que a modelagem de testes seja uma atividade à parte após uma análise de testes anterior. Em níveis inferiores de teste, é provável que a análise e a modelagem de testes sejam realizadas como atividade integrada.

Também é provável que algumas tarefas, que normalmente ocorrem durante a implementação de testes, sejam integradas ao processo de modelagem de testes quando uma abordagem iterativa à elaboração dos testes necessários para a execução, por exemplo, a criação de dados de teste, for utilizada. Na verdade, tal abordagem pode otimizar a cobertura das condições de teste, criando, assim, casos de teste de baixo ou alto nível.

1.5 Implementação de testes

A implementação de testes é a atividade durante a qual os testes são organizados e priorizados pelo Analista de Teste (TA). Em contextos formalmente documentados, a implementação de testes é a atividade em que os testes modelados são implementados como casos de teste, procedimentos de teste e dados de teste. Algumas organizações que seguem a norma IEEE829 definem as entradas e os resultados esperados relacionados em especificações de casos de teste e em etapas de teste em especificações de procedimentos de teste.

Frequentemente, as entradas, os resultados esperados e as etapas de cada teste são documentadas juntas. A implementação de testes também inclui a criação de dados de teste armazenados (p. ex., em arquivos simples ou tabelas de bancos de dados).

A implementação de testes também envolve verificações finais para ter a certeza de que a equipe de teste está pronta para a execução dos testes. A verificação pode incluir a garantia de que o ambiente de teste, os dados de teste e o código necessário (possivelmente executando alguns testes em ambientes de teste ou de aceite de código) serão entregues e de que todos os casos de teste serão redigidos e revisados e ficarão prontos para a execução. Também pode incluir a

verificação de critérios de entrada implícitos e explícitos do nível de teste em questão (ver item 1.7). A implementação de testes também pode envolver o desenvolvimento de uma descrição pormenorizada do ambiente e dos dados de teste.

O nível de detalhe e complexidade relacionada do trabalho realizado durante a implementação de testes pode ser influenciado pelo detalhamento dos produtos de trabalho do teste (p. ex., casos e condições de teste). Em alguns casos, particularmente quando os testes forem arquivados para reutilização a longo prazo em testes de regressão, os testes podem fazer descrições detalhadas das etapas necessárias para a execução dos testes a fim de garantir uma execução confiável e coerente apesar do testador que executa o teste. Se os testes forem regulamentados, devem provar a conformidade com as normas pertinentes (ver o item 2.9).

Durante a implementação dos testes, a ordem de execução de testes manuais e automatizados será incluída em um cronograma de execução de testes. O Gerente de Teste deve examinar atentamente as restrições, inclusive riscos e prioridades, que talvez precisem da execução de testes em uma ordem específica ou em um equipamento específico. As dependências no ambiente de teste ou nos dados de teste também precisam ser conhecidas e verificadas.

Podem existir algumas desvantagens nos primeiros momentos da implementação de testes. Com o ciclo de vida ágil, por exemplo, o código pode mudar bastante de uma iteração para outra, tornando obsoleta boa parte dos trabalhos de implementação.

Mesmo sem um ciclo de vida tão suscetível a mudanças como o ágil, qualquer ciclo de vida iterativo ou incremental pode levar a alterações consideráveis entre iterações, o que torna os testes com *script* pouco confiáveis ou sujeitos à necessidade de manutenção frequente. Isso vale para ciclos de vida sequenciais mal gerenciados em que os requisitos mudam com frequência, até mesmo nas fases avançadas do projeto. Antes de embarcar em trabalhos exaustivos de implementação de testes, é bom compreender o ciclo de desenvolvimento do *software* e a previsibilidade das características do *software* que estarão disponíveis para o teste.

Podem existir algumas vantagens nos primeiros momentos da implementação de testes. Por exemplo, testes concretos constituem exemplos comprovados de como o *software* deve se comportar se for elaborado de acordo com a base de teste. Os especialistas em domínios de negócios provavelmente acharão a confirmação de testes concretos mais fácil do que a verificação de regras de negócios abstratas e, assim, talvez identifiquem outros pontos fracos nas especificações do *software*. Os testes verificados podem ilustrar bem o comportamento necessário para *designers* e desenvolvedores de *software*.

1.6 Execução de testes

A execução do teste começa assim que o objeto de teste for entregue e os critérios de entrada para a execução do teste forem atendidos. Os testes devem ser modelados ou, pelo menos, definidos antes da execução dos testes. Devem existir ferramentas, especificamente para o

gerenciamento de testes, o rastreamento de defeitos e, se for o caso, a automação da execução de testes. O rastreamento dos resultados dos testes, inclusive o rastreamento de métricas, deve estar funcionando e os dados rastreados devem ser assimilados por todos os integrantes da equipe. As normas de registro de teste e de divulgação de defeitos devem ser publicadas e estar disponíveis. Ao garantir a existência de tais itens antes da execução de testes, ela pode continuar com eficiência.

Os testes devem ser executados de acordo com os casos de teste, embora o Gerente de Teste deva pensar em dar ao testador alguma liberdade para que possa cobrir outros cenários e comportamentos de testes interessantes que forem observados durante os testes. Ao seguir uma estratégia de teste que seja pelo menos parcialmente reativa, é preciso reservar algum tempo para sessões de teste com técnicas baseadas na experiência e em defeitos. É claro que qualquer falha detectada durante tais testes sem *script* deverá descrever as variações do caso de teste redigido que são necessárias para a reprodução da falha. Os testes automatizados seguirão as instruções definidas sem desvios.

Durante a execução de testes, a principal função do Gerente de Teste consiste em monitorar o andamento do teste de acordo com o plano de teste e, se necessário, iniciar e realizar ações de controle para conduzir o teste a um desfecho bem-sucedido em termos de missão, objetivos e estratégia. Para tal, o Gerente de Teste pode derivar os resultados do teste das condições de teste, da base de teste e, enfim, dos objetivos do teste e fazer uma projeção dos resultados do teste com base em seus objetivos. O processo é descrito detalhadamente no item 2.6.

1.7 Avaliação de critérios de saída e divulgação

A documentação e a divulgação do monitoramento e do controle do andamento de testes são discutidas pormenorizadamente no item 2.6.

Do ponto de vista do processo de teste, é importante garantir a existência de processos eficazes para dispor de uma fonte de informações necessária para a avaliação de critérios de saída e a divulgação.

A definição de informações obrigatórias e de métodos de coleta de informações faz parte do planejamento, do monitoramento e do controle de testes. Durante a análise, a modelagem, a implementação e a execução de testes, o Gerente de Teste deve garantir que os integrantes da equipe de teste responsável pelas atividades forneçam as informações necessárias de maneira exata e oportuna a fim de facilitar processos eficazes de avaliação e divulgação.

A frequência e o nível de detalhamento necessários para a divulgação dependem do projeto e da organização. Eles devem ser negociados durante a etapa de planejamento de testes e devem incluir consultas com os *stakeholders* responsáveis pelo projeto.

1.8 Atividades de fechamento de teste

Assim que a conclusão da execução do teste for determinada, as principais saídas devem ser captadas e repassadas ao responsável ou arquivadas. Coletivamente, são as chamadas atividades de fechamento de teste. As atividades de fechamento de teste são divididas em quatro grandes grupos:

1. **Verificação de conclusão de teste:** a garantia de que todos os trabalhos de teste foram realmente concluídos. Por exemplo, todos os testes planejados devem ser executados ou intencionalmente pulados e todos os defeitos conhecidos devem ser corrigidos e confirmados com testes, encaminhados para lançamento futuro ou aceitos como restrições permanentes;
2. **Transferência de artefatos de teste:** a entrega de produtos de trabalho valiosos às pessoas que precisam deles. Por exemplo, defeitos conhecidos que tenham sido delegados ou aceitos devem ser comunicados às pessoas que utilizarão e apoiarão o uso do sistema. Os testes e os ambientes de teste devem ser propiciados aos responsáveis pelos testes de manutenção. Os conjuntos de testes de regressão (quer automatizados, quer manuais) devem ser documentados e entregues à equipe de manutenção;
3. **Aprendizados:** a realização de ou a participação em reuniões retrospectivas em que aprendizados importantes (tanto do projeto de teste quanto de todo o ciclo de vida de desenvolvimento de *software*) podem ser documentados. Durante tais reuniões, são definidos planos para garantir a repetição das boas práticas e a não repetição das más práticas ou, quando os problemas não puderem ser resolvidos, os planos são contemplados pelos planos dos projetos. Entre as áreas que serão consideradas estão:
 - a. A representação do usuário nas sessões de análise de risco de qualidade foi ampla o suficiente? Por exemplo, devido à detecção tardia de grupos de defeitos imprevistos, a equipe poderia ter decidido que um número maior de representantes de usuários deveria participar das sessões de análise de riscos de qualidade em projetos futuros;
 - b. As estimativas eram exatas? Por exemplo, as estimativas podem ter sido muito mal avaliadas e, assim, futuras estimativas precisarão dar conta disto com os motivos subjacentes, por exemplo, os testes foram ineficientes ou a estimativa foi insuficiente;
 - c. Quais foram tendências e os resultados da análise de causa e efeito dos defeitos? Por exemplo, verifique se solicitações tardias de alterações afetaram a qualidade da análise e do desenvolvimento, busque tendências que indiquem práticas ruins, por exemplo, pular um nível de teste que teria detectado defeitos e de maneira mais rentável por conta de uma suposta economia de tempo. Verifique se as tendências dos defeitos dizem respeito a áreas como novas tecnologias, mudanças de pessoal ou falta de habilidade;
 - d. Existem possíveis oportunidades de aprimoramento de processos? Houve algum desvio imprevisto do plano que deveria ser contemplado no planejamento futuro?

4. **Arquivamento de resultados**, registros, relatórios e outros documentos e produtos de trabalho no sistema de gestão de configurações também deve ser realizado. Por exemplo, o plano de teste e o plano do projeto devem ser armazenados em um arquivo de planejamento com um *link* claro para o sistema e com a versão utilizada.

Estas tarefas são importantes, embora frequentemente desconsideradas, e devem ser explicitamente incluídas no plano de teste.

É comum que uma ou mais destas tarefas sejam omitidas, normalmente por conta de transferências prematuras ou demissões de integrantes da equipe do projeto, de pressão em função de recursos ou prazos em projetos futuros ou do desgaste da equipe. Em projetos realizados com contrato, como o desenvolvimento personalizado, o contrato deve especificar as tarefas exigidas.

2 Gerenciamento de testes [750 min]

Palavras-chave

abordagem de teste, análise de risco, avaliação de riscos, condições de teste, controle de testes, diretor de teste, estimativa de teste, estratégia de teste, gerenciamento de riscos, gerenciamento de testes, identificação de riscos, líder de teste, mitigação de riscos, monitoramento de testes, nível de risco, nível de teste, plano de teste de nível, plano de teste, plano-mestre de teste, política de teste, risco de produto, risco de projeto, risco de qualidade, risco, teste baseado em risco, Wideband Delphi.

Objetivos de aprendizagem

2.2 O gerenciamento de testes em contexto

TM-2.2.1 (K4) Analisar os stakeholders, as circunstâncias e as necessidades de um projeto ou um programa de software, inclusive o modelo de ciclo de vida de desenvolvimento de software, e identificar as atividades de teste ideais;

TM-2.2.2 (K2) Entender como as atividades e os produtos de trabalho do ciclo de vida de desenvolvimento de software afetam os testes, e vice-versa;

TM-2.2.3 (K2) Explicar as formas de gerenciamento de problemas de gerenciamento de testes ligados aos testes baseados na experiência e aos testes não funcionais.

2.3 Teste baseado em risco e outras abordagens para priorização de testes e alocação de esforços

TM-2.3.1 (K2) Explicar as diferentes formas pelas quais os testes baseados em risco reagem aos riscos;

TM-2.3.2 (K2) Explicar e dar exemplos de diferentes técnicas de análise de riscos de produto;

TM-2.3.3 (K4) Analisar, identificar e avaliar os riscos de qualidade de produtos, resumindo os riscos e o nível avaliado com base nas perspectivas dos principais stakeholders do projeto;

TM-2.3.4 (K2) Descrever como os riscos de qualidade de produtos identificados podem ser mitigados e gerenciados de acordo com o nível avaliado de risco ao longo do ciclo de vida e do processo de teste;

TM-2.3.5 (K2) Dar exemplos de opções diferentes para a seleção de testes, a priorização de testes e a alocação de esforços.

2.4 Documentação de testes e outros produtos de trabalho

TM-2.4.1 (K4) Analisar certas amostras de políticas e estratégias de teste e criar planos-mestre de teste, planos de teste de nível e outros produtos de trabalho de teste que são concluídos e conferem com tais documentos;

TM-2.4.2 (K3) Em determinado projeto, analisar os riscos do projeto e selecionar as opções adequadas de gerenciamento de risco (isto é, mitigação, contingência, transferência ou aceite);

TM-2.4.3 (K2) Descrever e exemplificar como as estratégias de teste afetam as atividades de teste;

TM-2.4.4 (K3) Definir as normas e os modelos de documentação referentes a produtos de trabalho de teste que atendem às necessidades da organização, do ciclo de vida e do projeto, adaptando os modelos disponíveis de organismos normativos, conforme for o caso.

2.5 Estimativa de testes

TM-2.5.1 (K3) Em determinado projeto, fazer uma estimativa de todas as atividades do processo de teste, utilizando todas as técnicas adequadas de estimativa;

TM-2.5.2 (K2) Compreender e exemplificar os fatores que podem influenciar as estimativas de teste.

2.6 Definição e utilização de métricas de teste

TM-2.6.1 (K2) Descrever e comparar métricas típicas relacionadas a testes;

TM-2.6.2 (K2) Comparar as diferentes dimensões do monitoramento do andamento dos testes.

TM-2.6.3 (K4) Analisar e divulgar os resultados dos testes em termos de risco residual, situação dos defeitos, situação da execução dos testes, situação da cobertura do teste e confiança para dar insights e fazer recomendações que permitam que os stakeholders do projeto tomem decisões de lançamento.

2.7 Valor comercial dos testes

TM-2.7.1 (K2) Dar exemplos de cada uma das quatro categorias que determinam o custo da qualidade;

TM-2.7.2 (K3) Estimar o valor do teste com base no custo da qualidade juntamente com outras considerações quantitativas e qualitativas e comunicar o valor estimado aos stakeholders do teste.

2.8 Teste distribuído, terceirizado e internalizado

TM-2.8.1 (K2) Entender os fatores necessários para o uso bem-sucedido das estratégias de alocação de pessoal da equipe de testes distribuídos, terceirizados e internalizados.

2.9 Gerenciamento da aplicação de normas do setor

TM-2.9.1 (K2) Resumir as fontes e os usos das normas de teste de software.

2.1 Introdução

No nível avançado, a especialização profissional virou realidade para o testador profissional. Este capítulo se concentra nas áreas do saber que o testador profissional precisa dominar para assumir os cargos de Líder de Teste, Gerente de Teste e Diretor de Teste. Neste *syllabus*, tais profissionais são denominados conjuntamente Gerente de Teste, já que cada organização possui um conceito diferente para os cargos e as alçadas das pessoas que desempenham tais funções.

2.2 O gerenciamento de testes em contexto

Uma das responsabilidades centrais do gestor consiste em garantir e utilizar recursos (pessoas, *software*, *hardware*, infraestrutura etc.) para a realização de processos que agreguem valor. Em relação aos gestores de *software* e TI, os processos frequentemente fazem parte de um projeto ou um programa voltado para a entrega de *software* ou sistemas para uso interno ou externo. Em relação aos Gerentes de Teste (TM), os processos envolvidos no teste, especificamente as atividades fundamentais dos processos de teste descritas no *syllabus* do nível fundamental e no capítulo 1 deste *syllabus*.

Como os processos de teste agregam valor apenas através da contribuição com o sucesso geral do projeto ou do programa (ou através da prevenção de um tipo de falha mais grave), o Gerente de Teste deve planejar e controlar os processos de teste de acordo com eles. Em outras palavras, ele deve providenciar adequadamente os processos de teste, inclusive as atividades e os produtos de trabalho relacionados, de acordo com as necessidades e as circunstâncias dos demais *stakeholders*, de suas atividades (p. ex., o ciclo de vida de desenvolvimento de *software* em que ocorre o teste) e de seus produtos de trabalho (p. ex., especificações de requisitos).

2.2.1 Compreensão dos stakeholders do teste

Os *stakeholders* do teste são os que têm interesse nas atividades de teste, nos produtos de trabalho de teste ou na qualidade do sistema ou do entregável final. O interesse do *stakeholder* pode consistir no envolvimento direto ou indireto nas atividades de teste, no recebimento direto ou indireto dos produtos de trabalho de teste ou no efeito direto ou indireto pela qualidade dos entregáveis produzidos pelo projeto ou pelo programa.

Embora os *stakeholders* de teste variem dependendo do projeto, do produto, da organização e de outros fatores, podem incluir as seguintes funções:

- Desenvolvedores, líderes de desenvolvimento e gestores de desenvolvimento: estes *stakeholders* implementam o *software* testado, recebem resultados de teste e, frequentemente, tomam providências com base em tais resultados (p. ex., a correção de defeitos notificados);

- Arquitetos de bancos de dados, arquitetos de sistemas e *designers*: estes *stakeholders* projetam o *software*, recebem os resultados dos testes e, frequentemente, tomam providências com base em tais resultados;
- Analistas de *marketing* e negócios: estes *stakeholders* determinam as características e o nível de qualidade inerente às características que precisam estar presentes no *software*. Também estão envolvidos na definição da cobertura de teste necessária, na revisão dos resultados dos testes e na tomada de decisões com base nos resultados dos testes;
- Alta administração, gestores de produtos e patrocinadores de projetos: estes *stakeholders* estão frequentemente envolvidos na definição da cobertura de teste necessária, na revisão dos resultados dos testes e na tomada de decisões com base nos resultados dos testes;
- Coordenadores de projetos: estes *stakeholders* são responsáveis pelo gerenciamento de projetos até que tenham um desfecho bem-sucedido, o que exige o equilíbrio das prioridades de qualidade, prazo, características e orçamento. Frequentemente buscam os recursos necessários para as atividades de teste e colaboram com o Gerente de Teste no planejamento e no controle de testes;
- Pessoal de suporte técnico, atendimento ao cliente e *help desk*: estes *stakeholders* prestam assistência a usuários e clientes que desfrutam das características e da qualidade do *software* entregue;
- Usuários diretos e indiretos: estes *stakeholders* utilizam o *software* diretamente (isto é, são os usuários finais) ou recebem saídas ou serviços produzidos ou suportados pelo *software*.

Para maiores informações sobre os *stakeholders* de teste, ver o capítulo 2 de [Goucher09].

A lista de *stakeholders* não é exaustiva. O Gerente de Teste deve identificar os *stakeholders* específicos do teste referente a seu projeto ou programa. O Gerente de Teste também precisa entender a natureza exata da relação do *stakeholder* com o teste e como a equipe de teste atende às necessidades dos *stakeholders*. Além de identificar os *stakeholders* de teste da maneira descrita acima, o Gerente de Teste deve identificar as outras atividades do ciclo de vida de desenvolvimento de *software* e os outros produtos de trabalho que afetam o teste e / ou são afetados pelo teste. Sem isto, o processo de teste pode não alcançar a eficácia e a eficiência ideais (ver o item 2.2.3).

2.2.2 Outras atividades de ciclo de vida de desenvolvimento de software e produtos de trabalho

Como o teste de *software* é uma avaliação da qualidade de um ou mais produtos de trabalho produzidos fora das atividades de teste, costuma existir no contexto de um conjunto maior de atividades de ciclo de vida de desenvolvimento de *software*. O Gerente de Teste deve planejar e orientar as atividades de teste e saber como estas outras atividades e seus produtos de trabalho afetam o teste, como já foi discutido no *syllabus* do nível fundamental, e como o teste afeta estas outras atividades e seus produtos de trabalho.

Por exemplo, em organizações que utilizam práticas de desenvolvimento ágil, os desenvolvedores frequentemente realizam o desenvolvimento orientado por testes, criam testes de unidade automatizados e integram o código continuamente (junto com os testes realizados com tal código) ao sistema de gerenciamento de configurações. O Gerente de Teste deve colaborar com o gestor de desenvolvimento para garantir que os testadores se integrem e se adaptem a estas atividades.

Os testadores poderão revisar os testes de unidade tanto para fazer sugestões de maior cobertura e eficácia em tais testes quanto para entender mais o *software* e sua implementação. O testador poderá avaliar formas de integrar seus próprios testes automatizados, especialmente testes funcionais de regressão, ao sistema de gerenciamento de configurações. [Crispin09]

Embora o relacionamento específico entre as atividades de teste, os demais *stakeholders* do teste, as atividades de trabalho do ciclo de vida de desenvolvimento do *software* e os produtos de trabalho variem dependendo do projeto, do ciclo de vida de desenvolvimento de *software* escolhido e de vários outros fatores, o teste mantém uma forte relação com os seguintes elementos:

- **Engenharia e gerenciamento de requisitos:** o Gerente de Teste precisa levar em consideração os requisitos durante a definição da abrangência e a estimativa dos trabalhos de teste, continuar ciente das alterações nos requisitos e realizar ações de controle de testes para se adaptar às alterações. O Analista Técnico de Teste (TTA) e o Analista de Teste (TA) devem participar da revisão de requisitos;
- **Coordenação de projetos:** o Gerente de Teste, em parceria com o Analista de Teste (TA) e o Analista Técnico de Teste (TTA), deve fornecer o prazo e os requisitos de recursos ao Gerente de Projetos. O Gerente de Teste deve trabalhar com o Gerente de Projetos para entender as mudanças no plano do projeto e realizar as ações de controle de testes para se adaptar a tais mudanças;
- **Gerenciamento de configurações, de lançamentos e de alterações:** o Gerente de Teste, em parceria com a equipe de teste, deve definir os processos e os mecanismos de entrega de objetos de teste e captar os que aparecem no plano de teste. O Gerente de Teste pode pedir ao Analista de Teste (TA) e ao Analista Técnico de Teste (TTA) que criem testes de verificação de pacotes e garantir o controle de versões durante a execução de testes;
- **Desenvolvimento e manutenção de *software*:** o Gerente de Teste deve trabalhar com os gestores de desenvolvimento para coordenar a entrega dos objetos de teste, inclusive o conteúdo e as datas de cada lançamento de teste, e participar do gerenciamento de defeitos (ver o capítulo 4);
- **Suporte técnico:** o Gerente de Teste deve trabalhar com o Gerente de Suporte Técnico para garantir a entrega adequada de resultados de teste durante o fechamento de testes para que os envolvidos na assistência ao produto após o lançamento estejam cientes das falhas e das soluções de contorno conhecidas. Além disso, o Gerente de Teste deve colaborar com o Gerente de Suporte Técnico para analisar as falhas de produção a fim de implementar melhorias no processo de teste;

- **Elaboração de documentos técnicos:** o Gerente de Teste deve trabalhar com o Gerente de Documentação Técnica para garantir a entrega de documentos para teste de maneira oportuna e gerenciar os defeitos detectados nos documentos.

Além de identificar os *stakeholders* de teste da maneira descrita acima, o Gerente de Teste deve identificar as outras atividades do ciclo de vida de desenvolvimento de *software* e os outros produtos de trabalho que afetam o teste ou são afetados pelo teste. Caso contrário, o processo de teste não alcançará a eficácia e a eficiência ideais.

2.2.3 Alinhamento de atividades de teste e outras atividades de ciclo de vida

O teste deve ser parte integrante do projeto, independentemente dos modelos de desenvolvimento de *software* utilizados. Isto inclui:

- **Modelos sequenciais**, como o modelo em cascata, o modelo V e o modelo W. No modelo sequencial, todos os produtos e as atividades de trabalho de determinada fase (p. ex., os requisitos, a modelagem, a implementação, o teste de unidade, o teste de integração, o teste de sistema e o teste de aceite) são concluídos antes do início da etapa seguinte. O planejamento de testes, a análise de testes, a modelagem de testes e a implementação de testes se sobrepõem ao planejamento de projetos, à análise de negócios / requisitos, ao desenho do *software* e bancos de dados e à programação, sendo que a natureza exata da sobreposição depende no nível de teste em questão. A execução de testes procede sequencialmente de acordo com os níveis de teste discutidos no *syllabus* do nível fundamental e neste *syllabus*;
- **Modelos iterativos ou incrementais**, como o desenvolvimento rápido de aplicativos (RAD, na sigla em inglês) e o processo unificado Racional (RUP, na sigla em inglês): em um modelo iterativo ou incremental, as características que serão implementadas são agrupadas (p. ex., por prioridade de negócios ou risco) e, então, as diversas fases do projeto, inclusive os produtos e as atividades de trabalho, ocorrem para cada grupo de características. As etapas podem ser realizadas sequencial ou simultaneamente e as próprias iterações podem ser sequencias ou simultâneas. Durante a iniciação do projeto, o planejamento de testes de alto nível e a análise de testes ocorrem paralelamente com o planejamento de projetos e a análise de negócios / requisitos. O detalhamento de planejamento, análise, modelagem e implementação de testes ocorre no início de cada iteração simultaneamente. A execução de testes costuma envolver níveis de teste simultâneos. Cada nível de teste começa o quanto antes e pode continuar depois que os testes seguintes de nível superior começarem;
- **Metodologia ágil**, como *Scrum* e programação extrema (XP, na sigla em inglês): são os ciclos de vida iterativos em que as iterações são muito curtas (frequentemente, de duas a quatro semanas). Os produtos e as atividades de trabalho de cada iteração são concluídos antes do início da iteração seguinte (isto é, as iterações são sequenciais). Os testes acontecem à maneira

dos modelos iterativos, mas com um grau maior de simultaneidade das diversas atividades de teste e das atividades de desenvolvimento, inclusive uma simultaneidade considerável da execução de testes (em vários níveis) e das atividades de desenvolvimento. Todas as atividades em uma iteração, inclusive as atividades de teste, devem ser concluídas antes do início da iteração seguinte. Em um projeto ágil, a função do Gerente de Teste costuma passar de uma função gerencial direta a uma função de autoridade / consultoria técnica;

- **Espiral:** no modelo em espiral, os protótipos são utilizados no início do projeto para confirmar a viabilidade e testar as decisões de modelagem e implementação com o nível de prioridade de negócios e riscos técnicos para selecionar a ordem segundo a qual as experiências de prototipagem são realizadas. Os protótipos são testados para determinais quais aspectos dos problemas técnicos continuam sem solução. Assim que os principais problemas técnicos forem resolvidos, o projeto segue de acordo com o modelo sequencial ou iterativo.

