

GLOSSÁRIO PADRÃO DE TERMOS UTILIZADOS EM TESTE DE SOFTWARE

Versão 1.3br (01 de Julho de 2007)

Produzido pelo “Glossary Working Party”
International Software Testing Qualification Board

Editor Erik van Veenendaal (Holanda)

Notificação do Detentor dos Direitos Autorais

Este documento poderá ser copiado na íntegra ou em parte desde de que haja menção à sua fonte.

Tradução realizada pela TAG01 (Documentação) do BSTQB

Colaboradores

Rex Black (EUA)
Sigrid Eldh (Suécia)
Isabel Evan (Reino Unido)
Dorothy Graham (Reino Unido)
Julian Harty (Reino Unido)
David Hayman (Reino Unido)
Juha Itkonen (Finlândia)
Vipul Kocher (Índia)
Fernando Lamas de Oliveira (Portugal)
Tilo Linz (Alemanha)
Peter Morgan (Reino Unido)
Thomas Müller (Suíça)
Avi Ofer (Israel)
Dale Perry (EUA)
Horst Pohlmann (Alemanha)
Meile Posthuma (Holanda)
Erkki Pöyhönen (Finlândia)
Maaret Pyhäjärvi (Finlândia)
Andy Redwood (Reino Unido)
Stuart Reid (Reino Unido)
Piet de Roo (Holanda)
Shane Sauders (Reino Unido)
Steve Sampson (Reino Unido)
Shane Sanders (Reino Unido)
Hans Schaefer (Noruega)
Jurriën Seubers (Holanda)
Dave Sherratt (Reino Unido)
Mike Smith (Reino Unido)
Andreas Spillner (Alemanha)
Richard Taylor (Reino Unido)
Geoff Thompson (Reino Unido)
Stephanie Ulrich (Alemanha)
Matti Vuori (Finlândia)
Gearrel Welvaart (Holanda)
Pete Williams (Reino Unido)

Índice

Prefácio	4
1. Introdução.....	4
2. Escopo.....	4
3. Organização	4
4. Referências Normativas	5
5. Definições.....	6
A.....	6
B.....	10
C.....	10
D.....	18
E.....	19
F.....	22
G.....	25
I.....	26
L.....	28
M.....	28
N.....	31
O.....	31
P.....	32
Q.....	34
R.....	34
S.....	38
T.....	41
U.....	54
V.....	54
W.....	55

Prefácio

Na confecção deste glossário, a equipe de trabalho procurou compilar as visões e comentários mais diversos possível, ou seja, de membros da indústria, do comércio e de entidades e organizações governamentais, com o objetivo de padronizar internacionalmente os testes para que sejam aceitos nas mais diferentes áreas. A anuência total muito raramente, se é que chega a acontecer, é alcançada na compilação de documentos desta natureza. Este glossário recebeu contribuições das comunidades de testes de países como Austrália, Bélgica, Finlândia, Alemanha, Índia, Israel, Holanda, Noruega, Portugal, Suécia, Reino Unido e Estados Unidos.

Muitos testadores de software têm utilizado o BS 7925-1 desde a sua primeira publicação em 1998. Ele também tem sido utilizado como importante referência para a certificação concedida pela **Information Systems Examination Board (ISEB)**, (*Comitê de Exames de Sistemas de Informação*) tanto em nível Fundamental/Iniciante como Profissional. Inicialmente, o padrão desenvolvido voltava-se mais para os testes de componentes, porém, desde a sua publicação, muitos comentários e propostas para novas definições foram submetidos com o intuito de aprimorar e ampliar o padrão e para que este cobrisse uma gama maior de testes de software. Nesta nova versão do glossário de teste, pode-se observar que várias das sugestões de atualização apresentadas foram incorporadas. Ele será utilizado como documento de referência pelo esquema de testes de certificação de software aplicado pela **International Software Qualification Board (ISTBQ)**.

1. Introdução

A indústria, o comércio, assim como as instituições governamentais, profissionais e acadêmicas perdem muito tempo e realizam muitos esforços por conta de ambigüidades resultantes da inabilidade de diferenciar adequadamente termos tais como “cobertura dos comandos” (“statement coverage”) e “cobertura de decisão” (“decision coverage”), “suíte de testes” (“test suite”), “especificação de teste” (“test specification”), “plano de teste” (“test plan”) e outros termos similares que formam uma interface entre os vários setores da sociedade. Além disto, o uso técnico ou profissional destes termos está freqüentemente sujeito a variações de significados a eles atribuídos.

2. Escopo

Este documento apresenta conceitos, termos e definições para auxiliar a comunicação na área de testes de software e em disciplinas relacionadas.

3. Organização

O glossário contém uma única seção de definições organizadas alfabeticamente. Alguns termos são preferíveis aos seus sinônimos e, neste caso, aparecerá a definição do termo preferencial e dos respectivos sinônimos. Por exemplo, o termo **teste estrutural (structural testing)** refere-se ao **teste de caixa branca (white box testing)**. Para sinônimos, utilize o indicador “*Ver*”.