A fim de adaptar adequadamente as atividades de teste ao ciclo de vida, o Gerente de Teste deve entender bem os modelos de ciclo de vida utilizados na organização. Por exemplo, no modelo V, o processo fundamental de teste do ISTQB® aplicado ao nível de teste de sistema pode ser adequado da seguinte maneira:

- As atividades de planejamento de testes de sistema acontecem junto com o planejamento de projetos e o controle de testes continua até a conclusão da execução e do fechamento de testes de sistema;
- As atividades de análise e modelagem de testes de sistema acontecem junto com as especificações de requisitos, de modelagem de sistemas e arquiteturas (de alto nível) e de modelagem de componentes (de baixo nível);
- As atividades de implementação de testes de sistema podem ter início durante a modelagem de sistemas, embora boa parte delas ocorra junto com o teste de códigos e componentes, sendo que, frequentemente, os trabalhos de implementação de testes de sistema terminam poucos dias antes do começo da execução de testes de sistema;
- As atividades de execução de testes de sistema começam quando os critérios de entrada dos testes de sistema são atendidos (ou desconsiderados), o que, normalmente, significa que pelo menos o teste de componentes e, frequentemente, o teste de integração de componentes foram concluídos. A execução de testes de sistema continua até o cumprimento dos critérios de saída dos testes de sistema;
- A avaliação dos critérios de saída dos testes de sistema e a divulgação dos resultados dos testes de sistema ocorrem durante a execução de testes de sistema, geralmente com uma frequência e uma urgência maior quando os prazos dos projetos se aproximam;
- As atividades de fechamento de testes de sistema ocorrem depois que os critérios de saída dos testes de sistema foram atendidos e a execução de testes de sistema foi concluída, embora, às vezes, possam ser atrasadas até a conclusão dos testes de aceite e a finalização de todas as atividades do projeto.

Em ciclos de vida iterativos ou incrementais, as mesmas tarefas devem ser realizadas, mas o momento e a magnitude delas podem variar. Por exemplo, em vez de implementar o ambiente de testes no início do projeto, talvez seja mais eficiente implementar apenas a parte necessária para a iteração atual. Em qualquer um dos modelos de ciclo de vida iterativos ou incrementais, quanto mais avançado estiver o planejamento, mais longe poderá chegar o escopo do processo de teste fundamental.

Além das etapas de planejamento que ocorrem em cada projeto, a execução de testes e a divulgação podem ser influenciadas pelo ciclo de vida utilizado pela equipe. Por exemplo, em ciclos de vida iterativos, talvez seja eficaz elaborar relatórios inteiros e realizar sessões de revisão pós-iteração antes do início da iteração seguinte. Ao tratar cada iteração como miniprojeto, a equipe recebe a oportunidade de fazer correções e ajustes com base no que aconteceu durante a iteração anterior.

Como as iterações podem ser curtas e podem existir restrições de tempo, talvez faça sentido encurtar o tempo e diminuir os esforços na divulgação e na avaliação, mas as tarefas devem ser realizadas de maneira a rastrear o andamento geral dos testes e identificar quaisquer áreas problemáticas o quanto antes. Os problemas enfrentados nos processos em uma iteração podem facilmente afetar e até mesmo voltar a acontecer na iteração seguinte se medidas corretivas não forem tomadas.

Informações gerais sobre como adaptar os testes a outras atividades de ciclo de vida podem ser captadas na estratégia de testes (ver item 2.4.2). O Gerente de Teste deve realizar o alinhamento específico do projeto referente a cada nível de teste e qualquer combinação escolhida de ciclo de vida de desenvolvimento de *software* e processo de teste durante o planejamento de testes e / ou projetos.

Dependendo das necessidades da organização, do projeto e do produto, outros níveis de teste que forem além dos definidos no *syllabus* de nível fundamental podem ser exigidos, como:

- Teste de integração de *hardware*-software;
- Teste de integração de sistemas;
- Teste de interação entre características;
- Teste de integração de produtos do cliente.

Cada nível de teste deve ter os seguintes elementos claramente definidos:

- Os objetivos de teste, com as metas alcançáveis;
- O escopo e os itens do teste;
- A base de teste, com os meios de medição da cobertura da base (isto é, a rastreabilidade);
- Os critérios de entrada e de saída;
- Os entregáveis do teste, inclusive a divulgação de resultados;
- As técnicas de teste pertinentes e uma forma de garantir os graus adequados de cobertura com tais técnicas;

- As medições e as métricas atinentes aos objetivos do teste, aos critérios de entrada e de saída e à divulgação de resultados (inclusive as medições de cobertura);
- As ferramentas de teste que serão aplicadas em tarefas de teste específicas (se e quando for o caso);
- Os recursos (p. ex., os ambientes de teste);
- Os indivíduos e os grupos responsáveis dentro e fora da equipe de teste;
- A conformidade com normas organizacionais, regulamentares ou outras (se e quando for o caso).

Como já foi discutido neste capítulo, a melhor prática consiste em definir os elementos de maneira coerente em todos os níveis de teste para evitar lacunas perigosas que geram desperdícios em diferentes níveis de testes parecidos.

2.2.4 Gerenciamento de testes não funcionais

O não planejamento de testes não funcionais pode levar à detecção de problemas graves, às vezes desastrosos, de qualidade em um sistema após o lançamento. No entanto, muitos tipos de testes não funcionais são caros. Assim, o Gerente de Teste precisa escolher quais testes não funcionais serão realizados com base em riscos e restrições. Além disso, existem muitos tipos diferentes de testes não funcionais, alguns dos quais não seriam adequados em determinado aplicativo.

Como o Gerente de Teste pode não ter *expertise* o suficiente para lidar com todas as considerações de planejamento, o Gerente de Teste precisa delegar parte das responsabilidades de planejamento de testes ao Analista Técnico de Teste (TTA) (e, em alguns casos, ao Analista de Teste (TA)) alocado às atividades de testes não funcionais. O Gerente de Teste deve pedir que os analistas levem em conta os seguintes fatores gerais:

- Os requisitos dos *stakeholders*;
- As ferramentas necessárias;
- O ambiente de teste;
- Os fatores organizacionais;
- A segurança.

Para obter mais detalhes, consulte o capítulo 4 do syllabus *CTAL-TTA Technical Test Analyst*. Outra consideração importante para o Gerente de Teste consiste em como integrar os testes não funcionais ao ciclo de vida de desenvolvimento de *software*. Um erro comum consiste em esperar até que todos os testes funcionais sejam concluídos antes de dar início a testes não funcionais, o que pode levar à detecção tardia de defeitos não funcionais cruciais.

Em vez disso, os testes não funcionais devem ser priorizados e sequenciados de acordo com o risco. Existem formas de mitigar riscos não funcionais durante os primeiros níveis do teste ou até mesmo durante o desenvolvimento. Por exemplo, as revisões de usabilidade dos protótipos da interface de usuário durante a modelagem de sistemas podem ser muito eficazes na identificação

de defeitos de usabilidade que criariam grandes problemas de prazo se detectados no final dos testes de sistema.

Em ciclos de vida iterativos, o ritmo da mudança e das interações pode dificultar o foco em certos testes não funcionais que exigem a elaboração de estruturas de teste sofisticadas. As atividades de modelagem e implementação de testes que forem além dos prazos de uma única iteração devem ser organizadas como atividades de trabalho além das iterações.

2.2.5 Gerenciamento de testes baseados na experiência

Embora os testes baseados na experiência proporcionem vantagens ao detectar defeitos com eficiência que outras técnicas de teste podem acabar ignorando e confirmem a integridade de tais técnicas, apresenta alguns desafios em gerenciamento de testes. O Gerente de Teste deve estar ciente dos desafios e dos benefícios das técnicas baseadas na experiência, particularmente os testes exploratórios. É difícil determinar a cobertura atingida durante tais testes, considerando o registro leve normal e a preparação avançada mínima de testes. A reprodutibilidade dos resultados dos testes exige cuidados específicos com o gerenciamento, especialmente quando há vários testadores envolvidos.

Uma forma de gerenciar os testes baseados na experiência, especialmente os testes exploratórios, consiste em dividir os trabalhos de teste em períodos menores de 30 a 120 minutos, chamados às vezes de sessões de teste.

A definição do prazo limita e concentra os trabalhos que serão realizados em uma sessão e gera certo nível de monitoramento e conformidade com um cronograma. Cada sessão discute uma carta de teste, que é comunicada oralmente ou por escrito ao testador pelo Gerente de Teste. A carta de teste define as condições de teste que serão discutidas na sessão de teste, o que ajuda a manter o foco e impedir a sobreposição se várias pessoas estiverem realizando testes exploratórios simultaneamente.

Outra técnica de gerenciamento de testes baseados na experiência consiste em integrar os testes auto direcionados e espontâneos em sessões de testes pré-modeladas mais tradicionais. Por exemplo, o testador pode receber permissão (e prazo) para ir além das etapas explícitas, das entradas e dos resultados esperados nos testes predefinidos em suas explorações. O testador também pode receber a incumbência de sessões de testes auto direcionados como parte dos testes diários antes, durante ou depois de um dia de execução de testes predefinidos. Se tais sessões de teste identificarem defeitos ou áreas interessantes para futuros testes, os testes predefinidos podem ser atualizados.

No início das sessões exploratórias, o testador verifica e realiza as tarefas necessárias de configuração dos testes. Durante a sessão, o testador conhece o aplicativo testado, modela e executa os testes de acordo com a técnica aplicada e os aprendizados sobre o aplicativo, investiga quaisquer defeitos e registra os resultados do teste. (Se houver necessidade de repetibilidade de

testes, o testador também deve registrar as entradas, as ações e os eventos dos testes.) Após a sessão, uma sessão de *debriefing* pode acontecer para definir o direcionamento das sessões seguintes.

2.3 Teste baseado em risco e outras abordagens para priorização de testes e alocação de esforços

Um desafio do gerenciamento universal de testes consiste na seleção, na alocação e na priorização adequada de testes. Isto é, com base em um número praticamente infinito de condições de teste e combinações de condições que pode ser discutido, a equipe de teste deve selecionar um conjunto finito de condições, determinar a quantidade adequada de esforço deve ser alocada para cobrir cada condição com casos de teste e sequenciar os casos de teste resultantes por ordem de prioridade de modo a otimizar a eficácia e a eficiência dos trabalhos de teste que devem ser realizados. A identificação e a análise de risco, junto com outros fatores, podem ser utilizadas pelo Gerente de Teste para contribuir com a solução do problema, embora muitas restrições e variáveis que interagem entre si possam exigir um meio-termo.

2.3.1 Teste baseado em risco

O risco é a possibilidade de resultados ou eventos negativos ou indesejáveis. Os riscos existem sempre que algum problema diminui as percepções do cliente, do usuário, do participante ou do *stakeholder* da qualidade do produto ou do sucesso do projeto. Quando o efeito principal dos possíveis problemas estiver na qualidade do produto, os possíveis problemas são chamados de riscos de qualidade, riscos de produto ou riscos de qualidade de produto. Quando o efeito principal dos possíveis problemas estiver no sucesso do projeto, os possíveis problemas são chamados de riscos de projeto ou riscos de planejamento.

No teste baseado em riscos, os riscos de qualidade são identificados e avaliados durante uma análise de risco de qualidade de produtos com os *stakeholders*. Então, a equipe de teste modela, implementa e executa testes para mitigar os riscos de qualidade. A qualidade inclui a totalidade de aspectos, comportamentos, características e atributos que afeta a satisfação do cliente, do usuário e do *stakeholder*. Portanto, o risco de qualidade é uma possível situação em que problemas de qualidade podem existir em um produto.

Entre os exemplos de riscos de qualidade estão cálculos incorretos em relatórios (um risco funcional relacionado à acurácia), a resposta tardia aos comentários de um usuário (um risco não funcional relacionado à eficiência e ao tempo de resposta) e a dificuldade na hora de entender telas e campos (um risco não funcional relacionado à usabilidade e à entendibilidade).

Quando os testes detectam defeitos, mitigam o risco de qualidade ao trazer à tona os defeitos e as oportunidades de lidar com eles antes do lançamento. Quando os testes não detectam defeitos,

mitigam o risco de qualidade ao garantir que, sob as condições de teste, o sistema funciona corretamente.

O teste baseado em risco utiliza os riscos de qualidade do produto para selecionar as condições de teste, alocar os trabalhos de teste àquelas condições e priorizar os casos de teste resultantes. Existem várias técnicas de teste baseado em risco, com variação significativa tanto no tipo e no nível da documentação coletada quanto no nível de formalidade aplicado. Quer explícita, quer implicitamente, o teste baseado em risco visa a utilizar o teste para reduzir o nível geral de risco de qualidade e, especificamente, reduzi-lo a um nível aceitável.

Os testes baseados em risco consistem em quatro atividades principais:

- Identificação de riscos;
- Avaliação de riscos;
- Mitigação de riscos;
- Gerenciamento de riscos.

As atividades se sobrepõem umas às outras. As seguintes subseções discutirão cada uma das atividades.

Para ser mais eficaz, a identificação e a avaliação de riscos devem incluir representantes de todos os grupos de *stakeholders* de projetos e produtos, embora, às vezes, as realidades dos projetos façam que alguns *stakeholders* substituam outros *stakeholders*. Por exemplo, no desenvolvimento de *software* comercial de massa, podem pedir a uma pequena amostra de possíveis clientes que ajude a identificar possíveis defeitos que mais impactariam a utilização do *software*. Neste caso, a amostra de possíveis clientes substitui a base de clientes inteira. Por conta de sua *expertise* específica em riscos e falhas de qualidade em produtos, os testadores devem estar ativamente envolvidos no processo de identificação e avaliação de riscos.

Identificação de riscos

Os *stakeholders* conseguem identificar riscos através de uma ou mais das seguintes técnicas:

- Entrevistas com especialistas;
- Avaliações independentes;
- Uso de modelos de risco;
- Retrospectivas de projetos;
- *Workshops* sobre riscos;
- *Brainstorming*;
- *Checklists*;
- Experiências anteriores.

Ao envolver o maior número possível de *stakeholders*, o processo de identificação de riscos provavelmente detectará a maioria dos riscos significativos de qualidade do produto.

A identificação de riscos frequentemente gera subprodutos, isto é, problemas detectados que não são riscos de qualidade do produto. Entre os exemplos estão questões ou problemas gerais do produto ou do projeto ou problemas em documentos listados, como requisitos e especificações de modelagem. Além disso, os riscos do projeto são frequentemente identificados como subproduto da identificação de riscos de qualidade, mas não são o foco principal dos testes baseados em risco. No entanto, o gerenciamento de riscos de projeto é importante para todos os testes, não só para os testes baseados em risco, e é discutido em maiores detalhes no item 2.4.

Avaliação de riscos

Assim que a identificação de riscos tiver ocorrido, a avaliação de riscos, que é o estudo de tais riscos identificados, pode começar. Especificamente, a avaliação de riscos envolve a categorização de cada risco e a determinação da probabilidade e do impacto relacionado a cada risco. A avaliação de riscos também pode envolver a avaliação ou a atribuição de outras propriedades de cada risco, como o responsável pelo risco.

A categorização do risco significa colocar cada risco em um tipo adequado, como performance, confiabilidade, funcionalidade e assim por diante. Algumas organizações utilizam as características de qualidade da norma ISO 9126 [ISO9126], que está sendo substituída pela norma ISO 25000 [ISO25000], na categorização, mas muitas organizações utilizam outros esquemas de categorização.

A mesma *checklist* utilizada na identificação de riscos é frequentemente usada para dividir os riscos por categoria. As *checklists* genéricas de riscos de qualidade existem e muitas organizações as personalizam. Ao utilizar *checklists* como fundamento da identificação de riscos, a categorização do risco frequentemente ocorre durante a identificação.

A determinação do nível de risco normalmente envolve a avaliação da probabilidade de ocorrência e do impacto após a ocorrência de cada item de risco. A probabilidade de ocorrência é a chance de que exista um possível problema no sistema testado. Em outras palavras, a probabilidade é uma avaliação do nível do risco técnico. Entre os fatores que influenciam as chances de riscos em produtos e projetos estão:

- Tecnologias e equipes complexas;
- Problemas de pessoal e treinamento entre analistas de negócios, modeladores e programadores;
- Conflitos internos na equipe;
- Problemas contratuais com fornecedores;
- Equipes distribuídas geograficamente;
- Embates entre tradições e novas abordagens;
- Ferramentas e tecnologias;
- Lideranças gerenciais ou técnicas fracas;
- Pressões de tempo, recursos, orçamento e gestão;

- Falta de atividades anteriores de garantia de qualidade;
- Altos índices de mudança;
- Altos índices de defeitos anteriores;
- Problemas de interface e integração.

O impacto após a ocorrência é a gravidade do efeito sobre os usuários, os clientes ou outros *stakeholders*. Entre os fatores que influenciam o impacto em riscos em produtos e projetos estão:

- Frequência de uso da característica afetada;
- Importância da característica para bater uma meta de negócios;
- Imagem prejudicada;
- Perda de clientela;
- Possível perda ou responsabilidade financeira, ecológica ou social;
- Sanções legais civis ou penais;
- Perda de alvarás;
- Falta de soluções de contorno razoáveis;
- Visibilidade da falha que levou à publicidade negativa;
- Segurança.

O nível de risco pode ser avaliado quantitativa ou qualitativamente. Se a probabilidade e o impacto puderem ser definidos quantitativamente, é possível multiplicar os dois valores para calcular um número de prioridade risco. Apesar de tudo, normalmente, o nível de risco só pode ser definido qualitativamente. Isto é, é possível falar de chances muito altas, altas, médias, baixas ou muito baixas, mas não é possível expressar a probabilidade como porcentagem com muita precisão.

Igualmente, é possível falar de um impacto muito alto, alto, médio, baixo ou muito baixo, mas não é possível expressar o impacto em termos financeiros de maneira completa e precisa. Esta avaliação qualitativa dos níveis de risco não deve ser considerada inferior às abordagens quantitativas.

Na verdade, quando as avaliações quantitativas de níveis de risco são utilizadas inadequadamente, os resultados enganam os *stakeholders* em relação ao ponto até o qual a pessoa realmente entende e consegue gerenciar o risco. As avaliações qualitativas de níveis de risco são frequentemente combinadas através da multiplicação ou da adição para a criação de uma pontuação de risco agregado.

A pontuação de risco agregado pode ser expressa como um número de prioridade de risco, mas deve ser interpretada qualitativamente em relação à classificação em uma escala ordinal.

Além disso, a menos que se baseie em dados de risco abundantes e estatisticamente válidos, a análise de risco se fundamentará na percepção subjetiva de probabilidade e impacto dos *stakeholders*. Os coordenadores de projetos, programadores, usuários, analistas de negócios, arquitetos e testadores normalmente possuem percepções diferentes e, assim, opiniões possivelmente diferentes em relação ao nível de risco de cada item de risco.

O processo da análise de risco deve incluir alguma forma de chegar a um consenso ou, na pior das hipóteses, definir um nível acordado de risco (p. ex., através de uma ordem da administração ou da obtenção da média, da mediana ou da moda do item de risco). Além disso, a boa distribuição dos níveis de risco deve ser verificada através do alcance para garantir que os índices de risco sirvam de orientação importante em termos de sequenciamento de testes, priorização e alocação de esforços. Caso contrário, os níveis de risco não poderão servir de guia para as atividades de mitigação de riscos.

Mitigação de riscos

O teste baseado em riscos começa com uma análise dos riscos de qualidade (identificação e avaliação de riscos de qualidade do produto). Esta análise é a base do plano-mestre de teste e de outros planos de teste. Como se especificou no/s plano/s, os testes são modelados, implementados e executados para cobrirem os riscos. Os esforços ligados ao desenvolvimento e à execução de testes são proporcionais ao nível de risco, o que significa que técnicas de teste mais meticulosas (como testes alternados) são usadas para riscos maiores, enquanto técnicas de teste menos meticulosas (como partições de equivalência ou testes exploratórios com prazo) são utilizadas para riscos menores.

Além disso, a prioridade do desenvolvimento e da execução dos testes se baseia no nível de risco. Algumas normas relacionadas à segurança (p. ex., a FAA DO-178B/ED 12B, a IEC 61508) prescrevem técnicas de teste e grau de cobertura com base no nível de risco. Além disso, o nível de risco deve influenciar decisões como o uso de revisões de produtos de trabalho do projeto (inclusive testes), o nível de independência, o nível de experiência do testador e o grau de testes de confirmação (retestes) e de regressão realizados.

Durante o projeto, a equipe de teste deve continuar ciente das outras informações que mudam o conjunto de riscos de qualidade e / ou o nível de risco ligado a riscos de qualidade conhecidos. O ajuste periódico da análise de riscos de qualidade, que resulta em ajustes nos testes, deve ocorrer. Os ajustes devem ocorrer, no mínimo, nos momentos importantes do projeto. Os ajustes incluem a identificação de novos riscos, a reavaliação do nível de riscos existentes e a avaliação da eficácia das atividades de mitigação de risco.

Por exemplo, se a sessão de identificação e avaliação de riscos se baseou na especificação de requisitos durante a fase de requisitos, assim que a especificação de modelagem for finalizada, os riscos devem ser reavaliados. Outro exemplo: se um teste revelar que um componente contém um número de defeitos muito maior do que o esperado, é possível concluir que a probabilidade de defeitos nesta área era superior à prevista e, assim, ajustar a probabilidade e o nível geral de risco para cima. Isto pode aumentar a quantidade de testes realizados no componente.

Os riscos de qualidade de um produto também podem ser mitigados antes do início da execução de testes. Por exemplo, se houver problemas nos requisitos durante a identificação de riscos, a equipe do projeto pode fazer revisões pormenorizadas de especificações de requisitos como ação

de mitigação. Isto pode reduzir o nível do risco, o que pode significar que menos testes serão necessários para mitigar os riscos de qualidade restantes.

Gerenciamento de riscos no ciclo de vida

Idealmente, o gerenciamento de riscos ocorre ao longo do ciclo de vida. Se uma organização possuir um documento sobre a política e / ou a estratégia de teste, ele deve descrever o processo geral pelo qual os riscos do produto e do projeto são gerenciados nos testes e mostrar como o gerenciamento de riscos se integra a todas as etapas dos testes e as afeta.

Em uma organização madura em que a consciência do risco permeia a equipe do projeto, o gerenciamento de riscos acontece em muitos níveis e não só para fins de teste. Os riscos importantes não são tratados apenas no início de níveis de teste específicos, mas também nos primeiros níveis de teste. (p. ex., se a performance for identificada como área importante de risco de qualidade, os testes de performance não só terão início nos primeiros momentos do teste de sistema, mas também serão executados durante os testes de unidade e de integração.) As organizações maduras não só identificam os riscos, mas também as fontes de risco e as consequências dos riscos se estes virarem resultados.

Em relação aos defeitos que ocorrerem, a análise de causas-raiz é utilizada para compreender as fontes de risco mais a fundo e implementar melhorias em processos para impedir os defeitos. A mitigação é realizada ao longo do ciclo de vida de desenvolvimento de *software*. A análise de risco é esclarecida, considerando a atividade de trabalho relacionada, a análise do comportamento do sistema, a avaliação de riscos com base em custos, a análise de riscos de produto, a análise de riscos referentes a usuários finais e a análise de riscos de responsabilidade civil.

A análise de risco transcende o teste, e a equipe de teste participa de análises de risco no programa inteiro e suas influências.

A maioria dos métodos de teste baseado em risco também inclui técnicas para a utilização do nível de risco para o sequenciamento e a priorização de testes, garantindo, assim, a cobertura antecipada das áreas mais importantes e a descoberta dos defeitos mais importantes durante a execução de testes. Em alguns casos, todos os testes de risco mais alto são executados antes de quaisquer testes de risco mais baixo e os testes são executados por uma ordem estrita de risco (chamada frequentemente de “primeiro por conteúdo”). Em outros casos, uma amostragem é utilizada para selecionar uma amostra de testes com todos os riscos identificados, usando o risco para ponderar a seleção e, ao mesmo tempo, garantindo a cobertura de todos os itens de risco pelo menos uma vez (chamada frequentemente de “primeiro por abrangência”).

Tanto faz se o teste baseado em risco é executado primeiro por conteúdo ou primeiro por abrangência, é possível que o tempo alocado para o teste termine sem que todos os testes sejam executados. O teste baseado em risco permite que os testadores divulguem os riscos à administração em termos do nível de risco restante em determinado momento e permite que a

administração decida se vai prorrogar o teste ou repassar o risco restante aos usuários, aos clientes, ao *help desk* ou suporte técnico, ou à equipe operacional.

Durante a execução de testes, as técnicas mais sofisticadas de teste baseado em risco – que não precisam ser as mais formais ou intensivas – permitem que os participantes dos projetos, os coordenadores de projetos e produtos, os executivos, os diretores e os *stakeholders* dos projetos monitorem e controlem o ciclo de vida de desenvolvimento de *software*, inclusive através da tomada de decisões de lançamento, com base em um nível residual de risco. Isto exige que o Gerente de Teste divulgue os resultados dos testes em termos de risco de maneira que cada *stakeholder* do teste possa compreendê-los.

2.3.2 Técnicas de teste baseado em risco

Há uma série de técnicas de teste baseado em risco. Algumas destas técnicas são bem informais. Por exemplo, existem abordagens em que o testador analisa os riscos de qualidade durante testes exploratórios [Whittaker09]. Isto pode contribuir para o bom encaminhamento do teste, mas pode levar a um foco excessivo na probabilidade dos defeitos e não em seu impacto, além de não incluir as contribuições interfuncionais dos *stakeholders*. Além disso, tais abordagens são subjetivas e dependem das habilidades, da experiência e das preferências do testador individual. Assim, tais abordagens raramente obtêm todos os benefícios dos testes baseados em risco.

Muitos praticantes usam abordagens leves baseadas em risco para aproveitarem os benefícios dos testes baseados em risco ao mesmo tempo em que minimizam os custos. Elas misturam a reatividade e a flexibilidade das abordagens informais com a força e o consenso de abordagens mais formais. Entre os exemplos de abordagens leves estão: a *Pragmatic Risk Analysis and Management* (PRAM) [Black09], o *Systematic Software Testing* (SST) [Craig02] e o *Product Risk Management* (PRisMa) [vanVeenendaal12]. Além dos atributos comuns dos testes baseados em risco, as técnicas normalmente possuem as seguintes características:

- Evoluem com o tempo com base na experiência do setor com testes baseados em risco, especialmente em setores em que questões de eficiência são importantes;
- Baseiam-se no envolvimento aplicado de uma equipe interfuncional de *stakeholders* que representa as perspectivas empresariais e técnicas durante as etapas iniciais de identificação e avaliação de riscos;
- São otimizadas quando introduzidas durante as primeiras fases de um projeto, quando as opções de mitigar os riscos de qualidade são maximizados e quanto os principais produtos de trabalho e subprodutos da análise de risco podem ajudar a influenciar a especificação e a implementação do produto de forma a minimizar o risco;
- Utilizam as saídas geradas (pela matriz de risco ou pela tabela de análise de riscos) como base para o plano de teste e as condições de teste e, assim todas as atividades seguintes de gerenciamento e análise de testes;

- Embasam a divulgação dos resultados dos testes em termos de risco residual a todas as alçadas de *stakeholders* de teste.

Algumas destas técnicas (p. ex., o SST) exigem especificações de requisitos como entradas na análise de risco e não podem ser utilizadas a menos que as especificações de requisitos sejam fornecidas.

Outras técnicas (p. ex., o PRisMA e a PRAM) incentivam a utilização de uma estratégia mista baseada em riscos e requisitos, sendo que os requisitos e ou / outras especificações servem de entradas na análise de risco, mas podem funcionar totalmente com base nas contribuições dos *stakeholders*.

A utilização dos requisitos como entrada ajuda a garantir uma boa cobertura dos requisitos, mas o Gerente de Teste deve garantir que riscos importantes não indicados pelos requisitos – especialmente em áreas não funcionais – não sejam ignorados. Quando são bons requisitos priorizados servem de entrada, normalmente é possível ver uma forte correlação entre os níveis de risco e a prioridade dos requisitos.

Muitas destas técnicas também incentivam o uso do processo de identificação e avaliação de riscos como forma de criar consenso entre os *stakeholders* na abordagem de teste. Trata-se de uma vantagem e tanto, mas não exige que os *stakeholders* dediquem tempo à participação, quer em sessões de *brainstorming* em grupo, quer em entrevistas *tête-à-tête*.

A participação insuficiente dos *stakeholders* leva a lacunas na análise de riscos. Lógico, nem sempre os *stakeholders* concordam com o nível de risco, então, o responsável pela análise de riscos de qualidade deve trabalhar criativa e positivamente com os *stakeholders* para chegar ao melhor acordo possível. Todas as habilidades de um moderador capacitado à frente de reuniões de revisão se aplicam ao responsável pela análise de riscos de qualidade.

Como técnicas mais formais, as técnicas leves permitem a ponderação da probabilidade e do impacto para dar ênfase aos riscos comerciais ou técnicos (dependendo do nível de teste, por exemplo). Diferentemente de técnicas mais formais, apesar de tudo, as técnicas leves utilizam apenas dois fatores, a probabilidade e o impacto, e opiniões e escalas simples e qualitativas.

A natureza leve de tais abordagens possibilita flexibilidade, a aplicabilidade em uma ampla gama de domínios e acessibilidade em equipes com todos os níveis de experiência e habilidade (até mesmo com o pessoal não técnico e iniciante).

Em testes mais formais e intensivos, o Gerente de Teste conta com uma série de opções:

- **Análise de perigos**, que estende o processo analítico a montante, na tentativa de identificar os perigos subjacentes de cada risco;
- **Custo de exposição**, segundo o qual o processo de avaliação de riscos envolve a determinação de três fatores para cada item de risco de qualidade: 1) a probabilidade (expressa como porcentagem) de falha relacionada ao item de risco; 2) o custo de uma

perda (expressa como quantidade financeira) relacionado a uma falha típica relacionada ao item de risco, se ocorrer na produção; e 3) o custo de testar tais falhas;

- **Failure Mode and Effect Analysis** (FMEA) e suas variantes [Stamatis03], segundo a qual os riscos de qualidade, suas possíveis causas e seus prováveis efeitos são identificados e, então, as classificações de gravidade, prioridade e detecção lhes são atribuídas;
- **Desenvolvimento de função de qualidade** (QFD, na sigla em inglês), que é uma técnica de gerenciamento de riscos de qualidade com implicações de teste especificamente relacionadas aos riscos de qualidade que surgem de uma compreensão incorreta ou insuficiente dos requisitos dos clientes ou dos usuários;
- **Fault Tree Analysis** (FTA), segundo a qual diversas falhas reais (decorrentes de testes ou da produção) ou possíveis (riscos de qualidade) são sujeitadas a uma análise de causas-raiz que começa pelos defeitos que poderiam causar a falha, depois passa aos erros ou aos defeitos que poderiam causar tais defeitos e, então, continua até a identificação das diversas causas-raiz.

As técnicas específicas que devem ser utilizadas nos testes baseados em risco e o grau de formalidade de cada técnica dependem de considerações do projeto, do processo e do produto. Por exemplo, uma abordagem informal, como a técnica exploratória de *Whittaker*, pode ser aplicada em um *patch* ou *quick fix*.

Em projetos ágeis, a análise dos riscos de qualidade é totalmente integrada ao período inicial de cada *sprint* e a documentação dos riscos é misturada com o rastreamento de histórias de usuários. Os sistemas de sistemas exigem uma análise de riscos em cada sistema e no sistema de sistemas. Projetos de segurança crítica e de missão crítica exigem níveis maiores de formalidade e documentação.

As entradas, os processos e as saídas de testes baseados em riscos tendem a ser determinados pela técnica escolhida. Normalmente, as entradas incluem os *insights* dos *stakeholders*, as especificações e os dados históricos. Os processos costumam incluir a identificação, a avaliação e o controle de riscos.

Entre as saídas comuns estão uma lista de riscos de qualidade (com o nível relacionado de risco e a alocação recomendada de trabalhos de teste), os defeitos detectados em documentos de entrada, como especificações, questões ou problemas relacionados aos itens de risco e os riscos do projeto que afetam os testes ou o projeto como um todo.

Normalmente, o fator de sucesso mais importante nos testes baseados em risco é o envolvimento da equipe certa de *stakeholders* na identificação e na avaliação de riscos. Todos os *stakeholders* têm seu próprio conceito do que constitui qualidade em um produto e seu próprio conjunto de prioridades e inquietudes com a qualidade. No entanto, os *stakeholders* tendem a se enquadrar em duas categorias amplas: *stakeholders* comerciais e *stakeholders* técnicos.

Entre os *stakeholders* comerciais estão os clientes, os usuários, os funcionários operacionais, os funcionários do *help desk* e de suporte técnico, entre outros. Eles entendem o cliente e o usuário e, assim, conseguem identificar os riscos e avaliar o impacto do ponto de vista comercial.

Entre os *stakeholders* estão os desenvolvedores, os arquitetos, os administradores de bancos de dados, os administradores de rede, entre outros. Eles entendem as maneiras pelas quais o *software* pode falhar e, assim, conseguem identificar os riscos e avaliar a probabilidade do ponto de vista técnico.

Alguns *stakeholders* possuem uma perspectiva tanto comercial quanto técnica. Por exemplo, os especialistas no assunto que assumem funções de análise de testes ou negócios contam com uma visão mais ampla dos riscos devido à sua *expertise* técnica e comercial.

O processo de identificação de itens de risco gera uma lista considerável de riscos. Os *stakeholders* não precisam discutir sobre os itens de risco. Contanto que um *stakeholder* percebe algo como risco para a qualidade do sistema, esse algo é um item de risco. Contudo, é importante que os *stakeholders* cheguem a um consenso sobre a classificação do nível de risco.

Por exemplo, em abordagens leves que utilizam a probabilidade e o impacto como fatores de classificação, parte do processo deve incluir a identificação de um esquema comum de classificação da probabilidade e do impacto. Todos os *stakeholders*, inclusive o grupo de teste, devem utilizar a mesma escala e devem poder acordar uma única classificação de probabilidade e impacto referente a cada item de risco de qualidade.

Se o teste baseado em risco tiver que ser utilizado a longo prazo, o Gerente de Teste deverá defender e iniciar o teste baseado em risco junto aos *stakeholders*. O grupo interfuncional deve ver o valor da análise de risco para garantir o uso contínuo da técnica. Isto exige que o Gerente de Teste entenda as necessidades e as expectativas dos *stakeholders* e saiba se têm tempo disponível para participarem do processo.

O bom engajamento dos *stakeholders* no processo de análise de riscos de qualidade representa um benefício importante para o Gerente de Teste. O benefício é o de que, em projetos mal especificados com requisitos fracos ou ausentes, os *stakeholders* ainda conseguem identificar os riscos quando orientados por uma *checklist* adequada. O benefício pode ser visto quando, após a implementação do teste baseado em risco, a eficácia na detecção de defeitos da equipe de teste melhora. Isto acontece porque uma base de teste mais completa – neste caso, a lista de itens de risco de qualidade – está sendo utilizada.

Durante o fechamento do teste baseado em risco, as equipes de teste devem medir até que ponto perceberam os benefícios. Em muitos casos, isto implica responder a algumas das quatro perguntas seguintes ou a todas elas através de métricas e consultas com a equipe:

- A equipe de teste detectou uma porcentagem maior de defeitos importantes do que de defeitos menos importantes?

- A equipe de teste encontrou a maioria dos defeitos importantes nos primeiros momentos do período de execução de testes?
- A equipe de teste conseguiu explicar os resultados dos testes aos *stakeholders* em termos de risco?
- Os testes pulados pela equipe de teste (se é que houve algum) possuem um nível menor de risco relacionado do que os testes executados?

Na maioria dos casos, testes baseados em risco bem-sucedidos resultam em respostas afirmativas às quatro perguntas. A longo prazo, o Gerente de Teste deve definir metas de aprimoramento de processos referentes a tais métricas e se empenhar na melhoria da eficiência do processo de análise de riscos de qualidade. Lógico, outras métricas e outros critérios de sucesso podem ser aplicados aos testes baseados em risco e o Gerente de Teste deve pensar bem na relação entre tais métricas, os objetivos estratégicos que a equipe de teste busca e os comportamentos apresentados pela administração com base em um conjunto específico de métricas e critérios de sucesso.

2.3.3 Outras técnicas de seleção de testes

Embora muitos Gerentes de Teste (TM) usem o teste baseado em risco em suas estratégias de teste, muitos também incluem outras técnicas.

Uma das técnicas alternativas mais importantes de desenvolvimento e priorização de condições de teste é o teste baseado em requisitos. O teste baseado em requisitos pode utilizar diversas técnicas, como revisões de ambiguidade, análises de condições de testes e gráficos de causa e efeito. Frequentemente, as revisões de ambiguidade identificam e eliminam as ambiguidades nos requisitos (que servem de base de teste) com uma *checklist* de defeitos comuns em requisitos (ver [Wieggers03]).

Conforme a descrição de [Craig02], a análise das condições de teste implica uma leitura atenta das especificações de requisitos para identificar as condições de teste que devem ser cobertas. Se os requisitos tiverem uma prioridade atribuída, isto pode servir para alocar esforços e priorizar os casos de teste. Se uma prioridade não for atribuída, é difícil determinar os esforços adequados e a ordem do teste sem misturar o teste baseado em requisitos com o teste baseado em riscos.

No módulo do *CTAL-TA Advanced Test Analyst*, os gráficos de causa e efeito servem para cobrir combinações de condições de teste como parte da análise de testes. No entanto, possui um uso mais amplo, já que pode reduzir um problema extremamente grande nos testes a um número razoável de casos de teste e ainda render 100% de cobertura funcional da base de teste.

Os gráficos de causa e efeito também identificam as lacunas na base de teste durante a modelagem de casos de teste, que consegue identificar os defeitos nos primeiros momentos do ciclo de vida de desenvolvimento de *software* quando a modelagem de testes tem início com os pré-requisitos. O principal motivo da não adoção de gráficos de causa e efeito é a complexidade da

criação de tais gráficos. Embora embasem o método, pode ser complicado criar estas ferramentas manualmente.

Um obstáculo geral para o teste baseado em requisitos é o de que, frequentemente, as especificações de requisitos são ambíguas, impossíveis de testar, incompletas ou inexistentes. Nem todas as organizações são motivadas a solucionar estes problemas, assim, o testador se depara com situações em que é preciso escolher outra estratégia de teste.

Outro método que às vezes é utilizado para aumentar o uso dos requisitos existentes é a criação de perfis de uso ou operação, uma abordagem baseada em modelos que utiliza uma mistura de casos de uso, usuários (chamados às vezes de pessoas), entradas e saídas para retratar exatamente o uso do sistema na realidade. Isto permite testar não apenas a funcionalidade, mas também a usabilidade, a interoperabilidade, a confiabilidade, a segurança e a performance.

Durante a análise e o planejamento de testes, a equipe de teste identifica os perfis de uso e tenta cobri-los com os casos de teste. Com base nas informações disponíveis, o perfil de uso é uma estimativa do uso realista do *software*. Isto significa que, à maneira dos testes baseado em risco, os perfis de uso talvez não simulem perfeitamente a possível realidade. No entanto, se houver um número suficiente de informações e entradas dos *stakeholders*, tal modelo bastará (ver [Musa04]).

Alguns Gerentes de Teste também aplicam abordagens metódicas, como *checklists*, para determinar o objeto do teste, até que ponto o objeto deve ser testado e em que ordem ele deve ser testado. Em relação a produtos muitos estáveis, talvez seja suficiente usar uma *checklist* de grandes áreas funcionais e não funcionais a serem testadas e um repositório de casos de teste existentes. A *checklist* faz a heurística da alocação de esforços e o sequenciamento de testes, normalmente com base no tipo e na quantidade de alterações ocorridas.

Tais abordagens tendem a ser menos válidas quando utilizadas para testar alterações que não são pequenas.

Enfim, outro método comum é a utilização de uma abordagem reativa. Na abordagem reativa, muito poucas tarefas de análise, modelagem ou implementação de testes ocorrem antes da execução de testes. A equipe de teste se concentra na reação ao produto realmente entregue. Núcleos de *bugs*, quando detectados, viram o foco de outros testes. A priorização e a alocação são totalmente dinâmicas. A abordagem reativa pode complementar outras abordagens, mas, quando empregada exclusivamente, a abordagem reativa tende a desconsiderar grandes áreas do aplicativo que são importantes, mas que não sofrem com um grande número de *bugs*.

2.3.4 Priorização de testes e alocação de esforços no processo de teste

Independentemente da técnica – ou, melhor ainda, da mistura de técnicas – utilizada pelo Gerente de Teste, ele deve incorporá-la ao projeto e aos processos de teste. Por exemplo, em ciclos de vida

sequenciais (p. ex., o modelo V), a equipe de teste escolhe os testes, aloca os esforços de teste e, inicialmente, prioriza os testes durante a fase de requisitos, com ajustes periódicos, enquanto os ciclos de vida iterativos ou ágeis exigem uma abordagem por iterações. O planejamento e o controle de testes devem levar em conta até que ponto os riscos, os requisitos e / ou os perfis de uso evoluirão e reagir de acordo com isso.

Durante a análise, a modelagem e a implementação de testes, a alocação e a priorização determinadas durante o planejamento dos testes devem ser aplicadas. É comum que, nos processos de teste, ocorram uma análise e / ou uma modelagem cuidadosa apenas para que tais informações não sejam utilizadas para orientar o processo de teste no futuro. Isto normalmente ocorre durante a modelagem e a implementação.

Na execução de testes, a priorização determinada durante o planejamento de testes deve ser realizada, embora seja importante atualizar a priorização de tempos em tempos com base nas informações obtidas após a elaboração inicial do plano. Ao avaliar e divulgar os resultados de teste e a situação dos critérios de saída, o Gerente de Teste também deve avaliá-los e divulgá-los em termos de riscos, requisitos, perfis de uso, *checklists* e outras orientações utilizadas para selecionar e priorizar os testes. Se necessário, a triagem de testes deve ser realizada com base no esquema da priorização de testes.

Nos termos da divulgação de resultados e da avaliação dos critérios de saída, o Gerente de Teste pode medir até que ponto o teste foi concluído. Isto deve rastrear a origem de casos de teste e defeitos detectados na base de teste relevante. Por exemplo, nos testes baseados em risco, à medida que os testes são executados e os defeitos são encontrados, os testadores podem examinar o nível restante de risco residual. Isto apoia o uso de testes baseados em risco para a determinação do momento certo do lançamento. A divulgação dos testes deve lidar com os riscos cobertos e ainda em aberto e com os benefícios alcançados e ainda não alcançados. Para obter um exemplo de divulgação de resultados de testes com base na cobertura de riscos, ver [Black03].

Enfim, durante o fechamento do teste, o Gerente de Teste deve avaliar as métricas e os critérios de sucesso atinentes às necessidades e às expectativas dos *stakeholders* de teste, inclusive as necessidades e as expectativas de clientes e usuários em termos de qualidade. A equipe de teste só poderá dizer que foi realmente eficaz quando os testes atenderem a tais necessidades e expectativas.

2.4 Documentação de testes e outros produtos de trabalho

A documentação é frequentemente produzida como parte das atividades de gerenciamento de testes. Embora os nomes específicos dos documentos de gerenciamento de testes e a abrangência

de cada documento tendam a variar, seguem tipos comuns de documentos de gerenciamento de testes encontrados em organizações e projetos:

- **Política de testes:** descreve os objetivos e as metas de teste da organização;
- **Estratégia de testes:** descreve os métodos de teste gerais e independentes de projetos da organização;
- **Plano-mestre de teste** (ou plano de teste do projeto): descreve a implementação da estratégia de teste em um projeto específico;
- **Plano de teste de nível** (ou plano de teste de fase): descreve as atividades específicas que serão realizadas em cada nível de teste.

A organização física destes tipos de documento varia de um contexto para o outro. Em algumas organizações e em alguns projetos, podem ser combinados em um só documento; em outros, podem ser encontrados em documentos diferentes; em outros, o conteúdo pode ser manifestado intuitivamente, oralmente ou em metodologias tradicionais de teste. Organizações e projetos maiores e mais formais tendem a contar com todos estes documentos como produtos de trabalho escritos, enquanto organizações e projetos menores e menos formais tendem a possuir menos produtos de trabalho escritos. Este *syllabus* descreve cada um dos tipos de documentos à parte, embora, na prática, o contexto da organização e do projeto determine a utilização correta de cada tipo.

2.4.1 Política de teste

A política de teste descreve por que a organização realiza testes. Define os objetivos gerais dos testes que a organização deseja atingir. Esta política deve ser desenvolvida pela equipe sênior de gerenciamento de testes na organização em parceria com os gerentes executivos dos grupos de *stakeholders* de teste.

Em alguns casos, a política de teste complementarará ou fará parte de uma política de qualidade mais geral. Esta política de qualidade descreve os valores e as metas gerais de qualidade da administração.

Quando houver uma política de teste por escrito, pode ser um documento curto de alto nível que:

- Resuma o valor que a organização extrai dos testes;
- Defina os objetivos dos testes, como o aumento da confiança no *software*, a detecção de defeitos no *software* e a redução do nível de riscos de qualidade (ver o item 2.3.1);
- Descreva como avaliar a eficácia e a eficiência dos testes na hora de atender a tais objetivos;
- Exponha o processo normal de testes, talvez com o processo de teste fundamental do ISTQB® como base;
- Especifique como a organização aprimorará seus processos de teste (ver o capítulo 5).

A política de teste deve lidar com as atividades de teste referentes a desenvolvimento e manutenção. Além disso, pode mencionar normas internas e / ou externas referentes aos produtos de trabalho dos testes e à terminologia utilizada na organização.

2.4.2 Estratégia de teste

A estratégia de teste descreve a metodologia de teste geral da organização. Isto inclui a maneira pela qual o teste é utilizado para gerenciar riscos de produto e projeto, a divisão dos testes em níveis e as atividades de alto nível relacionadas aos testes. (A mesma organização pode ter estratégias diferentes para situações diferentes, como ciclos de vida de desenvolvimento de *software*, níveis de risco ou requisitos regulamentares diferentes.) A estratégia de teste e os processos e as atividades descritos nela devem conferir com a política de teste. Deve proporcionar as entradas e os critérios de saída genéricos dos testes à organização ou um ou mais programas.

Como já se mencionou, as estratégias de teste descrevem metodologias gerais de teste, que normalmente incluem:

- **Estratégias analíticas**, como testes baseados em risco, em que a equipe de teste analisa a base de teste para identificar as condições de teste que devem ser cobertas. Por exemplo, nos testes baseados em requisitos, a análise de testes extrai as condições de teste dos requisitos. Então, os testes são modelados e implementados para cobrirem tais condições. Em seguida, os testes são executados, normalmente com a prioridade do requisito coberto por cada teste para determinar a ordem em que os testes serão executados. Os resultados dos testes são divulgados em termos de situações de requisitos, por exemplo, requisitos testados e aprovados, requisitos testados e reprovados, requisitos ainda não totalmente testados, teste de requisito bloqueado *etc.*;
- **Estratégias baseadas em modelos**, como o perfil operacional, em que a equipe de teste desenvolve um modelo (baseado em situações reais ou previstas) do ambiente em que o sistema existe, as entradas e as condições às quais o sistema é sujeito e como o sistema deve se comportar. Por exemplo, em testes de performance baseados em modelos com um aplicativo para dispositivos móveis de forte crescimento, seria possível desenvolver modelos de tráfego de rede de entrada e de saída, de usuários ativos e inativos e da carga resultante de processamento com base no uso atual e nas projeções de crescimento com o tempo. Além disso, modelos podem ser desenvolvidos considerando o *hardware*, o *software*, a capacidade de dados, a rede e a infraestrutura do ambiente de produção atual. Os modelos também podem ser desenvolvidos em relação a taxas de rendimento ideais, esperadas e mínimas, tempos de resposta e alocação de recursos;
- **Estratégias metódicas**, como as baseadas em características de qualidade, em que a equipe de teste utiliza um conjunto predeterminado de condições de teste, como uma norma de qualidade (p. ex., a ISO25000 [ISO25000], que substitui a ISO 9126 [ISO9126]), uma *checklist* ou uma coleção de condições de teste lógicas generalizadas que pode dizer respeito a um domínio

específico, um aplicativo ou um tipo de teste (p. ex., o teste de segurança), e usa tal conjunto de condições de teste de uma iteração para a seguinte e de um *release* para o seguinte. Por exemplo, nos testes de manutenção com um *site* simples e estável de *e-commerce*, os testadores podem utilizar uma *checklist* que identifique as funções, os atributos e os *links* principais de cada página. Os testadores cobririam os elementos relevantes de tal *checklist* sempre que uma alteração fosse feita no *site*;

- **Estratégias que cumprem processos ou normas**, como os sistemas de saúde sujeitos às normas da Administração de Alimentos e Medicamentos dos EUA, em que a equipe de teste segue um conjunto de processos definido por um comitê normativo ou outro grupo de especialistas e em que os processos tratam da documentação, da identificação e da utilização adequadas da base e do/s oráculo/s de teste e da organização da equipe de teste. Por exemplo, em projetos que seguem técnicas de gerenciamento ágil tipo *scrum*, em cada iteração, os testadores analisam as histórias de usuário que descrevem características específicas, estimam os trabalhos de teste de cada característica como parte do processo de planejamento da iteração; identificam as condições de teste (frequentemente chamadas de critérios de aceite) de cada história de usuário, executam testes que cobrem tais condições e divulgam a situação de cada história de usuário (não testada, reprovada, aprovada) durante a execução do teste;
- **Estratégias reativas**, como a utilização de ataques baseados em defeitos em que a equipe de teste espera para modelar e implementar testes até que o *software* seja recebido, reagindo ao sistema realmente testado. Por exemplo, ao utilizar testes exploratórios em um aplicativo baseado em menu, um conjunto de cartas de teste correspondente às características, às seleções de menu e às telas pode ser desenvolvido. Cada testador recebe um conjunto de cartas de teste, que, então, pode ser utilizado para estruturar suas sessões de testes exploratórios. De tempos em tempos, o testador divulga os resultados das sessões de teste ao Gerente de Teste, que pode revisar as cartas com base nas descobertas;
- **Estratégias consultivas**, como testes orientados para usuários, em que a equipe de teste depende das contribuições de um ou mais *stakeholders* importantes para determinar as condições de teste que devem ser cobertas. Por exemplo, em testes de compatibilidade terceirizados com um aplicativo baseado na rede, uma empresa pode dar à prestadora terceirizada de serviços de teste uma lista de prioridades de versões de navegador, *software* antimalware, sistemas operacionais, tipos de conexão e outras opções de configuração que ela quer que seja avaliada no aplicativo. Então, a prestadora de serviços de teste pode utilizar técnicas como testes alternados (para opções de alta prioridade) e partições de equivalência (para opções de menor prioridade) para gerar os testes;
- **Estratégias de teste de contra regressão**, como a automação extensiva, em que a equipe de teste utiliza diversas técnicas para gerenciar o risco de regressão, especialmente a automação de testes de regressão funcionais e / ou não funcionais em um ou mais níveis. Por exemplo, se um teste de regressão for realizado com um aplicativo baseado na rede, o testador pode utilizar uma ferramenta de automação de testes baseada na interface gráfica de usuário para

automatizar os casos de uso típicos e excepcionais do aplicativo. Então, os testes são executados sempre que o aplicativo for modificado.

Estratégias diferentes podem ser combinadas. As estratégias especificamente selecionadas devem atender às necessidades e aos meios da organização e as organizações podem ajustar as estratégias para que estas se adaptem a operações e projetos específicos.

A estratégia de teste pode descrever os níveis de teste que serão realizados. Em tais casos, deve orientar os critérios de entrada e de saída de cada nível e as relações entre os níveis (p. ex., a divisão dos objetivos de cobertura de teste).

A estratégia de teste também pode descrever o seguinte:

- Os procedimentos de integração;
- As técnicas de especificação de testes;
- A independência dos testes (que pode variar dependendo do nível);
- As normas obrigatórias e opcionais;
- Os ambientes de teste;
- A automação de testes;
- As ferramentas de teste;
- A reusabilidade dos produtos de trabalho do *software* e do teste;
- Os testes de confirmação (retestes) e de regressão;
- O controle de testes e a divulgação;
- As medidas e as métricas de testes;
- O gerenciamento de defeitos;
- A abordagem de gerenciamento de configurações ao *testware*;
- As funções e as responsabilidades.

Estratégias de teste diferentes a curto e longo prazo podem ser necessárias. Estratégias de teste diferentes são propícias para organizações e projetos diferentes. Por exemplo, na presença de aplicativos de segurança crítica, uma estratégia mais intensiva pode ser mais adequada do que em outros casos. Além disso, a estratégia de teste também muda em relação aos diversos modelos de desenvolvimento.

2.4.3 Plano-mestre de teste

O plano-mestre de teste cobre todos os trabalhos de teste que serão realizados em um projeto específico, inclusive os níveis específicos que serão realizados e as relações entre tais níveis e entre os níveis de teste e as atividades de desenvolvimento correspondentes. O plano-mestre de teste deve discutir como o testador implementará a estratégia de teste deste projeto (isto é, a abordagem de teste).

O plano-mestre de teste deve conferir com a política e a estratégia de teste e, em áreas específicas em que isto não acontece, deve explicar tais desvios e exceções, inclusive qualquer possível impacto decorrente dos desvios. Por exemplo, se a estratégia de teste de uma organização consistir em realizar testes de regressão em um sistema que não será alterado imediatamente antes do lançamento, mas o projeto atual não tiver testes de regressão, o plano de teste deve explicar por que isto foi planejado e o que será feito para mitigar qualquer risco devido à variação da estratégia de costume.

O plano de teste também deve explicar quaisquer outros efeitos que forem esperados desta variação. Por exemplo, pular o teste de regressão pode exigir o agendamento de um *release* de manutenção um mês após o lançamento inicial do projeto. O plano-mestre de teste deve complementar o plano do projeto ou o guia de operações, o que significa que deve descrever os trabalhos de teste que fazem parte do projeto ou da operação maior.

Embora o conteúdo e a estrutura específica do plano-mestre de teste variem dependendo da organização, de suas normas de documentação e da formalidade do projeto, entre os temas comuns de um plano-mestre de teste estão:

- Os itens que serão testados e os que não serão testados;
- As características de qualidade que serão testadas e as que não serão testadas;
- O cronograma e o orçamento de testes (que devem estar alinhados com o projeto ou com o orçamento operacional);
- Os ciclos de execução de testes e sua relação com o plano de lançamento de *software*;
- As relações e os entregáveis entre os testes e outras pessoas ou departamentos;
- A definição dos itens de teste dentro e fora do escopo em cada nível descrito;
- Os critérios de entrada, os critérios de continuação (suspensão / retomada) e os critérios de saída específicos de cada nível e as relações entre os níveis;
- Os riscos do projeto de teste;
- A governança geral dos trabalhos de teste;
- As responsabilidades pela execução de cada um dos níveis de teste;
- As entradas e as saídas de cada um dos níveis de teste.

Em projetos ou operações menores em que apenas um nível de teste é formalizado, o plano-mestre de teste e o plano de teste referente àquele nível formalizado frequentemente serão combinados em um só documento. Por exemplo, se o teste de sistema for o único nível formalizado, os testes informais de componente e de integração forem realizados pelos desenvolvedores e os testes informais de aceite forem realizados pelos clientes como parte de um processo de teste beta, o plano de teste de sistema inclui os elementos mencionados nesta seção.

Além disso, o teste normalmente depende de outras atividades do projeto. Se as atividades não forem suficientemente documentadas, particularmente em termos de influência e relacionamento com os testes, os temas relacionados a tais atividades podem ser cobertos no plano-mestre de teste (ou no plano de teste de nível adequado). Por exemplo, se o processo de gerenciamento de

configurações não for documentado, o plano de teste deve especificar como os objetos de teste serão entregues à equipe de teste.

2.4.4 Plano de teste de nível

Os planos de teste de nível descrevem as atividades específicas que serão realizadas em cada nível de teste ou, em alguns casos, em cada tipo de teste. Quando necessário, os planos de teste de nível se baseiam no plano-mestre de teste referente ao nível ou ao tipo de teste específico que está sendo documentado. Fornecem prazos, tarefas e marcos que não são necessariamente discutidos no plano-mestre de teste. Além disso, na medida em que normas e modelos diferentes correspondem à especificação de testes em níveis diferentes, os detalhes seriam cobertos em planos de teste de nível.

Em projetos ou operações menos formais, um único plano de teste é frequentemente o único documento de gerenciamento de testes redigido. Em tais situações, alguns dos elementos de informação mencionados acima poderiam ser discutidos no documento do plano de teste.

Em projetos ágeis, os planos de teste tipo *sprint* ou de iteração podem substituir os planos de teste de nível.

2.4.5 Gerenciamento de riscos de projeto

Uma parte importante do planejamento adequado inclui lidar com os riscos do projeto. Os riscos do projeto podem ser identificados com processos semelhantes aos descritos no item 2.3. Quando os riscos do projeto forem identificados, devem ser comunicados ao Gerente de Projetos, que deve tomar providências com base neles. Nem sempre a organização de teste consegue reduzir tais riscos. No entanto, alguns riscos de projeto podem e devem ser mitigados com sucesso pelo Gerente de Teste, como:

- Prontidão de ambiente e ferramentas de teste;
- Disponibilidade e qualificação do pessoal de teste;
- Ausência de normas, regras e técnicas para os trabalhos de teste.

Entre as abordagens ao gerenciamento de riscos de projeto estão a preparação antecipada do *testware*, a realização de testes com os ambientes de teste, a realização de testes com as primeiras versões do produto, a aplicação de critérios de entrada mais rigorosos aos testes, a aplicação de requisitos de testabilidade, a participação em revisões dos primeiros produtos de trabalho do projeto, a participação em gerenciamento de alterações e o monitoramento do andamento e da qualidade do projeto.

Assim que um risco de projeto for identificado e analisado, existem quatro opções principais para gerenciar tal risco:

1. Mitigar o risco através de medidas preventivas para reduzir a probabilidade ou o impacto;
2. Elaborar planos de contingência para reduzir o impacto se o risco virar uma realidade;
3. Transferir o risco a alguma outra parte para que esta lide com ele;
4. Ignorar ou aceitar o risco.

A escolha da melhor opção depende dos benefícios e das oportunidades criados pela opção, do custo e, possivelmente, de quaisquer outros riscos ligados à opção. Quando um plano de contingência for identificado em um risco de projeto, a melhor prática é a de identificar um acionador (que determinará quando e como o plano de contingência será invocado) e um responsável (que realizará o plano de contingência).

2.4.6 Outros produtos de trabalho do teste

Os testes envolvem a criação de uma série de outros produtos de trabalho, como relatórios de defeitos, especificações de casos de teste e registros de teste. A maioria de tais produtos de trabalho é produzida pelo Analista de Teste (TA) e pelo Analista Técnico de Teste (TTA). No *syllabus* do Analista de Teste, as considerações sobre a produção e a documentação de tais produtos de trabalho são discutidas. O Gerente de Teste deve garantir a coerência e a qualidade dos produtos de trabalho através das seguintes atividades:

- Definição e o monitoramento de métricas que monitorem a qualidade de tais produtos de trabalho, como a porcentagem de relatórios de defeitos rejeitados;
- Colaboração com o Analista de Teste e o Analista Técnico de Teste para escolher e personalizar modelos adequados a estes produtos de trabalho;
- Colaboração com o Analista de Teste e o Analista Técnico de Teste para definir as normas destes produtos de trabalho, como o grau de detalhamento necessário em testes, registros e relatórios;
- Revisão de produtos de trabalho de teste com as técnicas adequadas e por parte dos participantes e dos *stakeholders* adequados.

O alcance, o tipo e a especificidade dos documentos do teste podem ser influenciados pelo ciclo de vida de desenvolvimento de *software* escolhido, as normas e os regulamentos aplicáveis e os riscos de qualidade do produto e de projeto ligados ao sistema específico que está sendo desenvolvido, entre outras considerações.

Existem diversas fontes de modelos para testar produtos de trabalho, como a IEEE 829 [IEEE829]. É importante que o Gerente de Teste se lembre que os documentos da IEEE 829 podem ser utilizados em qualquer setor. Assim, os modelos contêm um alto nível de detalhamento que pode ou não corresponder a uma organização específica. É ideal adaptar os documentos da IEEE 829 para a criação de modelos para a utilização em uma organização específica. A aplicação coerente dos modelos reduz os requisitos de treinamento e ajuda a unificar os processos na organização.

O teste também envolve a criação de relatórios de resultados dos testes, que normalmente são elaborados pelo Gerente de Teste e são descritos abaixo.

2.5 Estimativa de testes

A estimativa, como atividade de gerenciamento, é a criação de um alvo aproximado de custos e datas de conclusão relacionado às atividades envolvidas em determinada operação ou projeto. As melhores estimativas:

- Representam o conhecimento coletivo de profissionais experientes e contam com o apoio dos participantes envolvidos;
- Elaboram catálogos específicos e detalhados dos custos, dos recursos, das tarefas e das pessoas envolvidas;
- Apresentam, no tocante a cada atividade estimada, o custo, o esforço e a duração mais prováveis.

Há muito se sabe que a estimativa da engenharia de sistemas e *software* é cheia de dificuldades, tanto técnicas quanto políticas, embora as boas práticas em estimativas de coordenação de projetos sejam geralmente aceitas. A estimativa de testes é a aplicação das boas práticas às atividades de teste relacionadas a um projeto ou uma operação.

A estimativa de teste deve cobrir todas as atividades envolvidas no processo de teste descrito no capítulo 1. As estimativas de custo, esforço e, especialmente, duração da execução de testes são frequentemente as mais interessantes para a administração já que esta costuma ser o caminho crítico do projeto.

No entanto, as estimativas de execução de testes tendem a ser difíceis de gerar e pouco confiáveis quando a qualidade geral do *software* é baixa ou desconhecida. Além disso, a familiaridade e a experiência com o sistema provavelmente afetarão a qualidade das estimativas. Uma prática comum é a de estimar o número de casos de teste necessários durante a execução de testes, mas isto funciona apenas se for possível supor que o número de defeitos no *software* que será testado é baixo. As suposições feitas durante a estimativa sempre devem ser documentadas como parte da estimativa.

As estimativas de teste devem levar em consideração todos os fatores que podem influenciar o custo, o esforço e a duração das atividades de teste. Estes fatores incluem os seguintes, sem se limitar a eles:

- Nível necessário de qualidade do sistema;
- Tamanho do sistema a ser testado;
- Dados históricos dos testes de projetos de teste anteriores, que podem ser acrescidos de dados do setor ou de *benchmarks* de outras organizações;

- Fatores dos processos, inclusive a estratégia de teste, o ciclo de vida de desenvolvimento ou manutenção, a maturidade dos processos e a acurácia da estimativa do projeto;
- Fatores materiais, inclusive a automação e as ferramentas de teste, o/s ambiente/s de teste, os dados de teste, o/s ambiente/s de desenvolvimento, a documentação do projeto (p. ex., requisitos, modelagens *etc.*) e os produtos de trabalho de teste reutilizáveis;
- Fatores de pessoal, inclusive o compromisso, as expectativas, a habilidade, a experiência e as atitudes de gestores, líderes técnicos, executivos e administradores na equipe do projeto, a estabilidade da equipe do projeto, os relacionamentos na equipe do projeto, o suporte ao ambiente de teste e depuração, a disponibilidade de terceirizados e consultores capacitados e o domínio da área;
- Complexidade do processo, da tecnologia, da organização, do número de *stakeholders* de teste, da composição e da localização dos subgrupos;
- As necessidades consideráveis de reforço, treinamento e orientação;
- Assimilação ou o desenvolvimento de novas ferramentas, tecnologias, processos, técnicas, *hardware* personalizado ou uma grande quantidade de *testware*;
- Requisitos com muitas especificações detalhadas de teste, especialmente para cumprir uma norma desconhecida de documentação;
- Complexidade dos prazos de chegada de componentes, especialmente em testes de integração e desenvolvimento de testes;
- Fragilidade dos dados dos testes (p. ex., dados de validade limitada).

A qualidade do *software* entregue para teste também é um grande fator que o Gerente de Teste deve levar em consideração em sua estimativa. Por exemplo, se os desenvolvedores adotaram boas práticas, como o teste de unidade automatizado e a integração contínua, até 50% dos defeitos serão eliminados antes da entrega do código à equipe de teste (ver [Jones11] para saber mais sobre a eficácia destas práticas na eliminação de defeitos). Algumas pessoas afirmam que as metodologias ágeis, inclusive o desenvolvimento orientado por testes, resulta em níveis superiores de qualidade nos testes.

A estimativa pode ser realizada de baixo para cima ou de cima para baixo. Entre as técnicas que podem ser utilizadas na estimativa de testes, quer separadamente, quer em conjunto, estão:

- A intuição, a suposição e a experiência anterior;
- A estrutura analítica do projeto (EAP);
- As sessões de estimativas em equipe (p. ex., *Wideband Delphi*);
- As normas e os regulamentos das empresas;
- As porcentagens dos níveis gerais de trabalho do projeto ou de alocação de pessoal (p. ex., a proporção entre testadores e desenvolvedores);
- O histórico e as métricas da organização, inclusive modelos derivados de métricas que estimam o número de defeitos, o número de ciclos de teste, o número de casos de teste, o esforço médio de cada teste e o número de ciclos de regressão envolvidos;

- As médias do setor e os modelos preditivos, como os pontos de função, as linhas de código, o trabalho estimado dos desenvolvedores ou outros parâmetros do projeto (p. ex., ver [Jones07]).

Na maioria dos casos, a estimativa, quando preparada, deve ser entregue à administração juntamente com uma justificativa (ver o item 2.7). Frequentemente, negociações vêm à tona, o que frequentemente leva à realização de uma nova estimativa. Idealmente, a estimativa de teste final representa o melhor equilíbrio possível entre as metas da organização e do projeto nas áreas de qualidade, prazo, orçamento e características.

Vale lembrar que qualquer estimativa se baseia nas informações disponíveis no momento de sua elaboração. No início de um projeto, as informações podem ser bem limitadas. Além disso, as informações que estiverem disponíveis no início de um projeto podem mudar com o tempo. Para manterem a exatidão, as estimativas devem ser atualizadas para refletirem as novas informações e as mudanças nelas.

2.6 Definição e utilização de métricas de teste

Na gestão, há um clichê que diz que o que é medido é feito. Além disso, o que não é medido não é feito, porque o que não é medido é fácil de ignorar. Portanto, é importante que um conjunto adequado de métricas seja definido para qualquer empreendimento, inclusive os testes.

As métricas de teste podem pertencer a uma ou mais das seguintes categorias:

- As métricas do projeto, que medem o progresso em relação aos critérios de saída definidos do projeto, como a porcentagem de casos de teste executados, aprovados e reprovados;
- As métricas do produto, que medem algum atributo do produto, como até que ponto ele foi testado ou a densidade do defeito;
- As métricas do processo, que medem os recursos do processo de teste ou desenvolvimento, como a porcentagem de defeitos detectada pelo teste;
- As métricas do pessoal, que medem os recursos de indivíduos ou grupos, como a implementação de casos de teste em determinado cronograma.

Qualquer métrica pode pertencer a duas, três ou até mesmo quatro categorias. Por exemplo, uma carta de tendências que mostre a taxa diária de chegada de defeitos pode ser relacionada a um critério de saída (nenhum novo defeito detectado em uma semana), à qualidade do produto (os testes não conseguiram detectar nenhum outro defeito nele) e os recursos do processo de teste (os testes identificam um grande número de defeitos nos primeiros momentos do período de execução de testes).

As métricas de pessoal são particularmente sensíveis. Às vezes, os gestores confundem as métricas que são principalmente métricas de processo com métricas de pessoal, o que leva a resultados desastrosos quando as pessoas distorcem as métricas para que lhes sejam favoráveis. A motivação

e a avaliação adequadas do pessoal de teste são discutidas no capítulo 7 deste *syllabus* e no *Expert Test Management Syllabus*.

No nível avançado, a maior preocupação é o uso das métricas para medir o andamento dos testes, isto é, as métricas do projeto. Algumas das métricas do projeto utilizadas na medição do andamento dos testes também dizem respeito ao produto e ao processo. Outras informações sobre a utilização gerencial das métricas de produto e processo podem ser encontradas no *Expert Test Management Syllabus*. Outras informações sobre a utilização das métricas de processo podem ser encontradas no *Expert Improving the Test Process Syllabus*.

- A utilização das métricas permite que os testadores divulguem os resultados de maneira coerente e possibilita o rastreamento coerente do andamento dos testes com o tempo. Frequentemente, o Gerente de Teste precisa apresentar métricas em diversas reuniões que podem ser frequentadas por *stakeholders* de várias alçadas, do pessoal técnico à diretoria. Como, às vezes, as métricas são utilizadas para determinar o sucesso geral de um projeto, é preciso tomar muito cuidado na hora de determinar o que será rastreado, a frequência da divulgação e o método que será utilizado para apresentar as informações. Em particular, o Gerente de Teste deve levar em consideração o seguinte:
- Definição de métricas: um conjunto limitado de métricas úteis deve ser definido. As métricas devem ser definidas com base em objetivo/s específico/s do projeto, do processo e / ou do produto. As métricas devem ser definidas visando ao equilíbrio, já que uma única métrica pode passar uma impressão enganosa da situação ou das tendências. Assim que as métricas forem definidas, sua interpretação deve ser acordada por todos os *stakeholders* para evitar confusão quando as medições forem discutidas. Frequentemente, existe a tendência de definir métricas demais em vez de se concentrar apenas nas mais pertinentes;
- Rastreamento de métricas: a divulgação e a fusão das métricas devem ser automatizadas ao máximo para reduzir o tempo gasto na coleta e no processamento das medições. As variações das medições com o tempo em relação a uma métrica específica podem refletir informações que condigam não com a interpretação acordada na fase de definição de métricas. O Gerente de Teste deve estar preparado para analisar atentamente possíveis divergências das expectativas nas medições e os motivos de tais divergências;
- Divulgação de métricas: o objetivo consiste em entender as informações imediatamente para fins de gestão. As apresentações podem ilustrar determinada métrica em certo momento ou mostrar a evolução de uma métrica com o passar do tempo para que as tendências possam ser avaliadas;
- Validade das métricas: o Gerente de Teste também deve verificar as informações divulgadas. As medições coletadas por conta de uma métrica podem não refletir a verdadeira situação de um projeto ou podem comunicar uma tendência positiva ou negativa demais. Antes de apresentar qualquer dado, o Gerente de Teste deve revisar sua exatidão e a mensagem que provavelmente passará.

Existem cinco dimensões principais nas quais o andamento dos testes é monitorado:

- Riscos de produto (qualidade);
- Defeitos;
- Testes;
- Cobertura;
- Confiança.

Os riscos de produto, os defeitos, os testes e a cobertura podem ser e frequentemente são medidos e divulgados de modos específicos durante o projeto ou a operação. Se tais medidas disserem respeito a critérios de saída definidos no plano de teste, podem servir de norma objetiva para avaliar a conclusão dos trabalhos de teste. A confiança é mensurável através de pesquisas ou com a cobertura como métrica substituta. No entanto, frequentemente a confiança é comunicada subjetivamente também.

Entre as métricas relacionadas aos riscos de produto estão:

- A porcentagem de riscos totalmente coberta por testes aprovados;
- A porcentagem de riscos em que alguns ou todos os testes são reprovados;
- A porcentagem de riscos que ainda não foi totalmente testada;
- A porcentagem de riscos cobertos classificada por categoria de risco;
- A porcentagem de riscos identificados após a análise inicial de riscos de qualidade.

Entre as métricas relacionadas aos defeitos estão:

- A comparação entre o número cumulativo divulgado (detectado) e o número cumulativo resolvido (corrigido);
- O tempo médio entre as falhas e a taxa de chegada de falhas.

O detalhamento do número ou da porcentagem de defeitos que entra nas seguintes categorias:

- Itens ou componentes de teste específicos;
- Causas-raiz;
- Fonte do defeito (p. ex., especificações de requisitos, novas características, regressão *etc.*);
- *Releases* de testes;
- Introdução, detecção e remoção de fases;
- Prioridade / gravidade;
- Relatórios rejeitados ou em duplicidade;
- Tendências na demora da divulgação à resolução dos defeitos;
- Número de correções de defeitos que criaram outros defeitos (às vezes chamados de *daughter bugs*).

Entre as métricas relacionadas aos testes estão:

- O número total de testes planejados, especificados (implementados), executados, aprovados, reprovados, bloqueados e pulados;

- A situação dos testes de regressão e confirmação, inclusive tendências e totais das reprovações em testes de regressão e confirmação;
- Horas de teste planejadas por dia em comparação com o número real de horas trabalhadas;
- A disponibilidade do ambiente de teste (porcentagem de horas de teste planejadas quando o ambiente de teste puder ser utilizado pela equipe de teste).

Entre as métricas relacionadas à cobertura de teste estão:

- Os requisitos e a cobertura de elementos de modelagem;
- A cobertura de riscos;
- A cobertura de ambientes / configurações;
- A cobertura de códigos.

É importante que o Gerente de Teste saiba como interpretar e utilizar as métricas de cobertura para entender e divulgar a situação dos testes. Em níveis superiores de teste, como os testes de sistema, de integração de sistema e o de aceite, as principais bases de teste costumam ser produtos de trabalho, como especificações de requisitos, especificações de modelagens, casos de uso, histórias de usuário, riscos de produto, ambientes suportados e configurações suportadas. As métricas de cobertura estrutural de códigos se aplicam mais aos níveis inferiores de teste, como os testes de unidade (p. ex., cobertura de comandos e de desvios) e de integração de componentes (p. ex., a cobertura de interface). Embora o Gerente de Teste possa utilizar métricas de cobertura do código para medir até que ponto os testes aplicam a estrutura do sistema testado, a divulgação de resultados de testes de maior nível normalmente não envolve as métricas de cobertura de códigos. Além disso, o Gerente de Teste deve entender que, mesmo se os testes de unidade e de integração de componentes atingirem 100% das metas de cobertura estrutural, os defeitos e os riscos de qualidade continuam sendo tratados em níveis mais altos de teste.

As métricas também podem dizer respeito às atividades do processo de teste fundamental (descrito no *syllabus* do nível fundamental e neste *syllabus*). Ao fazer isto, as métricas podem ser utilizadas no processo de teste para monitorar o processo de teste em si e o progresso em relação aos objetivos do projeto.

Entre as métricas para o monitoramento das atividades de planejamento e controle de testes estão:

- Cobertura de riscos, requisitos e outros elementos da base de teste;
- Detecção de defeitos;
- Horas planejadas e trabalhadas para o desenvolvimento de *testware* e a execução de casos de teste.

Entre as métricas para o monitoramento das atividades de análise de testes estão:

- O número de condições de teste identificadas;
- O número de defeitos detectados durante a análise de testes (p. ex., ao identificar riscos ou outras condições de teste com a base de teste).

Entre as métricas para o monitoramento das atividades de modelagem de testes estão:

- A porcentagem das condições de teste cobertas pelos casos de teste;
- O número de defeitos detectados durante a modelagem de testes (p. ex., o desenvolvimento de testes com a base de teste).

Entre as métricas para o monitoramento das atividades de implementação de testes estão:

- A porcentagem de ambientes de teste configurados;
- A porcentagem de registros de dados de teste carregados;
- A porcentagem de casos de teste automatizados.

Entre as métricas para o monitoramento das atividades de execução de testes estão:

- A porcentagem de casos de teste planejados que foram executados, aprovados e reprovados;
- A porcentagem das condições de teste cobertas por casos de teste executados (e / ou aprovados);
- A divulgação / a resolução planejada e real de defeitos;
- A cobertura planejada e real atingida.

As métricas para o monitoramento das atividades de realização e conclusão de testes incluirão o mapeamento de marcos, critérios de entrada e critérios de saída (definidos e aprovados no planejamento de testes), que pode incluir os seguintes elementos:

- O número de condições de teste, casos de teste ou especificações de teste planejados e executados que foram aprovados ou reprovados;
- O total de defeitos, frequentemente dividido por gravidade, prioridade, situação atual, subsistema afetado ou outra classificação (ver capítulo 4);
- O número de mudanças exigidas, aceitas, desenvolvidas e testadas;
- O custo planejado e o real;
- A duração planejada e a real;
- As datas planejadas e reais para o teste de marcos;
- As datas planejadas e reais para os marcos do projeto relacionados aos testes (p. ex., *code freeze*);
- A situação do risco de produto (qualidade), frequentemente dividida em riscos mitigados e riscos não mitigados, grandes áreas de risco, novos riscos detectados após a análise de testes *etc.*;
- A porcentagem de perda de testes, custos ou tempo devido a eventos de bloqueio ou alterações planejadas;
- A situação de testes de confirmação e regressão.

Entre as métricas para o monitoramento das atividades de fechamento de testes estão:

- A porcentagem de casos de teste executados, aprovados, reprovados, bloqueados e pulados durante a execução de testes;
- A porcentagem de casos de teste introduzidos em repositórios de casos de teste reutilizáveis;
- A porcentagem de casos de teste automatizados ou casos de teste automatizados planejados e reais;
- A porcentagem de casos de teste integrados aos testes de regressão;
- A porcentagem de relatórios de defeitos resolvidos / não resolvidos;
- A porcentagem dos produtos de trabalho de teste identificados e arquivados.

Além disso, as técnicas-padrão de gerenciamento de projetos, como as estruturas analíticas de projetos são frequentemente utilizadas para o monitoramento dos processos de teste. Em equipes ágeis, os testes fazem parte do andamento da história de usuário no gráfico de *Burndown*. Quando forem utilizadas técnicas esbeltas de gestão, o andamento do teste por histórias é frequentemente monitorado ao fazer o cartão de histórias de usuário atravessar uma coluna no quadro *Kanban*.

Com um conjunto definido de métricas, as medições podem ser divulgadas oralmente na forma de narrativas, numericamente em tabelas ou ilustrativamente em gráficos. As medições podem ser utilizadas para uma série de finalidades, inclusive:

- A análise para descobrir as tendências e as causas que podem ser discernidas através dos resultados dos testes;
- A divulgação para comunicar as descobertas dos testes aos participantes e aos *stakeholders* do projeto;
- O controle para mudar o roteiro do teste ou do projeto como um todo e monitorar os resultados do ajuste de roteiro.

As formas adequadas de coletar, analisar e divulgar tais medidas de teste dependem das necessidades de informação, das metas e da capacidade específica das pessoas que utilizarão as medições. Além disso, o conteúdo específico dos relatórios de teste deve variar com base no público.

Para fins de controle de testes, é essencial que as métricas ao longo do processo de testes (assim que o planejamento de testes estiver completo) proporcionem ao Gerente de Teste as informações necessárias para orientar os trabalhos de teste à conclusão bem-sucedida da missão, das estratégias e dos objetivos dos testes. Portanto, o planejamento deve levar em consideração as necessidades de informação e o monitoramento deve incluir a coleta de quaisquer métricas de produtos de trabalho necessárias. O volume de informações necessárias e o trabalho realizado para coletá-las dependem de diversos fatores do projeto, inclusive porte, complexidade e risco.

O controle de testes deve responder às informações geradas pelos testes e às mudanças nas condições em que determinado projeto ou empreendimento existe. Por exemplo, se o teste dinâmico revelar núcleos de defeitos em áreas em que os defeitos eram considerados improváveis

ou se o período de execução de testes for reduzido devido a um atraso no início dos testes, a análise de riscos e o plano deverão ser revisados. Isto pode levar à repriorização dos testes e à realocação dos trabalhos restantes de execução de testes.

Se divergências do plano de teste forem descobertas através do relatório de andamento de testes, o controle de testes deve ser realizado. O controle de testes visa a remanejar o projeto e / ou o teste e encaminhá-lo ao sucesso. Ao utilizar os resultados dos testes para influenciar ou medir os trabalhos de controle no projeto, as seguintes opções devem ser levadas em consideração:

- A revisão da análise de riscos de qualidade, das prioridades de teste e / ou dos planos de teste;
- A adição de recursos ou o crescimento do projeto ou do teste;
- O atraso da data de lançamento;
- O relaxamento ou o fortalecimento dos critérios de saída do teste;
- A mudança do escopo (funcional e / ou não funcional) do projeto.

A implementação de tais opções normalmente exige o consenso entre os *stakeholders* do projeto ou da operação e a autorização dos coordenadores do projeto ou da operação.

As informações constantes em relatórios de defeitos devem depender bastante das necessidades de informação do público-alvo, por exemplo, o gerenciamento de projetos ou de negócios. Para um *Gerente de Projetos*, ter informações detalhadas sobre defeitos é provavelmente interessante. Para um *Business Manager*, a situação dos riscos de produto pode ser o assunto mais importante.

2.7 Valor de negócio dos testes

O Gerente de Teste deve trabalhar para otimizar os testes a fim de entregar um bom valor de negócio. A realização excessiva de testes não gera bom valor de negócio, pois os testes criam atrasos prolongados e custam mais do que economizam. A realização insuficiente de testes não gera bom valor de negócio, pois defeitos demais serão entregues aos usuários. O ideal fica entre estes dois extremos. O Gerente de Teste deve ajudar os *stakeholders* a entender o ideal e o valor que os testes geram.

Embora a maioria das organizações considere os testes valiosos em algum sentido, poucos gestores, inclusive o Gerente de Teste, conseguem quantificar, descrever ou articular tal valor. Além disso, muitos Gerentes de Teste (TM), líderes de teste e testadores se concentram nos detalhes táticos dos testes (aspectos próprios de determinada tarefa ou nível de teste ao mesmo tempo em que desconsideram questões estratégicas maiores (de nível mais alto) relacionadas aos testes em que outros participantes do projeto, especialmente os gestores, têm interesse.

O teste agrega valor à organização, ao projeto ou à operação quantitativa e qualitativamente:

- Os valores quantitativos incluem a detecção de defeitos que são impedidos ou corrigidos antes do lançamento, a detecção de defeitos conhecidos antes do lançamento (não

corrigidos, porém documentados, talvez com soluções de contorno), a redução do risco através da execução de testes e a divulgação de informações sobre a situação do projeto, do processo e do produto;

- Entre os valores qualitativos estão a melhoria da imagem por conta da qualidade, lançamentos mais tranquilos e mais previsíveis, o aumento da confiança, a proteção contra a responsabilidade legal e a redução do risco de perda de missões inteiras ou até mesmo de vidas.

O Gerente de Teste deve entender quais destes valores correspondem à sua organização, ao seu projeto ou à sua operação e deve conseguir articular os testes em termos destes valores.

Um método consagrado de medição do valor quantitativo e da eficiência dos testes se chama custo da qualidade (ou, às vezes, custo da má qualidade). O custo da qualidade envolve a classificação de custos operacionais e de projetos em quatro categorias relacionadas aos custos advindos dos defeitos em produtos:

- Custos de prevenção, por exemplo, a capacitação de desenvolvedores para criarem um código com maior manutenibilidade ou segurança;
- Custos de detecção, por exemplo, a elaboração de casos de teste, a configuração de ambientes de teste e a revisão dos requisitos;
- Custos de falhas internas, por exemplo, a correção de defeitos detectados durante os testes ou as revisões antes da entrega;
- Custos de falhas externas, por exemplo, custos de suporte ligados ao *software* defeituoso entregue aos clientes.

Uma parte do orçamento de teste consiste no custo da detecção (isto é, o dinheiro que seria gasto mesmo se os testadores não encontrassem nenhum defeito, como o dinheiro gasto no desenvolvimento de testes), enquanto o resto diz respeito ao custo de falhas internas (isto é, o custo real ligado aos defeitos detectados). Os custos totais de detecção e falhas internas ficam normalmente bem abaixo dos custos de falhas externas, que transformam os testes em itens de excelente valor. Ao determinar os custos nestas quatro categorias, o Gerente de Teste consegue defender a realização de testes em um contexto comercial de maneira convincente.

Para maiores informações sobre o valor comercial dos testes, inclusive o custo da qualidade, consulte [Black03].

2.8 Teste distribuído, terceirizado e internalizado

Em muitos casos, parte dos trabalhos de teste ou até mesmo a totalidade deles é realizada por pessoas em locais diferentes, empregadas por empresas diferentes ou separadas da equipe do projeto. Se os trabalhos de teste acontecerem em vários locais, os trabalhos de teste são distribuídos. Se os trabalhos de teste forem realizados em um ou mais locais por pessoas que não são funcionários da empresa e não dividem o mesmo local com a equipe do projeto, os trabalhos

de teste são terceirizados. Se os trabalhos de teste forem realizados por pessoas que dividem o mesmo local com a equipe do projeto, mas que não são seus colegas, os trabalhos de teste são internalizados.

Todos os trabalhos de teste têm uma coisa em comum: a necessidade de canais claros de comunicação e expectativas bem definidas de missões, tarefas e entregáveis. A equipe do projeto deve depender menos dos canais de comunicação informal, como conversas nos corredores e a socialização entre colegas. É importante ter formas certas de comunicação, inclusive para lidar com assuntos como a intensificação de problemas, os tipos de informação que serão comunicados e os métodos de comunicação que serão utilizados. Todos, em todos os lados dos relacionamentos da equipe, devem entender bem suas funções e suas responsabilidades e as funções e as responsabilidades das outras partes para evitarem mal-entendidos e expectativas irrealistas. O local, o fuso horário, as diferenças culturais e linguísticas fazem que os problemas de comunicação e expectativas sejam ainda mais prováveis.

Além disso, os trabalhos de teste têm outra coisa em comum: a necessidade de alinhamento de metodologias. Embora o mau alinhamento de metodologias possa acontecer em qualquer projeto, é mais provável que ocorra em situações em que o trabalho é distribuído e / ou realizado por entidades externas. Se dois grupos de teste utilizam metodologias diferentes ou o grupo de teste usa uma metodologia diferente da dos grupos de desenvolvimento ou de coordenação de projetos, isto pode levar a problemas consideráveis, especialmente durante a execução de testes. Por exemplo, se um cliente usar o desenvolvimento ágil enquanto a prestadora de serviços de teste possui uma metodologia de teste predefinida que supõe um ciclo de vida sequencial, haverá controvérsias referentes ao prazo e à natureza da entrega dos itens de teste para a prestadora de serviços de teste.

Em relação aos testes distribuídos, a divisão dos trabalhos de teste em vários locais deve ser explícita e decidida de maneira inteligente. Sem tais orientações, o grupo mais componente pode nem realizar os testes para os quais estão qualificados. Além do mais, se nenhuma equipe souber suas responsabilidades, talvez ela não faça o que deveria fazer. As expectativas de cada uma das equipes devem ser claramente comunicadas. Sem um gerenciamento cuidadoso, os trabalhos de teste como um todo podem sofrer com lacunas (que aumentam o risco de qualidade residual na entrega) e redundâncias (o que reduz a eficiência).

Enfim, em todos os trabalhos de teste, é crucial que a equipe de projeto inteira desenvolva e mantenha a confiança de que a/s equipe/s de teste realizará/ão suas funções adequadamente apesar das barreiras organizacionais, culturais, linguísticas e geográficas. A falta de confiança leva a ineficiências e atrasos ligados à verificação de atividades, à atribuição de culpa por causa de problemas e à politicagem organizacional.

2.9 Gerenciamento da aplicação de normas do setor

Nos *syllabus* dos níveis fundamental e avançado, alusões são feitas a uma série de normas. As normas mencionadas cobrem os ciclos de vida de desenvolvimento de *software*, os testes de *software*, as características de qualidade de *software*, as revisões e o gerenciamento de defeitos. O Gerente de Teste deve estar ciente das normas, da política da organização sobre a utilização de normas e da exigência, da necessidade ou da utilidade das normas.

As normas podem vir de fontes diferentes, como:

- Internacionais ou com objetivos internacionais;
- Nacionais, como aplicações nacionais de normas internacionais;
- De áreas específicas, como quando normas internacionais ou nacionais são adaptadas a áreas específicas ou desenvolvidas para áreas específicas.

As entidades normativas internacionais incluem a ISO e o IEEE. A ISO, *International Standards Organization*, também conhecida como IOS, *International Organization for Standardization*. É composta por representantes dos organismos nacionais mais relevantes da área padronizada. Esta entidade internacional promoveu uma série de normas úteis para os testadores de *software*, como a ISO 9126 (substituída pela ISO 25000), a ISO 12207 [ISO12207] e a ISO 15504 [ISO15504].

O IEEE é o *Institute of Electrical and Electronics Engineers*, uma organização profissional sediada nos EUA com representantes nacionais em mais de cem países. Esta organização propôs uma série de normas úteis para os testadores de *software*, como a IEEE 829 [IEEE829] e a IEEE 1028 [IEEE1028].

Muitos países têm suas próprias normas nacionais. Algumas delas são úteis para o teste de *software*. Um exemplo é a norma britânica BS 7925-2 [BS7925-2], que contém informações relacionadas a muitas das técnicas de modelagem de testes descritas no *Advanced Test Analyst (TA) Syllabus* e no *Advanced Technical Test Analyst (TTA) Syllabus*.

Algumas normas são próprias de áreas específicas e algumas delas possuem implicações para o teste de *software*, a qualidade do *software* e o desenvolvimento de *software*. Por exemplo, em aviação, a norma DO0178B da Agência Nacional de Aviação Civil dos EUA (e a ED-12B, o equivalente europeu) corresponde ao *software* utilizado em aeronaves civis. Esta norma prescreve certos níveis de critérios de cobertura estrutural com base no nível de importância do *software* testado.

Outro exemplo de norma de área específica é o *Title 21 CFR Part 820* da Administração de Alimentos e Medicamentos dos EUA [FDA21]. Esta norma recomenda certas técnicas de testes estruturais e funcionais. A norma também recomenda estratégias e princípios de teste que conferem com os *syllabus* do ISTQB.

Em alguns casos, os testes são influenciados por normas ou metodologias generalizadas que não se preocupam com os testes, mas que afetam bastante o contexto dos processos de *software* em

que os testes ocorrem. Um exemplo é a estrutura de aprimoramento de processos de *software* CMMI®. Ela inclui duas áreas de processos principais, a verificação e a validação, que são interpretadas frequentemente como níveis de teste (como o teste de sistema e o de aceite, respectivamente). Possui implicações em termos da estratégia de teste e, frequentemente, segundo as interpretações mais comuns, exige que os testes baseados em requisitos analíticos sejam incluídos na estratégia de teste.

Três outros exemplos importantes são o PMBoK do PMI, o PRINCE2® e a ITIL®. O PMI e o PRINCE2 são estruturas de coordenação de projetos muito utilizadas na América do Norte e na Europa, respectivamente. A ITIL é uma estrutura para garantir que o grupo de TI preste serviços valiosos à organização em que existe. A terminologia e as atividades especificadas em tais estruturas diferem bastante dos *syllabus* e do glossário do ISTQB.

Ao trabalhar em uma organização que utiliza o PMBoK do PMI, o PRINCE2 e / ou a ITIL, o Gerente de Teste deve compreender as estruturas escolhidas, sua implementação e sua terminologia suficientemente bem para atuar com eficácia em tal contexto.

Independentemente das normas ou das metodologias adotadas, é importante se lembrar de que são criadas por grupos de profissionais. As normas refletem a experiência e o conhecimento coletivos do grupo-fonte, mas também seus pontos fracos. O Gerente de Teste deve estar ciente das normas que são válidas para seu ambiente e seu contexto, independentemente de serem normas formais (internacionais, nacionais ou de área específica) ou normas internas e práticas recomendadas.

Ao levar em consideração o uso de várias normas, tenha em mente que algumas normas não conferem com outras ou até mesmo fornecem conceitos conflitantes. O Gerente de Teste deve determinar a utilidade das diferentes normas para o contexto específico em que os testes ocorrem. As informações declaradas em uma norma podem ser úteis para um projeto ou podem atrapalhá-lo. No entanto, as normas podem servir de referência para boas práticas consagradas e de base para a organização do processo de teste.

Em alguns casos, a conformidade com as normas é exigida e possui implicações para os testes. O Gerente de Teste deve estar ciente de qualquer exigência de adesão às normas e garantir a continuidade da complacência adequada.

3 Revisões [180 min]

Palavras-chave

acompanhamento, auditoria, inspeção, moderador, plano de revisão, revisão de gerenciamento, revisão informal, revisão técnica, revisão, revisor.

Objetivos de aprendizagem

3.2 Revisões de gerenciamento e auditorias

TM-3.2.1 (K2) Entender as características principais de revisões de gerenciamento e auditorias.

3.3 Gerenciamento de revisões

TM-3.3.1 (K4) Analisar um projeto para selecionar o tipo de revisão adequado e definir um plano para a realização de revisões a fim de garantir execução, acompanhamento e prestação de contas adequados;

TM-3.3.2 (K2) Compreender os fatores, as habilidades e o tempo necessário para a participação em revisões.

3.4 Métricas de revisões

TM-3.4.1 (K3) Definir as métricas dos processos e produtos que serão utilizadas nas revisões.

3.5 Gerenciamento de revisões formais

TM-3.5.1 (K2) Explicar e dar exemplos das características das revisões formais.

3.1 Introdução

As revisões foram apresentadas no *syllabus* do nível fundamental do ISTQB como atividades de testes estáticos com produtos. As auditorias e as revisões de gerenciamento se concentram mais no processo de *software* do que nos produtos de trabalho de *software*.

Como as revisões são uma forma de teste estático, o Gerente de Teste pode ser responsável por seu sucesso geral, particularmente em relação aos produtos de *testware*. No contexto mais geral de projetos de *software*, contudo, esta responsabilidade deve ser questão de política organizacional. Considerando a possível aplicação generalizada de revisões formais em muitas disciplinas, antes e durante projetos de *software*, a parte responsável pode ser um Gerente de Teste, um Gerente de Qualidade ou um Coordenador de Revisão capacitado. Neste *syllabus*, o responsável (quem quer que seja) é denominado líder de revisão.

O líder de revisão deve garantir um ambiente que leve à implementação dos fatores de sucesso definidos no *syllabus* do nível fundamental do ISTQB. Além disso, o líder de revisão deve conceber um plano de medição para garantir que as revisões gerem valor eficaz.

Como os testadores conhecem bem o comportamento operacional e as características obrigatórias do sistema de *software*, o envolvimento do testador no processo de revisão é importante.

Os participantes de revisões devem contar com treinamento em revisão para compreenderem melhor suas funções respectivas em qualquer processo de revisão. Todos os participantes de revisões devem estar comprometidos com os benefícios de uma revisão bem-feita.

Quando bem-feitas, as revisões são a maior e mais rentável contribuição para a qualidade geral proporcionada. Assim, é sumamente importante que o líder de revisão consiga implementar revisões eficientes em seus projetos e demonstrar os benefícios de tais revisões.

Entre as possíveis revisões em um projeto estão:

- Revisões contratuais, iniciadas no começo do projeto e em marcos importantes do projeto;
- Revisões de requisitos, iniciadas quando os requisitos estão disponíveis para revisão, que idealmente cobrem os requisitos funcionais e não funcionais;
- Revisões de modelagem de alto nível, iniciadas quando a modelagem arquitetônica está disponível para revisão;
- Revisões de modelagem detalhada, iniciadas quando a modelagem detalhada está disponível para revisão;
- Revisões de código, realizadas quando os módulos individuais de *software* são criados, o que pode incluir os testes de unidade, os resultados e o próprio código;
- Revisões de produtos de trabalho de testes, que podem cobrir o/s plano/s de teste, as condições de teste, os resultados das análises de riscos de qualidade, os testes, os dados dos testes, os ambientes de teste e os resultados dos testes;

- Revisões de entradas (prontidão de teste) e saídas de teste de cada nível de teste, que, respectivamente, verificam os critérios de entrada de teste antes de dar início à execução de testes e os critérios de saída de teste antes da conclusão dos testes;
- Revisões de aceite, utilizadas para obter a aprovação de clientes ou *stakeholders* em um sistema.

Além de aplicar diversos tipos de revisão em um produto, é importante que o líder de revisão se lembre-se de que, embora as revisões consigam detectar defeitos em documentos estáticos, as revisões devem crescer com outras formas de testes estáticos (como análises estáticas) e testes dinâmicos do código. A combinação destas técnicas melhora a cobertura de testes e localizará mais defeitos.

Técnicas diferentes possuem focos diferentes. Por exemplo, uma revisão pode eliminar um problema no nível de requisitos antes da implementação do problema no código. A análise estática pode ajudar a aplicar as normas de codificação e verificar os problemas que a equipe pode achar difícil de detectar através do exame dos produtos de trabalho. As inspeções não só levam à descoberta e à eliminação de defeitos, mas também treinam os autores para evitar a criação de defeitos em seus produtos de trabalho.

O *syllabus* do nível fundamental do ISTQB apresentou os seguintes tipos de revisões:

- A revisão informal;
- O acompanhamento;
- A revisão técnica;
- A inspeção.

Além destas revisões, o Gerente de Teste também pode realizar:

- Revisões de gerenciamento;
- Auditorias.

3.2 Revisões de gerenciamento e auditorias

As revisões de gerenciamento são utilizadas para monitorar o andamento, avaliar a situação e tomar decisões sobre ações futuras. As revisões apoiam decisões sobre o futuro do projeto, como a adaptação ao nível de recursos, a implementação de ações corretivas ou a alteração do escopo do projeto.

Seguem as características principais das revisões de gerenciamento:

- São realizadas por ou para gestores com responsabilidade direta pelo projeto ou pelo sistema;
- São realizadas por ou para *stakeholders* ou responsáveis, por exemplo, um gestor ou diretor de alçada superior;

- Verificam a coerência com os planos e os desvios deles;
- Verificam a suficiência dos procedimentos de gerenciamento;
- Avaliam os riscos do projeto;
- Avaliam o impacto das ações e as formas de medir tais impactos;
- Produzem listagens de itens de ação, problemas a serem resolvidos e decisões tomadas.

As revisões de gerenciamento dos processos, como retrospectivas de projetos (isto é, aprendizados) são parte integrante das atividades de aprimoramento de processos.

O Gerente de Teste deve participar de revisões de gerenciamento do andamento dos testes e pode iniciá-las.

As auditorias costumam ser realizadas para demonstrar a conformidade com um conjunto definido de critérios, provavelmente uma norma aplicável, uma restrição regulamentar ou uma obrigação contratual. Assim, as auditorias servem para fazer uma avaliação independente da complacência com processos, regulamentos, normas *etc.* Seguem as características principais das auditorias:

- São realizadas e moderadas por um auditor-chefe;
- As provas da complacência são coletadas através de entrevistas, depoimentos e exames de documentos;
- Os resultados documentados incluem observações, recomendações, ações corretivas e avaliações de aprovação / reprovação.

3.3 Gerenciamento de revisões

As revisões devem ser planejadas para ocorrerem em intervalos ou marcos naturais no projeto de *software*. Normalmente, as revisões devem ser realizadas após as definições de requisitos e modelagens, sendo que as revisões relacionadas começam pelos objetivos de negócios e chegam ao nível mais baixo da modelagem. As revisões de gerenciamento devem acontecer em marcos importantes do projeto, frequentemente como parte de uma atividade de verificação antes, durante e depois da execução de testes e outras fases significativas do projeto. A estratégia de revisão deve ser coordenada com a política de teste e a estratégia geral de teste.

Antes da formulação do plano geral de revisão no nível do projeto, o líder de revisão (que pode ser um Gerente de Teste) deve levar em consideração:

- O que deve ser revisado (produto e processos);
- Quem deve participar de revisões específicas;
- Quais são os fatores de risco relevantes que devem ser cobertos.

No início da fase de planejamento do projeto, o líder de revisão deve identificar os itens que devem ser revisados e selecionar o tipo adequado de revisão (revisão informal, acompanhamento, revisão técnica ou inspeção ou uma mistura de dois ou três tipos) e o nível de formalidade. Então, uma maior capacitação em revisão poderia ser recomendada. A partir daí o orçamento (tempo e recursos) pode

ser alocado ao processo de revisão. A determinação do orçamento deve incluir uma avaliação de risco e um cálculo do retorno sobre o investimento.

O retorno sobre o investimento das revisões é a diferença entre o custo de realização da revisão e o custo de lidar com os mesmos defeitos em uma etapa posterior (ou ignorá-los totalmente) se a revisão não houvesse sido realizada. O cálculo do custo de qualidade, explicado no item 2.7, pode ser utilizado para ajudar a determinar este número.

A determinação do período ideal para a realização de revisões depende do seguinte:

- A disponibilidade dos itens a serem revisados em um formato suficientemente definitivo;
- A disponibilidade do pessoal adequado para a revisão;
- O momento em que a versão final do item deve estar disponível;
- O tempo necessário para o processo de revisão daquele item específico.

As métricas adequadas da avaliação de revisões devem ser definidas pelo líder de revisão durante o planejamento de testes. Se as inspeções forem utilizadas, breves inspeções devem ser realizadas a pedido do autor enquanto a documentação é concluída (p. ex., requisitos ou seções individuais).

Os objetivos do processo de revisão devem ser definidos durante o planejamento de testes. Isto inclui a realização de revisões eficazes e eficientes e decisões consensuais a respeito do *feedback* das revisões.

As revisões de projetos são frequentemente realizadas no sistema geral e subsistemas e até mesmo elementos de *software* individuais podem ser necessários também. O número de revisões, o tipo das revisões, a organização das revisões e as pessoas envolvidas dependem do porte e da complexidade do projeto e dos riscos de produto.

Para serem eficientes, os participantes das revisões devem possuir níveis adequados de conhecimento, tanto técnico quanto processual. Para a realização de revisões eficazes, a meticulosidade e o detalhismo são algumas das habilidades necessárias para os revisores. A clareza e a priorização correta são atributos que devem ser examinados em bons comentários de revisões. A necessidade de conhecimento processual pode significar que há necessidade de treinamento para garantir que os revisores entendam suas funções e suas responsabilidades no processo de revisão.

O planejamento de revisões deve lidar com os riscos ligados a fatores técnicos, fatores organizacionais e problemas de pessoal na realização de revisões. A disponibilidade de revisores com conhecimento técnico suficiente é crucial para uma revisão bem-sucedida. Todas as equipes no projeto devem se envolver no planejamento das revisões, que devem garantir que cada equipe se comprometa com o sucesso do processo de revisão. O planejamento deve garantir que cada organização aloque tempo suficiente para que os revisores se preparem para as revisões e participem delas em certos momentos do cronograma do projeto. Também deve haver planejamento de tempo para qualquer treinamento técnico ou processual necessário para os revisores. Os revisores suplentes devem ser identificados caso os revisores principais não estejam disponíveis devido a mudanças na equipe ou planos empresariais.

Durante a execução real das revisões formais, o líder de revisão deve garantir que:

- Medições suficientes sejam fornecidas pelos participantes das revisões para possibilitar a avaliação da eficiência da revisão;
- *Checklists* sejam criadas e mantidas para o aprimoramento de revisões futuras;
- Avaliações da gravidade e da prioridade de defeitos sejam definidas para a utilização no gerenciamento de defeitos de problemas detectados durante revisões (ver o capítulo 4).

Após cada revisão, o líder de revisão deve:

- Coletar métricas de revisão e garantir que os problemas detectados sejam suficientemente resolvidos para atender aos objetivos de teste específicos da revisão;
- Utilizar as métricas de revisão como entradas na hora de determinar o retorno sobre o investimento (ROI, na sigla em inglês) das revisões;
- Fornecer *feedback* aos *stakeholders*;
- Fornecer *feedback* aos participantes das revisões.

Para avaliar a eficácia das revisões, o Gerente de Teste pode comparar os resultados reais detectados nos testes seguintes (isto é, após as revisões) com os resultados dos relatórios das revisões. Quando um produto de trabalho for revisado, aprovado com base na revisão, mas declarado defeituoso posteriormente, o líder de revisão deve descobrir como o processo de revisão pode ter deixado que os defeitos fossem ignorados. Entre as causas prováveis estão problemas no processo de revisão (p. ex., critérios de entrada ou saída ruins), a formação inadequada da equipe de revisão, ferramentas insuficientes de revisão (*checklists etc.*), capacitação e experiência insuficientes dos revisores e tempo insuficiente para preparativos e reuniões de revisão.

O padrão de ignorar defeitos (especialmente defeitos graves), repetido em vários projetos, indica que existem problemas consideráveis na realização das revisões. Em tal caso, o líder de revisão precisa revisar o processo e tomar as providências cabíveis. Também é possível que, por diversos motivos, as revisões percam a eficácia com o passar do tempo. Tal efeito será revelado em retrospectivas de projetos através de uma menor eficácia das revisões na detecção de defeitos. Então, como já foi dito, o líder de revisão deve investigar e corrigir as causas. Em todo caso, as métricas das revisões não devem ser utilizadas para punir ou recompensar revisores ou autores específicos. Pelo contrário, devem se concentrar no processo de revisão em si.

3.4 Métricas de revisões

O líder de revisão (que, como já foi afirmado em seções anteriores, podem ser o Gerente de Teste) deve garantir que as métricas estejam disponíveis para:

- Avaliar a qualidade do item revisado;
- Avaliar o custo da realização de revisões;
- Avaliar o benefício a jusante da realização de revisões.

O líder de revisão pode usar as medições para determinar o retorno sobre o investimento e a eficiência das revisões. Estas métricas também podem ser utilizadas para as atividades de divulgação e aprimoramento de processos.

Para cada produto de trabalho revisado, as seguintes métricas podem ser medidas e divulgadas para a avaliação de produtos:

- O tamanho do produto de trabalho (páginas, linhas de código *etc.*);
- O tempo de preparação (antes da revisão);
- O tempo de realização de revisões;
- O tempo de retrabalho para a correção de defeitos;
- A duração do processo de revisão;
- O número de defeitos detectados e sua gravidade;
- A identificação de núcleos de defeitos no produto de trabalho (isto é, áreas que possuem uma densidade maior de defeitos);
- O tipo de revisão (revisão informal, acompanhamento, revisão técnica ou inspeção);
- A densidade média de defeitos (p. ex., defeitos por página ou por mil linhas de código);
- Os defeitos residuais estimados (ou a densidade de defeitos residuais).

Em cada revisão, as seguintes métricas podem ser medidas e divulgadas para a avaliação de processos:

- A eficácia na detecção de defeitos (levar em conta os defeitos detectados em um momento posterior do ciclo de vida);
- O aprimoramento da execução de processos de revisão e do momento em que são realizados;
- A porcentagem de cobertura de produtos de trabalho planejados;
- Os tipos de defeitos detectados e sua gravidade;
- As pesquisas dos participantes sobre a eficácia e a eficiência do processo de revisão;
- O custo das métricas de qualidade de defeitos de revisões, defeitos de testes dinâmicos e defeitos de produção;
- A correlação da eficácia das revisões (comparação entre tipo de revisão e eficácia na detecção de defeitos);
- Número de revisores;
- Defeitos detectados por hora trabalhada;
- Estimativa de tempo de projeto economizado;
- Média de detecção de defeitos (isto é, a detecção total e o tempo de correção dividido pelo número de defeitos).

Além disso, as métricas mencionadas na avaliação de produtos acima também são úteis na avaliação de processos.

3.5 Gerenciamento de revisões formais

O *syllabus* do nível fundamental do ISTQB descreve as diferentes fases de uma revisão formal: planejamento, início, preparativos individuais, reunião de revisão, retrabalho e acompanhamento. Para a implementação correta das revisões formais, os líderes de revisão precisam garantir que todas as etapas do processo de revisão sejam seguidas:

As revisões formais possuem uma série de características, como:

- Critérios de entrada e de saída definidos;
- *Checklists* que são utilizadas pelos revisores;
- Entregáveis, como relatórios, fichas de avaliação ou outros resumos das revisões;
- Métricas para a divulgação da eficácia, da eficiência e do andamento das revisões.

Antes de dar início a uma revisão formal, o cumprimento dos pré-requisitos da revisão (definidos no procedimento ou na lista de critérios de entrada) deve ser confirmado pelo líder de revisão.

Se as condições obrigatórias da revisão formal não forem atendidas, o líder de revisão pode propor uma das seguintes alternativas à autoridade de revisão para que tome uma decisão definitiva:

- A redefinição da revisão com um ajuste dos objetivos;
- As ações corretivas necessárias para que a revisão continue;
- O adiamento da revisão.

Para o controle das revisões formais, elas são monitoradas no contexto do programa geral (de nível mais alto) e são relacionadas às atividades de garantia de qualidade do projeto. O controle das revisões formais inclui o *feedback* com as métricas dos produtos e dos processos.

4 Gerenciamento de defeitos [150 min]

Palavras-chave

anomalia, causa-raiz, comitê de triagem de defeitos, contenção de fases, defeito, falso resultado negativo, falso resultado positivo, gravidade, prioridade.

Objetivos de aprendizagem

4.2 O ciclo de vida dos defeitos e o ciclo de vida do desenvolvimento de software

TM-4.2.1 (K3) Desenvolver um processo de gerenciamento de defeitos para uma organização de teste, inclusive o fluxo de trabalho dos relatórios de defeitos, que possa ser utilizado no monitoramento e no controle dos defeitos de um projeto ao longo do ciclo de vida de teste;

TM-4.2.2 (K2) Explicar os processos e os participantes necessários para o gerenciamento eficaz dos defeitos.

4.3 Informações de relatórios de defeitos

TM-4.3.1 (K3) Definir os dados e as informações classificatórias que devem ser coletadas durante o processo de gerenciamento de defeitos.

4.4 Avaliação de recursos de processos com informações de relatórios de defeitos

TM-4.4.1 (K2) Explicar como as estatísticas de relatórios de defeitos podem ser utilizadas para a avaliação dos recursos dos processos de teste e de desenvolvimento de software.

4.1 Introdução

O processo de gerenciamento de defeitos de uma organização e a ferramenta utilizada para gerenciar estes trabalhos são sumamente importantes não só para a equipe de teste, mas também para todas as equipes envolvidas no desenvolvimento de *software*. As informações coletadas pelo gerenciamento eficaz de defeitos permitem que o Gerente de Teste e outros *stakeholders* de projetos conheçam a situação de um projeto ao longo do ciclo de vida de desenvolvimento e, ao coletar e analisar os dados com o tempo, podem facilitar a localização de pontos de possível melhoria nos processos de teste e desenvolvimento.

Além de entender o ciclo de vida geral dos defeitos e como ele é utilizado para monitorar e controlar os processos de teste e desenvolvimento de *software*, o Gerente de Teste também deve conhecer os dados que devem ser captados e defender a utilização adequada do processo e da ferramenta escolhida de gerenciamento de defeitos.

4.2 O ciclo de vida dos defeitos e o ciclo de vida do desenvolvimento de software

Como já se explicou no *syllabus* do nível fundamental, os defeitos surgem quando uma pessoa comete um erro durante a criação de um produto de trabalho. O produto de trabalho pode ser uma especificação de requisitos, uma história de usuário, um documento técnico, um caso de teste, o código de um programa ou qualquer outro produto de trabalho gerado durante o processo de desenvolvimento ou manutenção de *software*.

Os defeitos podem surgir em qualquer momento do ciclo de vida de desenvolvimento de *software* e em qualquer produto de trabalho relacionado ao *software*. Portanto, cada fase do ciclo de vida de desenvolvimento de *software* deve incluir atividades para a detecção e a eliminação de possíveis defeitos. Por exemplo, as técnicas de testes estáticos (isto é, revisões e análises estáticas) podem ser utilizadas em especificações de modelagem, especificações de requisitos e no código antes de repassar tais produtos de trabalho às atividades seguintes.

Quanto antes o defeito for detectado e eliminado, menor será o custo total de qualidade do sistema. O custo de qualidade de determinado nível de defeitos é minimizado quando cada defeito for removido na mesma fase em que surgiu (isto é, quando o processo de *software* atinge a contenção perfeita de fases). Além disso, como se explicou no *syllabus* do nível fundamental, em vez de detectar falhas, os testes estáticos encontram defeitos diretamente e, assim, o custo da eliminação de defeitos é menor porque as atividades de depuração não são necessárias para isolar o defeito.

Durante as atividades de teste dinâmico, como os testes de unidade, de integração e de sistema, a presença de um defeito é revelada quando provocar uma falha, o que resulta em uma discrepância

entre os resultados reais e os resultados esperados de um teste (isto é, uma anomalia). Em alguns casos, um falso resultado negativo ocorre quando o testador não enxerga a anomalia. Se o testador não enxergar a anomalia, ocorreu uma situação que exige maiores investigações. A investigação começa com a apresentação de um relatório de defeitos.

No desenvolvimento orientado por testes, os testes de unidade automatizados servem de especificações de modelagem executáveis. À medida que o código é desenvolvido, é imediatamente exercido com estes testes. Alguns ou todos os testes falharão até a conclusão do desenvolvimento da unidade. Portanto, a falha de tal teste não constitui um defeito e, normalmente, não é rastreada.

4.2.1 Fluxo de trabalho e estados de defeitos

A maioria das organizações de teste utiliza uma ferramenta para gerenciar relatórios de defeitos através do ciclo de vida de defeitos. Um relatório de defeitos normalmente progride com um fluxo de trabalho e atravessa uma sequência de estados à medida que passa por um ciclo de vida de defeitos. Na maioria destes estados, um único participante do ciclo de vida de defeitos é responsável pelo relatório e pela realização de uma tarefa que, quando concluída, fará que o relatório de defeitos passe ao estado seguinte (e seja atribuído à parte responsável seguinte).

Em estados terminais, como quando um relatório de defeitos é encerrado (o que, normalmente, significa que o defeito subjacente foi corrigido e a correção foi verificada através de um teste de confirmação), cancelado (o que, normalmente, significa que o relatório de defeitos é inválido), impossível de reproduzir (o que, normalmente, significa que a anomalia não pode mais ser observada) ou repassado (o que, normalmente, significa que a anomalia diz respeito a um defeito real, mas que o defeito não será corrigido durante o projeto), o relatório não possui um responsável, pois nenhuma outra ação é necessária.

Nos defeitos descobertos pelos testadores durante o teste, existem três estados em específico em que a ação cabe à equipe de teste:

O estado inicial:

- Neste estado, um ou mais testadores coletam as informações necessárias para o responsável pela resolução reproduzir o defeito (ver o item 4.3 para obter mais informações que serão incluídas no relatório de defeitos);
- Este estado também pode ser denominado aberto ou novo.

O estado de retorno:

- Neste estado, o destinatário do relatório recusou o relatório ou pede mais informações ao testador. Este estado pode indicar uma deficiência no processo inicial de coleta de informações ou no próprio teste e o Gerente de Teste deve monitorar as taxas excessivas de retorno. O testador deve fornecer outras informações ou confirmar que o relatório realmente deve ser recusado;

- Este estado também pode ser denominado recusado ou à espera de esclarecimento.

O estado do teste de confirmação:

- Neste estado, o testador executará um teste de confirmação (frequentemente seguindo as etapas de reprodução da falha do próprio relatório de defeitos) para determinar se a correção realmente solucionou o problema. Se o teste de confirmação indicar que o defeito foi corrigido, o testador deve encerrar o relatório. Se o teste de confirmação indicar que o defeito não foi corrigido, o testador deve reabrir o relatório, fazendo que seja novamente atribuído ao responsável anterior, o qual, então, pode concluir os trabalhos necessários para a correção do defeito;
- Este estado também pode ser denominado resolvido ou em verificação.

4.2.2 Gerenciamento de relatórios de defeitos inválidos e em duplicidade

Em alguns casos, ocorre uma anomalia não como sintoma de um defeito e, sim, devido a um problema com o ambiente de teste, os dados dos testes, algum outro elemento do *testware* ou um mal-entendido do testador. Se o testador elaborar um relatório de defeitos que, depois, não disser respeito a um defeito no produto de trabalho testado, trata-se de um falso resultado positivo. Tais relatórios são normalmente cancelados ou encerrados como relatórios de defeitos inválidos.

Além disso, em alguns casos, um defeito consegue apresentar sintomas diferentes, que, para o testador, podem ser totalmente desconexos. Se dois ou mais relatórios de defeitos forem apresentados e, depois, e eles disserem respeito à mesma causa-raiz, um dos relatórios de defeitos é normalmente retido, enquanto os outros são encerrados como relatórios de defeitos em duplicidade.

Embora os relatórios de defeitos inválidos e em duplicidade representem certo nível de ineficiência, relatórios assim são inevitáveis e o Gerente de Teste deve aceitá-los pelo que são. Quando os gestores tentam eliminar todos os relatórios de defeitos inválidos e em duplicidade, o número de falsos resultados negativos costuma aumentar, já que os testadores não têm mais o incentivo de apresentar relatórios de defeitos. Isto diminui a eficácia da organização de teste na detecção de defeitos, o que diz respeito ao objetivo principal da organização de teste na maioria dos casos.

4.2.3 Gerenciamento de defeitos interfuncionais

Embora a organização de teste e o Gerente de Teste costumem ser responsáveis pelo processo geral de gerenciamento de testes e pela ferramenta de gerenciamento de testes, uma equipe interfuncional é geralmente responsável pelo gerenciamento dos defeitos divulgados de determinado projeto. Além do Gerente de Teste, os integrantes do comitê de gerenciamento de defeitos (ou de triagem de defeitos) normalmente incluem os *stakeholders* de desenvolvimento,

gerenciamento de projetos, gerenciamento de produtos e outros com interesse no *software* em desenvolvimento.

À medida que as anomalias são descobertas e introduzidas na ferramenta de gerenciamento de defeitos, o comitê de gerenciamento de defeitos deve se reunir para determinar se cada relatório de defeitos representa um defeito válido e se deve ser corrigido ou repassado. A decisão exige que o comitê de gerenciamento de defeitos pondere os benefícios, os riscos e os custos ligados à correção ou à não correção do defeito. Se o defeito for corrigido, a equipe deve definir a prioridade de corrigir o defeito em relação a outras tarefas do projeto. O Gerente de Teste e a equipe de teste devem ser consultados em relação à importância relativa de um defeito e deve fornecer as informações objetivas disponíveis.

Uma ferramenta de rastreamento de defeitos não deve substituir a boa comunicação e as reuniões do comitê de gerenciamento de defeitos não devem substituir o uso eficaz de uma boa ferramenta de rastreamento de defeitos. A comunicação, o suporte adequado de ferramentas, um ciclo de vida de defeitos bem definido e um comitê de gerenciamento de defeitos engajado são necessários para o gerenciamento eficaz e eficiente de defeitos.

4.3 Informações de relatórios de defeitos

Quando um defeito for detectado (nos testes estáticos) ou uma falha for observada (nos testes dinâmicos), os dados devem ser coletados pela/s pessoa/s envolvida/s e incluída/s no relatório de defeitos. Estas informações devem bastar para três finalidades:

- O gerenciamento do relatório ao longo do ciclo de vida de defeitos;
- A avaliação da situação do projeto, especialmente em termos de qualidade do produto e do andamento do teste;
- A avaliação dos recursos do processo (como já se discutiu no item 4.4 abaixo).

Os dados necessários para o gerenciamento de relatórios de defeitos e a situação do projeto podem variar dependendo do momento em que o defeito é detectado no ciclo de vida, normalmente com menos informações necessárias antes (p. ex., revisões de requisitos e testes de unidade). No entanto, as informações principais coletadas devem conferir ao longo do ciclo de vida e, idealmente, em todos os projetos para permitir uma comparação significativa de dados de defeitos de processos ao longo do projeto e em todos os projetos.

A coleta de dados de defeitos pode facilitar o monitoramento do andamento dos testes, o controle e a avaliação de critérios de saída. Por exemplo, as informações dos defeitos devem apoiar a análise da densidade de defeitos, a análise de tendências de defeitos detectados e resolvidos, o tempo médio da detecção à resolução dos defeitos e a intensidade das falhas (p. ex., a análise de tempo médio entre falhas).

Os dados dos defeitos que serão coletados podem incluir:

- O nome da pessoa que descobriu o defeito;
- A função da pessoa (p. ex., usuário final, analista de negócios, desenvolvedor, funcionário de suporte técnico);
- O tipo de teste realizado (p. ex., os testes de usabilidade, de performance, de regressão);
- Um resumo do problema;
- Uma descrição detalhada do problema;
- As etapas para a reprodução da falha (de um defeito) e os resultados reais e esperados (com destaque para a anomalia), inclusive capturas de tela, *dumps* de bancos de dados e registros, se for o caso;
- A fase do ciclo de vida de introdução, detecção e eliminação do defeito, inclusive o nível de teste, se for o caso;
- O produto de trabalho em que o defeito surgiu;
- A gravidade do impacto sobre o sistema e / ou os *stakeholders* do produto (normalmente determinada pelo comportamento técnico do sistema);
- A prioridade de corrigir o problema (normalmente determinada pelo impacto comercial da falha);
- O subsistema ou o componente em que o defeito existe (referente à análise de núcleos de defeitos);
- A atividade do projeto ocorre quando o problema for detectado;
- O método de identificação que revelou o problema (p. ex., revisão, análise estática, teste dinâmico, uso de produção);
- O tipo de defeito (o que normalmente corresponde à taxonomia de defeitos, quando ela for utilizada);
- A característica de qualidade afetada pelo defeito;
- O ambiente de teste em que o defeito foi observado (no tocante a testes dinâmicos);
- O projeto e o produto em que o problema existe;
- O responsável atual, isto é, a pessoa que, atualmente, recebeu a incumbência de resolver o problema, partindo da premissa de que o relatório não é definitivo;
- A situação atual do relatório (geralmente gerenciada pela ferramenta de rastreamento de defeitos como parte do ciclo de vida);
- Os produtos de trabalho específicos (p. ex., itens de teste e seus números de lançamento) em que o problema foi observado e os produtos de trabalho específicos em que o problema foi, enfim, resolvido;
- O impacto sobre os interesses dos *stakeholders* do projeto e do produto;
- As conclusões, as recomendações e as aprovações referentes às providências tomadas ou não tomadas para a resolução do problema;
- Os riscos, os custos, as oportunidades e os benefícios ligados à correção ou à não correção do defeito;

- As datas em que as diversas transições do ciclo de vida de defeitos ocorreram, os responsáveis do relatório com base em cada transição e as providências tomadas pelos integrantes da equipe do projeto para isolar, reparar e verificar a correção do defeito;
- A descrição de como o defeito foi resolvido e as recomendações para testar a correção (se o defeito foi resolvido por uma alteração no *software*);
- Outras referências, como o teste que revelou o defeito e o risco, o requisito ou outro elemento da base de teste relacionado ao defeito (em testes dinâmicos).

Diversas normas e documentos, como a ISO 9126 [ISO9126] (substituída pela ISO 25000), a IEEE 829 [IEEE829], a IEEE 1044 [IEEE1044] e a classificação de defeitos ortogonais, existem para ajudar o Gerente de Teste a determinar quais informações devem ser coletadas para a divulgação dos defeitos.

Independentemente das informações específicas consideradas necessárias para os relatórios de defeitos, é sumamente importante que o testador apresente informações completas, concisas, exatas, objetivas, relevantes e oportunas. Mesmo quando a intervenção manual e a comunicação em pessoa superarem os problemas com dados de relatórios de defeitos em termos de resolução de defeitos individuais, os problemas com dados de relatórios de defeitos podem apresentar obstáculos insuperáveis para a avaliação adequada da situação dos projetos, do andamento dos testes e dos recursos dos processos.

4.4 Avaliação de recursos de processos com informações de relatórios de defeitos

Como já foi discutido no capítulo 2, os relatórios de defeitos podem ser úteis para o monitoramento e a divulgação da situação dos projetos. Embora as implicações das métricas para os processos sejam principalmente tratadas no *Expert Test Management Syllabus*, no nível avançado, o Gerente de Teste deve saber o que os relatórios de defeitos significam em termos de avaliar os recursos dos processos de teste e de desenvolvimento de *software*. Além das informações de monitoramento do andamento dos testes mencionadas no capítulo 2 e no item 4.3, as informações de defeitos precisam apoiar as iniciativas de aprimoramento de processos. Entre os exemplos estão:

- A utilização de informações das fases de introdução, detecção e eliminação de fase em fase para avaliar a contenção de fases e sugerir formas de aprimorar a eficácia da detecção de defeitos em cada fase;
- A utilização das informações da fase de introdução para a análise de Pareto das fases em que o maior número de defeitos foi introduzido para possibilitar aprimoramentos direcionados a fim de reduzir o número total de defeitos;
- A utilização das informações das causas-raiz de defeitos para determinar os motivos subjacentes da introdução de defeitos a fim de possibilitar o aprimoramento de processos com a finalidade de reduzir o número total de defeitos;

- A utilização das informações das fases de introdução, detecção e eliminação para realizar uma análise de custo de qualidade a fim de minimizar o custo ligado aos defeitos;
- A utilização das informações de componentes com defeitos para realizar uma análise de núcleos de defeitos a fim de compreender melhor os riscos técnicos (nos testes baseados em risco) e possibilitar a reengenharia de componentes problemáticos.

A utilização de métricas para avaliar a eficácia e a eficiência do processo de teste é discutida no *Expert Test Management Syllabus*.

Em alguns casos, as equipes podem escolher não rastrear os defeitos encontrados durante parte do ciclo de vida de desenvolvimento de *software* ou o ciclo de vida de desenvolvimento de *software* inteiro. Embora isto seja frequentemente realizado em nome da eficiência e para reduzir os custos dos processos, na realidade, isto reduz a visibilidade nos recursos dos processos de teste e desenvolvimento de *software*. Isto dificulta a realização das melhorias sugeridas acima devido à falta de dados confiáveis.

5 Aprimoramento de processos de teste [135 min]

Palavras-chave

Capability Maturity Model Integration (CMMI), Critical Testing Processes (CTP), Test Maturity Model integration (TMMi), Test Process Improvement (TPI), Software Test and Evaluation Process (STEP)

Objetivos de aprendizagem

5.2 Processo de aprimoramento de testes

TM-5.2.1 (K2) Explicar, com exemplos, por que é importante aprimorar o processo de teste.

5.3 Aprimoramento de processos de teste

TM-5.3.1 (K3) Definir um plano de aprimoramento de processos de teste com o modelo IDEAL.

5.4 Aprimoramento de processos de teste com TMMi

TM-5.4.1 (K2) Resumir os antecedentes, o escopo e os objetivos do modelo de aprimoramento de processos de teste TMMi.

5.5 Aprimoramento de processos de teste com TPI Next

TM-5.5.1 (K2) Resumir os antecedentes, o escopo e os objetivos do modelo de aprimoramento de processos de teste TPI Next.

5.6 Aprimoramento de processos de teste com CTP

TM-5.6.1 (K2) Resumir os antecedentes, o escopo e os objetivos do modelo de aprimoramento de processos de teste CTP.

5.7 Aprimoramento de processos de teste com STEP

TM-5.7.1 (K2) Resumir os antecedentes, o escopo e os objetivos do modelo de aprimoramento de processos de teste STEP.

5.1 Introdução

Assim que for definido, o processo geral de teste de uma organização deve passar por melhorias contínuas. Neste capítulo, problemas genéricos de aprimoramento são discutidos. Depois, segue uma apresentação a alguns modelos específicos que podem ser utilizados no aprimoramento de processos de teste. O Gerente de Teste deve supor que será a força motriz por trás das alterações e das melhorias nos processos de teste e, assim, deve conhecer as técnicas do setor discutidas neste capítulo. Outras informações sobre os processos de aprimoramento de testes são discutidas no *Expert Improving the Test Process Syllabus*.

5.2 Processo de aprimoramento de testes

Assim como as organizações usam o teste para aprimorar o *software*, as técnicas de aprimoramento de processos podem ser escolhidas e utilizadas para melhorar o processo de desenvolvimento de *software* e os entregáveis de *software* resultantes. O aprimoramento de processos também pode ser aplicado aos processos de teste. Existem modos e meios diferentes de melhorar o teste de *software* e de sistemas com *software*. Tais métodos buscam melhorar o processo e, assim os entregáveis fornecem diretrizes e pontos de melhoria.

Frequentemente, o teste é responsável por grande parte dos custos totais do projeto. No entanto, apenas alguns cuidados são tomados com o processo de teste nos diversos modelos de aprimoramento de processos de *software*, como a CMMI® (seguem detalhes).

Modelos de aprimoramento de testes, como a integração de modelos de maturidade em teste (TMMi®), o processo sistemático de teste e avaliação (STEP), os processos críticos de teste (CTP) e a TPI Next®, foram desenvolvidos para lidar com a falta de atenção com o teste na maioria dos modelos de aprimoramento de processos de *software*. Quando adequadamente utilizados, estes modelos podem gerar métricas que podem ser utilizadas entre organizações para fins de comparação.

Os modelos apresentados neste *syllabus* não pretendem ser uma recomendação de uso. São apresentados para mostrar como os modelos funcionam e o que eles incluem.

5.2.1 Introdução ao aprimoramento de processos

As melhorias de processos são relevantes para o processo de desenvolvimento de *software* e de teste. Aprender com os próprios erros possibilita o aprimoramento dos processos que as organizações utilizam para desenvolver e testar *software*. O ciclo de melhoria de Deming (planejar, executar, verificar, atuar) vem sendo utilizado há décadas e ainda é relevante quando o testador precisa aprimorar o processo utilizado hoje.

Uma hipótese do aprimoramento de processos é a crença de que a qualidade do sistema é altamente influenciada pela qualidade do processo utilizado para desenvolver o *software*. O aprimoramento da qualidade no setor de *software* reduz a necessidade de recursos para manter o *software* e, assim, gera mais tempo para a criação de outras soluções melhores no futuro. Os modelos dos processos criam um ponto de partida para o aprimoramento ao comparar os recursos dos processos da organização

com o modelo. Os modelos também formam uma estrutura para o aprimoramento dos processos da organização com base nos resultados de uma avaliação.

A avaliação de processos leva à determinação de recursos de processos, o que motiva o aprimoramento de processos. Isto pode provocar uma avaliação de processos posterior para medir o efeito da melhoria.

5.2.2 Tipos de aprimoramento de processos

A utilização de modelos de avaliação é um método comum que garante uma abordagem padronizado ao aprimoramento de processos de teste com práticas consagradas.

Os modelos de aprimoramento de processos são divididos em duas categorias:

1. O modelo de referências de processos, que gera medidas de maturidade nos termos da avaliação a fim de avaliar os recursos de uma organização em comparação com o modelo, avaliar a organização dentro da estrutura e criar um roteiro para o aprimoramento do processo;
2. O modelo de referências de conteúdo, que faz avaliações orientadas para negócios das oportunidades de aprimoramento de uma organização, inclusive, em alguns casos, de *benchmarking* com as médias do setor com medições objetivas. Esta avaliação pode ser utilizada para criar um roteiro para o aprimoramento de processos.

O aprimoramento de processos de teste também pode ser alcançado sem modelos através da utilização de abordagens analíticas e reuniões retrospectivas, por exemplo.

5.3 Aprimoramento de processos de teste

O setor de TI pode funcionar com modelos de aprimoramento de processos de teste para alcançar níveis maiores de maturidade e profissionalismo. Os modelos de normas do setor contribuem para o desenvolvimento de métricas e medidas entre organizações que podem ser utilizadas para comparação. Da necessidade de aprimoramento de processos no setor de teste, vários conjuntos de processos recomendados foram concretizados. Entre eles estão o STEP, a TMMi, a TPI Next e o CTP.

Os modelos por estágios, como a TMMi e a CMMI, criam normas para a comparação entre empresas e organizações diferentes. Os modelos contínuos, como o CTP, o STEP e a TPI Next, permitem que uma organização lide com os problemas de mais alta prioridade com mais liberdade na ordem de implementação. Eles são discutidos em um momento posterior desta seção.

Todos os modelos permitem que a organização determine a sua posição em termos de processos de teste atuais. Assim que a avaliação for realizada, a TMMi e a TPI Next indicam um roteiro para o aprimoramento dos processos de teste. Em contrapartida, o STEP e o CTP propiciam à organização meios para determinar de onde virá o maior retorno sobre o investimento de aprimorar os processos e permitem que a organização escolha o roteiro adequado.

Assim que ficar acordado que os processos de teste devem ser revisados e aprimorados, as etapas de implementação de aprimoramento de processos a serem adotadas para esta atividade podem ser definidas no modelo IDEALSM [IDEAL96].

- A iniciação do processo de aprimoramento;
- O diagnóstico da situação atual;
- A definição de um plano de aprimoramento de processos de teste;
- A ação para a implementação do aprimoramento;
- O aprendizado com o programa de aprimoramento.

Iniciação do processo de aprimoramento

Antes do início das atividades de aprimoramento de processos, os objetivos, as metas, o escopo e a cobertura das melhorias nos processos são acordados pelos *stakeholders*. A escolha do modelo de aprimoramento de processos também é feita neste momento.

O modelo pode ser selecionado a partir de opções publicamente disponíveis (como CTP, STEP, TMMi e TPI NExt) ou desenvolvido internamente. Além disso, critérios de sucesso devem ser definidos e um método pelo qual serão medidos ao longo das atividades de aprimoramento deve ser determinado.

Diagnóstico da situação atual

A abordagem acordada à avaliação é realizada e um relatório de avaliação de testes é criado e contém uma avaliação das práticas de teste atuais e uma lista de possíveis melhorias de processos.

Definição de um plano de aprimoramento de processos de teste

A lista de possíveis melhorias de processos é priorizada. A priorização pode se basear no retorno sobre o investimento, nos riscos, no alinhamento com a estratégia organizacional e / ou nos benefícios quantitativos ou qualitativos mensuráveis. Após definir a ordem de prioridade, um plano para a entrega das melhorias é desenvolvido.

Ação para implementação do aprimoramento

O plano de aprimoramento dos processos de teste para a entrega de melhorias é implementado. Isto pode incluir a capacitação ou a mentoria necessária, o lançamento de processos e, enfim, sua implantação total.

Aprendizado com o programa de aprimoramento

Após ter implantado totalmente as melhorias nos processos, é essencial verificar quais benefícios (dos definidos antes, além de benefícios possivelmente inesperados) foram recebidos. Também é importante verificar quais dos critérios de sucesso da atividade de aprimoramento de processos foram atendidos.

Dependendo do modelo de processos utilizado, neste estágio do processo, o monitoramento do nível seguinte de maturidade começa e a decisão de recomençar o processo de aprimoramento ou de interromper a atividade naquele momento é tomada.

5.4 Aprimoramento de processos de teste com TMMi

A integração de modelos de maturidade em teste (TMMi) é composta por cinco níveis de maturidade e pretende complementar a CMMI. Cada um dos níveis de maturidade contém áreas definidas de processo que devem estar 85% completas ao atingir metas específicas e genéricas antes que a organização possa passar ao nível seguinte.

Os níveis de maturidade TMMi são:

Nível 1: Inicial

O nível inicial representa um estado em que não há processo de teste formalmente documentado ou estruturado. Normalmente, os testes são desenvolvidos *ad hoc* após a codificação e são equiparados à depuração. Entende-se que o objetivo do teste é o de provar que o *software* funciona.

Nível 2: Gerenciamento

O segundo nível é atingido quando os processos de teste são claramente separados da depuração. Podem ser alcançados ao definir a política e as metas de teste, introduzindo as etapas encontradas em um processo de teste fundamental (p. ex., o planejamento de testes) e implementando técnicas e métodos básicos de teste.

Nível 3: Definição

O terceiro nível é alcançado quando um processo de teste é integrado ao ciclo de vida de desenvolvimento de *software* e documentado em normas, procedimentos e métodos formais. As revisões ocorrem e deve existir uma função distinta de teste de *software* que possa ser controlada e monitorada.

Nível 4: Medição

O nível quatro é atingido quando o processo de teste consegue ser medido e gerenciado eficazmente na organização em favor de projetos específicos.

Nível 5: Otimização

O nível final representa um estado de maturidade de processos de teste em que os dados do processo de teste podem ser utilizados para ajudar a impedir defeitos e a ênfase é depositada na otimização do processo estabelecido. Para obter mais informações sobre a TMMi, ver [vanVeenendaal11] e [www.tmmi.org].

5.5 Aprimoramento de processos de teste com TPI Next

O modelo TPI Next define 16 áreas importantes e cada qual cobre um aspecto específico do processo de teste, como a estratégia de teste, as métricas, as ferramentas de teste e o ambiente de teste.

Quatro níveis de maturidade são definidos no modelo:

- Inicial;
- Controlado;
- Eficiente;
- Otimizado.

Checkpoints específicos são definidos para avaliar a área principal de cada um dos níveis de maturidade. As descobertas são resumidas e visualizadas através de uma matriz de maturidade que cobre todas as áreas importantes. A definição dos objetivos de aprimoramento e sua implementação podem ser ajustadas de acordo com as necessidades e a capacidade da organização de teste.

A abordagem genérica torna a TPI Next independente de qualquer modelo de aprimoramento de processos de *software*. Cobre os aspectos de engenharia de teste e apoia o processo decisório gerencial [deVries09]. Para obter mais informações sobre a TPI Next, ver [www.tpinext.com].

5.6 Aprimoramento de processos de teste com CTP

A premissa básica do modelo de avaliação dos processos críticos de teste (CTP) é a de que certos processos de teste são cruciais. Os processos críticos, se bem realizados, suportarão equipes de teste bem-sucedidas. Em contrapartida, se tais atividades forem mal realizadas, até mesmo testadores e Gerentes de Teste talentosos provavelmente não terão sucesso. O modelo identifica doze processos de teste críticos. O CTP é, sobretudo, um modelo de referência de conteúdo.

O modelo CTP é uma abordagem contextual que permite a adaptação do modelo, inclusive:

- A identificação de desafios específicos;
- O reconhecimento de atributos de bons processos;
- A seleção da ordem e da importância da implementação de melhorias nos processos.

O modelo CTP é adaptável no contexto de todos os modelos de ciclo de vida de desenvolvimento de *software*.

Além de entrevistas com participantes, o modelo CTP inclui a utilização de métricas para comparar as organizações com as médias e as boas práticas do setor. Para obter mais informações sobre o CTP, ver [Black03].

5.7 Aprimoramento de processos de teste com STEP

O processo sistemático de teste e a avaliação ou STEP, na sigla em inglês, à maneira do CTP e diferentemente da TMMI e da TPI Next, não exige que as melhorias aconteçam em uma ordem específica.

O STEP é, sobretudo, um modelo de referência de conteúdo que se baseia na ideia de que o teste é uma atividade de ciclo de vida que começa durante a elaboração de requisitos e continua até a desativação do sistema. A metodologia STEP enfatiza a prática de “testar, depois codificar” ao utilizar uma estratégia de teste baseada em requisitos para garantir que a criação antecipada de casos de testes valide as especificações de requisitos antes da modelagem e da codificação.

Entre as premissas básicas da metodologia estão:

- Uma estratégia de teste baseado em requisitos;
- O início do teste no começo do ciclo de vida;
- Os testes servem de modelos de requisitos e utilização;
- A modelagem de *testware* vem antes da modelagem de *software*;
- Os defeitos são detectados antes ou totalmente prevenidos;
- Os defeitos são sistematicamente analisados;
- Os testadores e os desenvolvedores trabalham juntos.

Em alguns casos, o modelo de avaliação STEP se mistura com o modelo de maturidade da TPI Next. Para obter mais informações sobre o STEP, ver [Craig02].

6 Ferramentas e automação de testes [135 min]

Palavras-chave

ferramenta de código aberto, ferramenta personalizada

Objetivos de aprendizagem

6.2 Seleção de ferramentas

TM-6.2.1 (K2) Descrever as questões de gerenciamento ao selecionar uma ferramenta de código aberto;

TM-6.2.2 (K2) Descrever as questões de gerenciamento ao escolher uma ferramenta personalizada;

TM-6.2.3 (K4) Avaliar determinada situação para conceber um plano para a seleção de ferramentas, inclusive riscos, custos e benefícios.

6.3 Ciclo de vida das ferramentas

TM-6.3.1 (K2) Explicar as diferentes fases do ciclo de vida de uma ferramenta.

6.4 Métricas das ferramentas

TM-6.4.1 (K2) Descrever como a coleta e a avaliação das métricas podem ser aprimoradas com a utilização de ferramentas.

6.1 Introdução

Esta parte incorpora o *syllabus* do nível fundamental ao cobrir uma série de conceitos gerais que o Gerente de Teste deve levar em conta em relação às ferramentas e à automação.

6.2 Seleção de ferramentas

Existem diversas questões que o Gerente de Teste deve levar em consideração na hora de escolher as ferramentas de teste.

Historicamente, a escolha mais comum consiste em adquirir uma ferramenta de um fornecedor comercial. Em alguns casos, talvez seja a única escolha viável. No entanto, existem outras possibilidades, como ferramentas personalizadas e de código aberto, que também são opções viáveis.

Apesar do tipo da ferramenta, o Gerente de Teste deve ter cuidado na hora de investigar o custo total de propriedade ao longo da vida útil prevista da ferramenta através da realização de uma análise de custo e benefício. Este assunto é discutido abaixo na parte acerca do retorno sobre o investimento (ROI).

6.2.1 Ferramentas de código aberto

As ferramentas de código aberto estão disponíveis em quase qualquer etapa do processo de teste, do gerenciamento de casos de teste ao rastreamento de defeitos, passando pela automação de casos de teste, por exemplo. Uma importante distinção das ferramentas de código aberto é a de que, embora a própria ferramenta geralmente não tenha um custo inicial alto de aquisição, talvez não haja nenhum suporte formal para a ferramenta. Muitas ferramentas de código aberto, contudo, possuem seguidores dedicados que estão dispostos a prestar assistência informal e não tradicional aos usuários.

Além disso, muitas ferramentas de código aberto foram criadas inicialmente para solucionar um problema específico ou lidarem com uma única questão. Portanto, a ferramenta talvez não realize todas as funções de uma ferramenta semelhante comercializada por um fornecedor. Por causa disto, uma análise cuidadosa das necessidades reais do grupo de teste deve ser realizada antes de selecionar uma ferramenta de código aberto.

Um dos benefícios de usar ferramentas de código aberto é o de que, normalmente, podem ser modificadas ou ampliadas por seus usuários. Se a organização de teste possuir competências centrais, a ferramenta pode ser modificada para funcionar com outras ferramentas e alterada para se adequar às necessidades da equipe de teste. Várias ferramentas podem ser combinadas para solucionar os problemas que as ferramentas de fornecedores não conseguem resolver. Lógico, quanto maiores forem o número de ferramentas utilizadas e o número de modificações realizadas, mais complexa e cara ficará a atividade. O Gerente de Teste deve garantir que a equipe não comece a utilizar ferramentas de código aberto só por usá-las. À maneira de outras ferramentas, os trabalhos sempre devem visar à extração de um retorno sobre o investimento positivo.

O Gerente de Teste deve entender o esquema de licenciamento da ferramenta selecionada. Muitas ferramentas de código aberto vêm com uma variação da licença pública geral do GNU, que especifica que a distribuição do *software* sempre deve estar nas mesmas condições em que foi recebido. Se a equipe de teste alterar a ferramenta para suportar melhor o teste, as alterações precisarão ser disponibilizadas a todos os usuários externos da ferramenta segundo a licença da ferramenta. O Gerente de Teste deve verificar as consequências legais da redistribuição do *software* para sua organização.

As organizações que desenvolverem *software* de segurança ou missão crítica ou que estiverem sujeitos à complacência regulamentar podem ter problemas com a utilização de ferramentas de código aberto. Embora muitas ferramentas de código aberto tenham uma qualidade muito alta, a acurácia de qualquer ferramenta de código aberto provavelmente não foi certificada. A acurácia e a adequação das ferramentas dos fornecedores para uma tarefa específica são frequentemente certificados (p. ex., a DO-178B). Embora a ferramenta de código aberto possa ser igualmente boa, a certificação pode ser a responsabilidade do grupo que a utiliza, o que cria outros custos.

6.2.2 Ferramentas personalizadas

A organização de teste pode descobrir que possui uma necessidade específica que fornecedores e ferramentas de código aberto não conseguem atender. Os motivos podem ser uma plataforma de *hardware* exclusiva, um ambiente customizado ou um processo modificado de maneira incomum. Em tais, se as competências centrais existirem na equipe, o Gerente de Teste pode pensar em desenvolver uma ferramenta personalizada.

Os benefícios de desenvolver uma ferramenta personalizada são os de que a ferramenta consegue atender às necessidades da equipe com exatidão e funcionar com eficiência no contexto exigido pela equipe. A ferramenta pode ser elaborada para fazer interface com outras ferramentas utilizadas e para gerar dados no formato específico de que a equipe precisa. Além disso, a ferramenta pode ter aplicações na organização que vão além do projeto imediato. No entanto, antes de planejar o lançamento da ferramenta para outros projetos, é importante primeiro levar em consideração o propósito, a finalidade, os prós e os possíveis contras.

O Gerente de Teste que pensar em criar ferramentas personalizadas também deve levar em conta os possíveis problemas. As ferramentas personalizadas frequentemente dependem da pessoa que as criou. Portanto, as ferramentas personalizadas devem ser adequadamente documentadas para que possam ser mantidas pelos demais. Se isto não for feito, podem ser abandonadas e cair em desuso quando o criador da ferramenta deixar o projeto. Com o tempo, as ferramentas personalizadas podem ter um escopo que vai além da finalidade inicial, o que pode provocar problemas de qualidade na ferramenta, o que leva a relatórios com falsos relatórios de defeitos positivos ou à criação de dados imprecisos. O Gerente de Teste deve ter em mente que as ferramentas personalizadas são apenas outro produto de *software* e, assim, ficam sujeitas aos mesmos tipos de problema de desenvolvimento que qualquer outro produto de *software*. As ferramentas personalizadas devem ser modeladas e testadas para garantir o funcionamento esperado.

6.2.3 Retorno sobre o investimento (ROI)

Cabe ao Gerente de Teste garantir que todas as ferramentas introduzidas na organização de teste agreguem valor ao trabalho da equipe e consigam apresentar um retorno sobre o investimento positivo à organização. Para garantir que a ferramenta consiga benefícios reais e duradouros, uma análise de custo e benefício deve ser realizada antes de adquirir ou elaborar a ferramenta. Nesta análise, o retorno sobre o investimento deve levar em consideração tanto os custos ordinários quanto os extraordinários, alguns dos quais são monetários e outros são custos em recursos ou tempo, e os riscos que podem reduzir o valor da ferramenta.

Entre os custos extraordinários estão:

- A definição de requisitos de ferramentas para atender aos objetivos e às metas;
- A avaliação e a seleção da ferramenta e do fornecedor de ferramentas corretos;
- A aquisição, a adaptação ou o desenvolvimento da ferramenta;
- A realização do treinamento inicial referente à ferramenta;
- A integração da ferramenta com outras ferramentas;
- A aquisição do *hardware* / *software* necessário para suportar a ferramenta.

Entre os custos ordinários estão:

- A responsabilidade pela ferramenta;
- As tarifas de licenciamento e suporte;
- Os custos de manutenção da própria ferramenta;
- A manutenção de artefatos criados pela ferramenta;
- Os custos contínuos de treinamento e mentoria;
- A adaptação da ferramenta a ambientes diferentes;
- A adaptação da ferramenta a futuras necessidades;
- O aprimoramento da qualidade e dos processos para garantir o uso ideal das ferramentas selecionadas.

O Gerente de Teste também deve levar em consideração os custos de oportunidade inerentes a qualquer ferramenta. O tempo de aquisição, administração, capacitação e utilização da ferramenta pode ser passado em tarefas de teste reais. Por conseguinte, pode haver necessidade de mais recursos de teste antes do lançamento da ferramenta.

Existem muitos riscos na utilização de ferramentas. Na verdade, nem todas as ferramentas geram mais benefícios do que riscos. Os riscos das ferramentas foram discutidos no *syllabus* do nível fundamental. O Gerente de Teste também deve levar em consideração os seguintes riscos ao levar em consideração o retorno sobre o investimento:

- A imaturidade da organização (não está pronta para usar a ferramenta);
- Os artefatos criados pela ferramenta podem ser difíceis de manter, exigindo várias revisões quando o *software* testado for alterado;
- A redução do envolvimento dos Analistas de Teste em tarefas de teste pode reduzir o valor do teste (p. ex., a eficácia da detecção de defeitos pode cair quando apenas *scripts* automatizados são executados).

Enfim, o Gerente de Teste deve examinar os benefícios que podem se acumular com base na utilização da ferramenta. Entre os benefícios da introdução e da utilização da ferramenta estão:

- A redução do trabalho repetitivo;
- A redução do tempo de ciclo do teste (p. ex., através da utilização de testes de regressão automatizados);
- A redução dos custos de execução de testes;
- O aumento de certos tipos de teste (p. ex., o teste de regressão);
- A redução do erro humano em fases diferentes de teste. Entre os exemplos estão:
 - Os dados dos testes podem ser mais válidos com ferramentas de geração de dados;
 - As comparações dos resultados dos testes podem ser mais precisas do que as ferramentas de comparação;
 - As entradas dos dados dos testes podem ser mais corretas com a utilização de uma ferramenta de elaboração de *scripts*.
- A redução do esforço necessário para acessar as informações sobre os testes. Entre os exemplos estão:
 - Relatórios e métricas gerados pela ferramenta;
 - A reutilização de ativos de teste, como casos de teste, *scripts* de teste e dados de teste.
- O aumento do teste que não era possível sem ferramentas (p. ex., testes de performance, testes de carga);
- O aprimoramento da situação dos testadores que criam a automatização e da organização de teste como um todo ao demonstrar a compreensão e a utilização de ferramentas sofisticadas.

Em geral, um grupo de teste raramente usa uma única ferramenta. O retorno sobre o investimento total que a equipe de teste obterá costuma ser uma função de todas as ferramentas utilizadas. As ferramentas precisam compartilhar as informações e funcionar juntas. É aconselhável ter uma estratégia abrangente para ferramentas de teste a longo prazo.

6.2.4 Processo seletivo

As ferramentas de teste são um investimento a longo prazo que provavelmente cobrem muitas iterações de um único projeto e / ou podem ser aplicadas em muitos projetos. O Gerente de Teste deve levar em consideração uma ferramenta prospectiva de diversos pontos de vista diferentes:

- Para a empresa, um retorno positivo sobre o investimento é necessário. Para obter alto valor nos investimentos, a organização deve garantir a interoperabilidade das ferramentas – que podem incluir as ferramentas de teste e as que não são utilizadas em testes. Em alguns casos, os processos e a conectividade da utilização de ferramentas devem ser aprimorados para alcançar esta interoperabilidade e fazer isto pode levar algum tempo;
- Para o projeto, a ferramenta deve ser eficaz (p. ex., para evitar erros quando testes manuais forem realizados, como a digitação de erros durante a entrada de dados). A ferramenta pode precisar de muito tempo para começar a gerar um retorno positivo sobre o investimento. Em muitos casos, o retorno sobre o investimento pode ocorrer no segundo lançamento ou

durante a manutenção e não durante o projeto inicial quando a automação foi implementada. O Gerente de Teste deve levar em consideração o ciclo de vida total do aplicativo;

- Para o usuário da ferramenta, a ferramenta deve ajudar os integrantes do projeto a realizar suas tarefas de maneira mais eficiente e eficaz. O aprendizado deve ser levado em consideração para garantir que os usuários consigam conhecer a ferramenta rapidamente com o mínimo de estresse. Quando lançadas, os usuários precisam de capacitação e mentoria para utilizarem as ferramentas de teste.

Para garantir que todos os pontos de vista sejam levados em consideração, é importante criar um roteiro para a apresentação das ferramentas de teste.

O processo seletivo da ferramenta de teste já foi discutido no *syllabus* do nível fundamental da seguinte maneira:

- Avaliar a maturidade da organização;
- Identificar os requisitos das ferramentas;
- Avaliar as ferramentas;
- Avaliar o suporte do fornecedor ou do serviço (ferramentas de código aberto, ferramentas personalizadas);
- Identificar os requisitos internos para o *coaching* e a mentoria referentes à utilização das ferramentas;
- Avaliar as necessidades de treinamento considerando as habilidades de automação de testes da equipe de teste atual;
- Realizar uma análise de custo e benefício (como se discute no item 6.2.3, referente ao retorno sobre o investimento).

Em relação a cada tipo de ferramenta, independentemente da fase de teste em que é utilizada, o Gerente de Teste deve levar em consideração os recursos abaixo:

Análise:

- Esta ferramenta conseguirá “entender” as entradas recebidas?
- A ferramenta é adequada a esta finalidade?

Modelagem:

- Esta ferramenta ajudará o *testware* de modelagem com base nas informações existentes (p. ex., ferramentas de modelagem de testes que criam casos de teste a partir dos requisitos)?
- A modelagem pode ser gerada automaticamente?
- O código real do *testware* pode ser gerado ou parcialmente gerado em um formato utilizável cuja manutenibilidade é possível?
- Os dados dos testes podem ser gerados automaticamente (p. ex., dados gerados a partir da análise do código)?

Seleção de dados e testes:

- Como a ferramenta seleciona os dados de que precisa (p. ex., qual caso de teste deve ser executado com qual conjunto de dados)?
- A ferramenta consegue aceitar os critérios de seleção introduzidos manual ou automaticamente?

- A ferramenta consegue determinar como “depurar” os dados de produção com base nas entradas selecionadas?
- A ferramenta consegue determinar quais testes são necessários com base nos critérios de cobertura (p. ex., considerando um conjunto de requisitos, a ferramenta consegue realizar a rastreabilidade para determinar quais casos de teste é necessário executar?)?

Execução:

- A ferramenta funcionará automaticamente ou haverá necessidade de intervenção manual?
- Como a ferramenta é interrompida e reinicializada?
- A ferramenta deve conseguir “ouvir” eventos pertinentes (p. ex., uma ferramenta de gerenciamento de testes deve atualizar automaticamente a situação dos casos de teste se um defeito divulgado no caso de teste for encerrado?)?

Avaliação:

- Como a ferramenta determinará se recebeu um resultado adequado (p. ex., a ferramenta utilizará um oráculo de teste para determinar a exatidão da resposta)?
- Qual tipo de recurso de recuperação de erros a ferramenta possui?
- A ferramenta realiza registro e divulgação adequados?

6.3 Ciclo de vida das ferramentas

Existem quatro etapas diferentes no ciclo de vida útil de uma ferramenta que o Gerente de Teste deve gerenciar:

1. **Aquisição:** a ferramenta deve ser adquirida como já foi discutido (ver o item 6.2 sobre a seleção de ferramentas). Depois que a decisão de adquirir a ferramenta for tomada, o Gerente de Teste deve atribuir a alguém (frequentemente o Analista de Teste (TA) ou o Analista Técnico de Teste (TTA)) a função de administrador da ferramenta. Esta pessoa deve decidir como e quando a ferramenta será utilizada, onde os artefatos criados serão armazenados, como serão chamados *etc.* Tomar estas decisões antes em vez de deixar que aconteçam *ad hoc* pode fazer uma grande diferença no retorno sobre o investimento eventual da ferramenta. Provavelmente haverá necessidade de ministrar treinamento aos usuários da ferramenta;
2. **Suporte e manutenção:** processos contínuos de suporte e manutenção da ferramenta serão necessários. A responsabilidade por manter a ferramenta pode passar ao administrador de ferramentas ou ser atribuída a um grupo de ferramentas dedicadas. Se a ferramenta funcionar com outras, o intercâmbio de dados e os processos de cooperação devem ser levados em consideração. As decisões sobre o *backup* e a restauração da ferramenta e de suas saídas precisam ser consideradas;
3. **Evolução:** a conversão deve ser levada em consideração. Com o tempo, o ambiente, as necessidades de negócios ou problemas com fornecedores podem exigir que grandes alterações sejam feitas na ferramenta ou em sua utilização. Por exemplo, o fornecedor de ferramentas pode exigir a atualização da ferramenta, o que provoca problemas em outras ferramentas que funcionam com ela. A necessidade de mudanças no ambiente por motivos

comerciais pode provocar problemas na ferramenta. Quanto mais complexo for o ambiente operacional de uma ferramenta, mais as mudanças evolutivas perturbarão sua utilização. Neste momento, dependendo da função que a ferramenta continuar desempenhando nos testes, o Gerente de Teste talvez precise garantir que a organização tenha uma forma de assegurar a continuidade do serviço;

4. **Desativação:** chegará um momento em que a vida útil da ferramenta acabará. Então, a ferramenta precisará ser desativada com graça. A funcionalidade proporcionada pela ferramenta precisará ser substituída e os dados precisarão ser conservados e arquivados. Isto pode ocorrer porque a ferramenta chegou ao fim de seu ciclo de vida ou simplesmente porque chegou a um ponto em que os benefícios e as oportunidades da conversão para uma nova ferramenta superam os custos e os riscos.

Ao longo do ciclo de vida da ferramenta, cabe ao Gerente de Teste garantir o funcionamento adequado e a continuação fluida dos serviços que a ferramenta presta à equipe de teste.

6.4 Métricas das ferramentas

O Gerente de Teste pode modelar e coletar métricas de objetivos das ferramentas utilizadas pelo Analista Técnico de Teste (TTA) e pelo Analista de Teste (TA). Ferramentas diferentes podem captar dados valiosos em tempo real e reduzir os trabalhos de coleta de dados. Os dados podem ser utilizados pelo Gerente de Teste para gerenciar os trabalhos gerais de teste.

Ferramentas diferentes se concentram na coleta de tipos diferentes de dados. Por exemplo:

- As ferramentas de gerenciamento de dados podem fornecer diversas métricas diferentes. A rastreabilidade dos requisitos aos casos de teste e os *scripts* automatizados permite a obtenção de métricas de cobertura. É possível captar os testes atualmente disponíveis, os testes planejados e a situação atual de execução (aprovado, reprovado, pulado, bloqueado, na fila) a qualquer momento;
- As ferramentas de gerenciamento de defeitos podem fornecer muitas informações sobre defeitos, inclusive a situação atual, a gravidade, a prioridade, a distribuição no sistema *etc.* Outros dados esclarecedores, como a fase em que os defeitos foram introduzidos e detectados, os índices de fuga *etc.*, ajudam o Gerente de Teste a orientar o aprimoramento de processos;
- As ferramentas de análise estática podem ajudar a detectar e divulgar os problemas de manutenibilidade;
- As ferramentas de performance podem fornecer informações valiosas sobre a escalabilidade do sistema;
- As ferramentas de cobertura podem ajudar o Gerente de Teste a entender quais partes do sistema realmente foram utilizadas através dos testes.

Os requisitos de divulgação das ferramentas devem ser definidos durante o processo seletivo de ferramentas. Os requisitos devem ser implementados adequadamente durante a configuração de ferramentas para garantir que as informações rastreadas pelas ferramentas possam ser divulgadas de forma compreensível para os *stakeholders*.

7 Habilidades interpessoais – formação de equipes [210 min]

Palavras-chave

independência de testes

Objetivos de aprendizagem

7.2 Habilidades individuais

TM-7.2.1 (K4) Com uma planilha de avaliação de habilidades, analisar os pontos fortes e fracos dos integrantes da equipe relacionados à utilização de sistemas de software ao conhecimento de domínios e negócios, às áreas de desenvolvimento de sistemas, ao teste de software ou às habilidades interpessoais;

TM-7.2.2 (K4) Analisar certa avaliação de habilidades de uma equipe para definir um plano de capacitação e de desenvolvimento de habilidades.

7.3 Dinâmica da equipe de teste

TM-7.3.1 (K2) Em determinadas situações, discutir as habilidades técnicas e sociais que são necessárias para liderar uma equipe de teste.

7.4 Adequação de testes à organização

TM-7.4.1 (K2) Explicar as opções de testes independentes.

7.5 Motivação

TM-7.5.1 (K2) Dar exemplos de fatores de motivação e desmotivação para os testadores.

7.6 Comunicação

TM-7.6.1 (K2) Explicar os fatores que influenciam a eficácia da comunicação na equipe de teste e entre uma equipe de teste e seus stakeholders.

7.1 Introdução

Um Gerente de Teste bem-sucedido recruta, contrata e mantém equipes com uma mistura adequada de habilidades. As habilidades necessárias podem mudar com o tempo, então, além de contratar as pessoas certas, proporcionar-lhes o treinamento adequado e oportunidades de crescimento é importante para reter a equipe de teste e mantê-la em um nível altíssimo de performance. Além das habilidades da equipe, o Gerente de Teste também deve manter um conjunto de habilidade que permitirá o funcionamento eficaz em um ambiente dinâmico de alta pressão.

Este capítulo trata de como avaliar as habilidades, como preencher as lacunas para criar uma equipe sinérgica, entrosada e eficaz em uma organização, como motivar essa equipe e como se comunicar com eficácia.

7.2 Habilidades individuais

A capacidade individual de testar *software* pode ser obtida através da experiência ou da educação e do treinamento. Cada um dos seguintes elementos pode contribuir para a base de conhecimento do testador:

- A utilização de sistemas de *software*;
- O conhecimento de domínios ou negócios;
- A participação em várias fases das atividades do processo de desenvolvimento de *software*, inclusive a análise, o desenvolvimento e o suporte técnico;
- A participação em atividades de teste de *software*.

Os usuários finais de sistemas de *software* sabem bem como o sistema funciona, onde as falhas provocariam o maior impacto e como o sistema deveria reagir em diversas situações. Os usuários com *expertise* em domínios sabem quais áreas são as mais importantes para o negócio e como elas afetam a capacidade de a empresa atender aos requisitos. Este conhecimento pode ser utilizado para facilitar a priorização das atividades de teste, criar dados e casos de teste realistas e verificar ou criar casos de uso.

O conhecimento do processo de desenvolvimento de *software* (a análise de requisitos, a arquitetura, a modelagem e a codificação) mostram como os erros levam à introdução de defeitos, onde os defeitos podem ser detectados e como impedir a introdução de defeitos. A experiência no suporte técnico propicia conhecimento da experiência do usuário, das expectativas e dos requisitos de usabilidade. A experiência em desenvolvimento de *software* é importante para a utilização das ferramentas de teste que exigem *expertise* em programa e modelagem e para a participação em análises de código estático, revisões de código, testes de unidade e testes de integração com foco técnico.

As habilidades específicas de teste de *software* incluem os recursos discutidos no *syllabus* do nível fundamental, no *Advanced Test Analyst (TA) Syllabus* e no *Advanced Technical Test Analyst (TTA) Syllabus*, como a capacidade de analisar especificações, participar de análises de riscos, modelar casos de teste, providenciar a execução de testes e registrar os resultados.

Especificamente para o Gerente de Teste, possuir conhecimento, habilidade e experiência em coordenação de projetos é importante, já que o gerenciamento de testes inclui muitas atividades de coordenação de projetos, por exemplo, fazer planos, rastrear o andamento dos testes e divulgar os resultados aos *stakeholders*. Na ausência de um Gerente de Projetos, o Gerente de Teste pode assumir a função de Gerente de Teste e Gerente de Projeto, particularmente durante as etapas finais de um projeto. Estas habilidades complementam os recursos discutidos no *syllabus* do nível fundamental e neste *syllabus*.

Além das habilidades técnicas, as habilidades interpessoais, como fazer e receber críticas construtivas, influenciar e negociar, são importantes na área de teste. O testador tecnicamente componente provavelmente só terá sucesso se possuir e usar as habilidades sociais necessárias.

Além de trabalhar eficazmente com os demais, o testador bem-sucedido também deve ser organizado e detalhista, além de falar e escrever bem.

A equipe de teste ideal possui uma mistura de habilidades e níveis de experiência e os integrantes da equipe devem querer e poder ensinar e aprender com seus colegas. Em alguns ambientes, algumas habilidades serão mais importantes ou mais respeitadas do que outras. Por exemplo, em um ambiente de teste técnico em que os testes API e as habilidades de programação são necessárias, as habilidades técnicas podem ser mais valorizadas do que o conhecimento de domínios. Em um ambiente de testes caixa-preta, a *expertise* em domínios pode ser a mais valorizada. Vale lembrar que os ambientes e os projetos mudam.

Ao elaborar uma planilha de avaliação de habilidades, o Gerente de Teste deve listar todas as habilidades que são importantes para o trabalho e adequadas ao cargo. Assim que as habilidades forem divididas em itens, cada indivíduo da equipe pode ser avaliado com base em um sistema de pontuação (p. ex., uma nota que vai de 1 a 5, sendo que 5 é o nível mais alto de habilidade esperado em determinada área).

Os indivíduos podem ser avaliados para determinar seus pontos fortes e fracos e, com base nessas informações, é possível criar planos de treinamento individuais ou coletivos. O Gerente de Teste pode definir metas de performance para que os indivíduos aprimorem suas habilidades em áreas específicas e estabeleçam critérios específicos utilizados para avaliar as habilidades do indivíduo.

As pessoas devem ser contratadas a longo prazo e não para sua contribuição em um único projeto. Quando o Gerente de Teste investe em testadores e cria um ambiente de aprendizado contínuo, os integrantes da equipe serão motivados a trabalhar suas habilidades e seu conhecimento para que estejam prontos quando a próxima oportunidade surgir.

O Gerente de Teste raramente conseguirá contratar os integrantes perfeitos para uma equipe. Mesmo se os integrantes da equipe forem perfeitos para o projeto atual, talvez não se deem bem com o projeto seguinte. O Gerente de Teste deve contratar pessoas inteligentes, curiosas, adaptáveis, trabalhadoras, capazes de colaborar de maneira eficaz com uma equipe e dispostas a aprender. Embora os indivíduos perfeitos provavelmente não existam, é possível formar uma equipe forte através do equilíbrio dos pontos fortes e fracos dos indivíduos.

Com a planilha de avaliação de habilidades, o Gerente de Teste consegue identificar onde a equipe é forte e onde ela é fraca. Estas informações servem de base para o plano de treinamento e de

desenvolvimento de habilidades. Com base nos pontos fracos que causam o maior impacto sobre a eficácia e a eficiência da equipe, o Gerente de Teste deve decidir como lidar com tais áreas.

Uma das abordagens é a capacitação, por exemplo, encaminhar as pessoas a um curso de treinamento, realizar sessões de treinamento na empresa, desenvolver treinamentos personalizados e utilizar cursos de *e-learning*. Outra abordagem é o autodidatismo, por exemplo, livros, seminários virtuais e recursos da Internet. Outra abordagem é o treinamento cruzado, por exemplo, fazer que alguém interessado no aprendizado de uma habilidade trabalhe com alguém que já possui a habilidade em uma tarefa que exija tal habilidade, fazer que especialistas locais façam apresentações curtas sobre sua área de *expertise etc.* (A mentoria é uma abordagem parecida em que uma pessoa nova em uma função forma uma dupla com uma pessoa experiente que já a desempenhou e a pessoa experiente serve de recurso constante para proporcionar conselhos e prestar assistência.)

Além de lidar com os pontos fracos, o Gerente de Teste deve se lembrar de que é preciso aproveitar os pontos fortes identificados na avaliação de habilidades como parte do plano de treinamento e desenvolvimento de habilidades. Para obter mais informações sobre os planos de desenvolvimento da equipe de teste, ver [McKay07].

7.3 Dinâmica da equipe de teste

Para formar a melhor equipe possível, a seleção dos funcionários é uma das funções mais importantes da função gerencial na organização. Existem muitos elementos que é preciso levar em conta, além das habilidades individuais específicas que são necessárias para o trabalho. Ao selecionar um indivíduo para a equipe, a dinâmica da equipe deve ser levada em consideração. Esta pessoa complementar as habilidades e os tipos de personalidade que já existem na equipe de teste?

É importante considerar as vantagens de ter vários tipos de personalidade na equipe de teste e diversas habilidades técnicas. Uma equipe de teste forte consegue lidar com vários projetos de complexidade variada ao mesmo tempo em que trata as interações interpessoais com outros integrantes da equipe do projeto com sucesso.

Frequentemente, os testes são atividades de alta pressão. Os cronogramas de desenvolvimento de *software* são frequentemente apertados, até mesmo irrealistas. Os *stakeholders* esperam muito da equipe de teste, às vezes de maneira excessiva. O Gerente de Teste deve contratar pessoas que lidam bem com situações de alta pressão, que conseguem evitar a frustração e conseguem se concentrar no trabalho mesmo quando os prazos parecem impossíveis. Cabe ao Gerente de Teste lidar com os prazos e as expectativas, mas este também deve entender que tais pressões são sentidas por todos os integrantes da equipe também.

Quando o Gerente de Teste contrata pessoas para a equipe, é importante levar em conta o ambiente de trabalho e combinar os tipos de personalidade com tal ambiente. Em um ambiente em que há mais trabalho do que tempo, o Gerente de Teste deve procurar pessoas que terminam suas tarefas e perguntam o que podem fazer a seguir.

Os indivíduos trabalham mais e se preocupam mais pelo que fazem caso se sintam valorizados e necessários. Os indivíduos precisam entender que cada qual é importante para a equipe e que sua

contribuição é vital para o sucesso da equipe como um todo. Quando esta dinâmica é criada, o treinamento ocorrerá informalmente, a carga de trabalho será equilibrada pelos próprios integrantes da equipe e o Gerente de Teste terá mais tempo para lidar com problemas externos.

Os novos integrantes da equipe devem ser assimilados rapidamente à equipe e devem receber supervisão e apoio suficientes. Cada um deve receber uma função definida na equipe. Isto pode se basear no processo de avaliação individual. A meta é a de tornar cada indivíduo bem-sucedido como indivíduo ao mesmo tempo em que ele contribui para o sucesso geral da equipe. Em grande parte, isto é feito combinando tipos de personalidade com funções na equipe, incorporando as habilidades inatas do indivíduo e aumentando sua carteira de habilidades.

Ao determinar quem deve ser contratado ou adicionado à equipe de teste, uma avaliação objetiva de habilidades pode ser útil. Isto pode ser feito através de entrevistas, testes com candidatos, revisões de amostras de trabalho e verificação de referências. Entre as habilidades que devem ser avaliadas estão:

As habilidades técnicas (duras) podem ser demonstradas das seguintes maneiras:

- Extraíndo casos de teste de um documento de requisitos;
- Revisando documentos técnicos (requisitos, códigos *etc.*);
- Redigindo comentários de revisão de maneira clara, compreensível e objetiva;
- Aplicando diversas técnicas de teste adequadamente a determinados cenários (p. ex., a utilização de tabelas de decisão para testar um conjunto de regras de negócios);
- Avaliando uma falha e documentando-a precisamente;
- Demonstrando compreensão das informações de classificação de defeitos;
- Demonstrando compreensão das causas-raiz dos defeitos;
- Usando uma ferramenta para testar determinada API, inclusive testes positivos e negativos;
- Usando SQL para encontrar e alterar informações em bancos de dados para testar determinados cenários;
- Modelando um ambiente de automação de testes que executará vários testes e coletará os resultados dos testes;
- Executando testes automatizados e resolvendo falhas;
- Elaborando planos de teste / especificações;
- Elaborando relatórios de resumo de testes que incluem uma avaliação dos resultados de teste.

As habilidades sociais (suaves) podem ser demonstradas das seguintes maneiras:

- Apresentando informações referentes a um projeto de teste atrasado;
- Explicando um relatório de defeitos a um desenvolvedor que acha que não há defeito nenhum;
- Treinando um colaborador;
- Apresentando um problema à administração referente a um processo ineficaz;
- Revisando um caso de teste criado por um colaborador e apresentando os comentários àquela pessoa;
- Entrevistando um colaborador;
- Elogiando um desenvolvedor.

Embora não seja uma lista completa e as habilidades específicas variem de um ambiente para o outro e de uma organização para a outra, é uma lista de habilidades que são normalmente necessárias. Ao fazer perguntas eficazes em entrevistas e permitir demonstrações de habilidades, o Gerente de Teste consegue avaliar as habilidades do candidato e determinar os pontos fortes e os pontos fracos. Assim que um bom conjunto de avaliações for criado, deve ser aplicado a todos os funcionários para determinar as áreas de crescimento e treinamento.

Além das habilidades necessárias pelos contribuintes individuais, o Gerente de Teste também deve possuir excelentes habilidades de comunicação e diplomacia. O Gerente de Teste deve conseguir apaziguar situações polêmicas, conhecer os meios corretos que devem ser utilizados para as necessidades de comunicação e se concentrar na formação e na manutenção de relacionamentos na organização.

7.4 Adequação de testes à organização

As organizações possuem muitas formas de adequar os testes à estrutura organizacional. Embora a qualidade seja responsabilidade de todos ao longo do ciclo de vida de desenvolvimento de *software*, uma equipe de teste independente pode contribuir bastante com a qualidade de um produto. A independência da função de teste varia bastante na prática, como é possível ver com base na seguinte lista, que vai da menor à maior independência:

Sem testadores independentes:

- Neste caso, não há independência e o código é testado pelo desenvolvedor que o desenvolveu;
- Se tiver tempo para realizar os testes, o desenvolvedor determinará que o código funciona como previsto, o que talvez não confira com os requisitos reais;
- O desenvolvedor consegue corrigir quaisquer defeitos detectados rapidamente.

O teste é realizado por um desenvolvedor que não é o que elaborou o código:

- Há pouca independência entre o desenvolvedor e o testador;
- O desenvolvedor que testa o código de outro desenvolvedor pode relutar em divulgar os defeitos;
- A mentalidade do desenvolvedor em relação ao teste normalmente se concentra em casos de teste positivos.

O teste é realizado por um testador (ou uma equipe de teste) que faz parte da equipe de desenvolvimento:

- O testador (ou a equipe de teste) pode notificar a coordenação do projeto ou o *Gerente de Desenvolvimento*;
- A mentalidade do testador se concentra mais na confirmação da adesão a requisitos;
- Como o testador integra a equipe de desenvolvimento, o testador pode ter responsabilidades de desenvolvimento que vão além do teste;
- O gestor do testador pode se concentrar mais em cumprir o prazo do que em atingir as metas de qualidade.

O teste pode ter um *status* inferior ao desenvolvimento na equipe:

- O testador pode não ter autoridade para influenciar ou aderir as metas de qualidade.

Os testes são realizados por especialistas em teste da empresa, da comunidade de usuários ou outra organização técnica não envolvida no desenvolvimento:

- As informações dos resultados dos testes são divulgadas objetivamente aos *stakeholders*;
- A qualidade é o foco principal da equipe;
- O desenvolvimento de habilidades e a capacitação se concentram nos testes.

O teste é considerado uma trajetória profissional:

- Há um gerenciamento específico dedicado ao grupo de teste, que se concentra na qualidade;
- Especialistas externos em testes testam tipos específicos de testes;
- A experiência é aplicada em áreas específicas em que os testes gerais provavelmente são insuficientes;
- Os tipos de teste podem ser a usabilidade, a segurança, a performance ou outras áreas em que a especialização é necessária;
- A qualidade deve ser o foco destes indivíduos, mas seu ponto de vista se limita às suas áreas de especialidade. Um produto de boa performance talvez não atenda aos requisitos funcionais, mas talvez o especialista em performance desconsidere isso.

Os testes são realizados por uma organização externa à empresa:

- Este modelo atinge a independência máxima entre o testador e o desenvolvedor;
- A transferência de conhecimento talvez não seja suficiente para o testador realizar testes com competência;
- Requisitos claros e uma estrutura bem definida de comunicação serão necessários;
- A qualidade da organização externa deve ser auditada regularmente.

A lista se baseia no foco normal dos indivíduos, mas talvez isso não corresponda a uma organização específica. O cargo e a alçada não determinam necessariamente o foco da pessoa. Os gestores de desenvolvimento podem se concentrar na qualidade e, assim, podem virar bons Gerentes de Teste. O Gerente de Teste independente pode se reportar a uma hierarquia focada em prazos e, assim, pode estar mais voltado para prazos do que para a qualidade. Para determinar a melhor localização de uma equipe de teste em uma organização, o Gerente de Teste deve entender as metas da organização.

Existem vários graus de independência entre as organizações de desenvolvimento e as de teste. É importante entender que talvez haja um meio-termo em que mais independência resulta em mais isolamento e menos transferência de conhecimento. Um nível menor de independência pode aumentar o conhecimento, mas também introduzir metas conflitantes. O nível de independências também será determinado pelo modelo de desenvolvimento de *software* utilizado, por exemplo, no desenvolvimento ágil, os testadores frequentemente fazem parte da equipe de desenvolvimento.

Qualquer uma das opções acima pode aparecer em uma organização. Testes podem ser realizados por uma organização de desenvolvimento e por uma organização independente de teste e pode haver uma certificação final de uma organização externa. É importante compreender as responsabilidades e as expectativas de cada fase de teste e estipular os requisitos para maximizar a

qualidade do produto acabado ao mesmo tempo em que os prazos são mantidos e o orçamento é respeitado.

7.5 Motivação

Existem muitas formas de motivar um indivíduo que ocupa um cargo na área de teste, entre elas:

- O reconhecimento do trabalho realizado;
- A aprovação da administração;
- O respeito na equipe do projeto e entre colegas;
- O aumento da responsabilidade e da autonomia;
- Uma recompensa suficiente pelo trabalho realizado (inclusive remuneração, aumento de responsabilidade, reconhecimento).

Existem influências sobre o projeto que podem dificultar a aplicação destas ferramentas motivacionais. Por exemplo, o testador pode trabalhar muito em um projeto com um prazo impossível. O testador pode fazer de tudo para melhorar o foco da equipe na qualidade, investir horas extra e trabalhar dobrado, mesmo assim, o produto pode ser lançado antes do tempo devido a influências externas. O resultado pode ser um produto de baixa qualidade, apesar de todos os esforços do testador. Isto pode facilmente desmotivar o testador se sua contribuição não foi compreendida e medida, apesar do sucesso do produto finalizado.

A equipe de teste deve garantir o rastreamento das métricas adequadas para provar que um bom trabalho foi feito para a realização de testes, a mitigação de riscos e o registro exato dos resultados. A menos que os dados sejam coletados e publicados, a equipe pode ficar facilmente desmotivada quando não receber o reconhecimento que acha que é devido por conta de um serviço bem-feito.

O reconhecimento não só é determinado pelo respeito e pela aprovação, coisas abstratas, mas também por oportunidades de promoção e pelo aumento de responsabilidades e mérito. Se o grupo de teste não for respeitado, estas oportunidades podem não estar disponíveis.

O reconhecimento e o respeito são adquiridos quando ficar claro que o testador contribui com o valor incremental do projeto. Em um projeto individual, isto é alcançado com o envolvimento do testador desde o início do projeto e com o engajamento contínuo ao longo do ciclo de vida.

Com o tempo, os testadores ganharão o respeito dos demais *stakeholders* do projeto por sua contribuição positiva para o projeto. Esta contribuição também deve ser quantificada em termos de reduções no custo de qualidade e de mitigação de riscos.

O Gerente de Teste desempenha um papel importante na motivação de indivíduos na equipe de teste e defende a equipe de teste perante a organização. O Gerente de Teste deve reconhecer as conquistas dos testadores individuais e fazer avaliações justas e honestas dos erros cometidos. Um Gerente de Teste justo e coerente motivará os integrantes da equipe pelo exemplo.

7.6 Comunicação

A comunicação com a equipe de teste acontece, sobretudo, pelos seguintes meios:

- A documentação dos produtos de teste: estratégia de teste, plano de teste, casos de teste, relatórios de resumo de testes, relatórios de defeitos *etc.*;
- O *feedback* sobre os documentos revisados: requisitos, especificações funcionais, casos de uso, documentos de testes de componentes *etc.*;
- A coleta e a difusão de informações: a interação com os desenvolvedores, outros integrantes da equipe de teste, a administração *etc.*

A comunicação entre os testadores e outros *stakeholders* deve ser profissional, objetiva e eficaz para gerar e manter o respeito pela equipe de teste. A diplomacia e a objetividade são necessárias ao dar *feedback*, particularmente *feedback* construtivo, sobre os produtos de trabalho dos demais. Além disso, a comunicação deve se concentrar na consecução de objetivos de teste e na melhoria da qualidade dos produtos e dos processos utilizados para a produção dos sistemas de *software*.

O Gerente de Teste se comunica com um público amplo, que inclui usuários, integrantes da equipe do projeto, administradores, grupos externos de teste e clientes. A comunicação com o público-alvo deve ser eficaz. Por exemplo, um relatório produzido para a equipe de desenvolvimento que mostrar as tendências no número e na gravidade dos defeitos pode ser detalhado demais para um *briefing* com a diretoria. Em qualquer comunicação, mas particularmente durante apresentações, é importante entender a mensagem enviada, as formas pelas quais a mensagem pode ser recebida e a explicação necessária para criar o ambiente correto para a aceitação da mensagem.

Como o Gerente de Teste frequentemente precisa apresentar a situação do projeto, é importante que estas informações possuam um nível adequado de detalhes (p. ex., normalmente, os gestores desejam ver as tendências dos defeitos em vez de defeitos individuais) e sejam apresentadas de maneira clara e facilmente compreensível (p. ex., com gráficos simples e coloridos). A comunicação eficiente ajuda a manter a atenção do público ao mesmo tempo em que a mensagem correta é transmitida. O Gerente de Teste deve considerar cada apresentação como oportunidade de promover a qualidade e os processos de qualidade.

O Gerente de Teste não se comunica apenas com as pessoas fora do departamento (comunicação para fora). Uma parte importante do trabalho do Gerente de Teste consiste em se comunicar de maneira eficaz dentro do grupo de teste (comunicação para dentro) para lhe repassar notícias, instruções, alterações de prioridades e outras informações-padrão que são comunicadas no processo normal de teste.

O Gerente de Teste também pode se comunicar com indivíduos específicos em alçadas superiores (comunicação para cima) e inferiores (comunicação para baixo) na hierarquia da organização. Valem as mesmas regras independentemente da direção da comunicação. A comunicação deve se adequar ao público. A mensagem deve ser enviada de maneira eficaz e a compreensão deve ser confirmada.

O Gerente de Teste deve dominar os diversos meios de comunicação. Muitas informações são comunicadas por *e-mail*, interações orais, reuniões formais ou informais, relatórios formais ou informais e até mesmo pelo uso das ferramentas de gerenciamento de testes, como as ferramentas de gerenciamento de defeitos. A comunicação deve ser profissional e objetiva.

A revisão, de qualidade e conteúdo, é uma etapa necessária até mesmo na comunicação mais urgente. A comunicação por escrito frequentemente vai além do projeto real. É importante que o

ISTQB® Certified Tester Syllabus

Test Manager



Gerente de Teste produza documentos de qualidade profissional que representem uma organização que promove a qualidade.

Referências

Normas

[**BS7925-2**] BS 7925-2, Software Component Testing Standard, cap. 2;

[**FDA21**] Administração de Alimentos e Medicamentos dos EUA. FDA Title 21 CFR Part 820, cap. 2;

[**IEEE829**] IEEE Standard for Software and System Test Documentation, cap. 2 e 4;

[**IEEE1028**] IEEE Standard for Software Reviews and Audits, cap. 2;

[**IEEE1044**] IEEE Standard Classification for Software Anomalies, cap. 4;

[**ISO9126**] ISO/IEC 9126-1:2001, Software Engineering - Software Product Quality, cap. 2 e 4;

[**ISO12207**] ISO 12207, Systems and Software Engineering, Software Life Cycle Processes, cap.2;

[**ISO15504**] ISO/IEC 15504, Information Technology – Process Assessment, cap.2;

[**ISO25000**] ISO/IEC 25000:2005, Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) , cap. 2 e 4;

[**RTCA DO-178B/ED-12B**] Software Considerations in Airborne Systems and Equipment Certification, RTCA/EUROCAE ED12B.1992, cap. 2.

Documentos do ISTQB

[**ISTQB_AL_OVIEW**] *BSTQB Advanced Level Overview*, versão 1.0

[**ISTQB_ALTM_SYL**] *BSTQB Advanced Level Test Manager (TM) Syllabus*, versão 1.0

[**ISTQB_ALTTA_SYL**] *BSTQB Advanced Level Technical Test Analyst (TTA) Syllabus*, versão 1.0

[**ISTQB_ETM_SYL**] *ISTQB Expert Test Management Syllabus*, versão 1.0

[**ISTQB_FL_SYL**] *BSTQB Foundation Level Syllabus*, versão 2011

[**ISTQB_GLOSSARY**] *Glossário de Termos de Teste de Software*, versão 3.0, 2017

[**ISTQB_ITP_SYL**] *ISTQB Expert Improving the Test Process Syllabus*, versão 1.0

Marcas comerciais

As seguintes marcas registradas e marcas de serviço foram utilizadas neste documento:

BSTQB® é uma marca registrada do *Brazilian Software Testing Qualifications Board*.

CMMI® é uma marca registrada no Instituto Nacional da Propriedade Industrial dos EUA pela Universidade Carnegie Mellon.

IDEAL® é uma marca de serviço do Instituto de Engenharia de *Software* (SEI) da Universidade Carnegie Mellon.

ISTQB® Certified Tester Syllabus

Test Manager



ISTQB® é uma marca registrada da *International Software Testing Qualifications Board*.

ITIL® é uma marca registrada e marca comunitária registrada da Secretaria de Comércio do Governo do Reino Unido e foi registrada no Instituto Nacional da Propriedade Industrial dos EUA.

PRINCE2® é uma marca registrada do Gabinete do Reino Unido.

TMMi® é uma marca registrada da Fundação TMMi.

TPI Next® é uma marca registrada da Sogeti Nederland B.V.

Livros

[Black03] BLACK, Rex. *Critical Testing Processes*, Addison-Wesley, 2003, ISBN 0-201-74868-1.

[Black11] BLACK, Rex; GRAHAM, Dorothy; VEENENDAAL, Erik van. *Foundations of Software Testing*, Thomson Learning, 2011, ISBN 978-1-84480-355-2.

[Craig02] CRAIG, Rick; JASKIEL, Stefan. *Systematic Software Testing*, Artech House, 2002, ISBN 1-580-53508-9

[Crispin09] CRISPIN, Lisa; GREGORY, Janet. *Agile Testing: A Practical Guide for Testers and Agile Teams*, Addison-Wesley, 2009, ISBN 0-321-53446-8.

[de Vries09] VRIES, Gerrit de et al. *TPI Next – UTN Publishers*, 2009, ISBN 90-72194-97-7.

[Goucher09] GOUCHER, Adam; RILEY, Tim (ed.). *Beautiful Testing*, O'Reilly, 2009, ISBN 978-0596159818.

[IDEAL96] MCFEELEY, Bob. *IDEAL: A User's Guide for Software Process Improvement*, Software Engineering Institute (SEI), 1996, CMU/SEI-96-HB-001.

[Jones07] JONES, Capers. *Estimating Software Costs*, 2nd edition, McGraw-Hill, 2007, ISBN 978-0071483001.

[Jones11] BONSIGNOUR, Olivier; JONES, Capers. *Economics of Software Quality*, Pearson, 2011, ISBN 978-0132582209.

[McKay07] MCKAY, Judy. *Managing the Test People – Rocky Nook*, 2007, ISBN 978-1933952123.

[Musa04] MUSA, John. *Software Reliability Engineering*, 2nd edition, Author House, 2004, ISBN 978-1418493882.

[Stamatis03] STAMATIS, D.H. *Failure Mode and Effect Analysis*, ASQC Quality Press, 2003, ISBN 0-873-89300.

[vanVeenendaal11] VEENENDAAL, Erik van. *The Little TMMi*, UTN Publishers, 2011, ISBN 9-490-986038.

[vanVeenendaal12] VEENENDAAL, Erik van. *Practical Risk-Based Testing*, UTN Publishers, 2012, ISBN 978-9490986070.

[Whittaker09] WHITTAKER, James. *Exploratory Testing*, Addison-Wesley, 2009, ISBN 978-0321636416.

Outras referências

As seguintes referências contêm informações disponíveis na Internet. Estas referências foram acessadas no momento da publicação deste *syllabus* de nível avançado.

<http://www.bstqb.org.br>

<http://www.istqb.org>

<http://www.sei.cmu.edu/cmami/>

<http://www.tmmi.org/>

<http://www.tpinext.com/>