Também são utilizadas referências cruzadas do tipo “*Ver também*”. Tais referências auxiliam o usuário a localizar rapidamente o termo no índice. Referências cruzadas do tipo “*Ver também*” auxiliam o entendimento das relações entre os termos mais abrangentes e os menos abrangentes e entre os significados que se sobrepõe a dois termos.

4. Referências Normativas

No momento da publicação, a edição indicada era válida. Todos os padrões estão sujeitos a revisão, e, sugerimos às partes de contratos baseados neste Padrão que investiguem a possibilidade de aplicar as mais recentes edições dos padrões listados abaixo. Os membros do IEC e do ISO mantêm registros dos Padrões Internacionais atualmente válidos.

- BS 7925-2. Software Component Testing.
- DO-178B1992. Software Considerations in Airborne Systems and Equipment Certification, Requirements and Technical Concepts for Aviation (RTCA SC 167).
- IEEE 610.12:1990. Standard Glossary of Software Engineering Terminology.
- IEEE 829:1998. Standard for Software Test Documentation.
- IEEE 10081993. Standard for Software Unit Testing.
- IEEE 10121986. Standard for Verification and Validation Plans.
- IEEE 10281997. Standard for Software Reviews and Audits.
- IEEE 10441993. Standard Classification for Software Anomalies.
- IEEE 12191998. Software Maintenance.
- ISO/IEC 2382-11993. Data processing – Vocabulary – Part 1 Fundamental Terms.
- ISO 9000:2000. Quality Management Systems – Fundamentals and Vocabulary.
- ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1 Quality characteristics and sub-characteristics.
- ISO/IEC 12207:1995. Information Technology – Software Life Cycle Processes.
- ISO/IEC 14598-1:1996. Information Technology – Software Product Evaluation- Part 1 General Overview.

5. Definições

A

Abordagem de teste

(Test approach)

Implementação da estratégia de teste para um projeto específico. Normalmente, inclui as decisões tomadas e baseadas no objetivo do projeto (teste) e na avaliação do risco feita, nos pontos de início relacionados ao processo de teste, nas técnicas de modelagem de teste a serem aplicadas, nos critérios de saída e nos tipos de testes a serem desempenhados.

Aceite

(Acceptance)

Ver Teste de aceite.

Acompanhamento

(Walkthrough)

Apresentação passo a passo feita pelo autor de um documento a fim de reunir informações e de estabelecer um entendimento comum sobre o seu conteúdo. [Freedman e Weinberg, IEEE 1028]. Ver também revisão por pares.

Acompanhamento estruturado

(Structured walkthrough)

Ver revisão.

Acurácia

(Accuracy)

Capacidade do produto de software de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados. [ISO 9126] Ver também Teste de funcionalidade.

Adaptabilidade

(Adaptability)

Capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado. [ISO 9126] Ver também portabilidade.

Adequação

(Suitability)

Capacidade que um produto de software tem de fornecer um conjunto apropriado de funções para as tarefas especificadas e os objetivos do usuário. [ISO 9126]. Ver também funcionalidade.

Alfa Teste

(Alpha testing)

Teste operacional, simulado ou real, realizado por usuários/consumidores em potencial ou por uma equipe de teste independente e conduzido nas instalações do desenvolvedor, mas fora da organização de desenvolvimento. O Alfa Teste é freqüentemente utilizado por “software off-the-shelf” (software de prateleira/pacote) como forma interna de teste de aceite.

Alterabilidade

(Changeability)

Capacidade que um produto de software tem para possibilitar a implementação de alterações específicas. [ISO 9126] Ver também manutenibilidade.

Alvo de teste

(Test target)

Conjunto de critérios de saída.

Ambiente de teste

(Test bed)

Ver ambiente de teste.

Ambiente de teste

(Test environment)

Ambiente que contém hardware, instrumentação, simuladores, ferramentas de software e outros elementos de suporte necessários à realização de um teste. [Subsequente a IEEE 610]

Ambiente operacional

(Operational environment)

Produtos de software ou hardware instalados nos locais de trabalho, residência dos usuários ou consumidores, onde o componente ou sistema sendo testado será utilizado. O software pode incluir sistemas operacionais, sistemas de gerenciamento de banco de dados e outros aplicativos.

Ambiente preparado para testes

(Test harness)

Ambiente de teste composto de simuladores (stubs) e controladores (drivers) necessários para a condução de um teste.

Analisador

(Analyzer)

Ver analisador estático.

Análise de árvore de faltas

(Fault tree analysis)

Método utilizado para analisar as causas de faltas (defeitos).

Análise de causa e efeito

(Cause-effect analysis)

Ver gráfico de causa e efeito.

Análise de cobertura

(Coverage analysis)

Medição da cobertura alcançada por um item de cobertura específico durante a execução do teste com relação aos critérios pré-determinados feita para determinar se há necessidade de se fazer teste adicional; e, em caso positivo, quais casos de teste serão necessários.

Análise de código estático

(Static code analysis)

Análise do código de fonte realizada sem a execução daquele software.

Análise de fluxo de dados

(Data flow analysis)

Forma de análise estática baseada na definição e na utilização de variáveis.

Análise de impacto

(Impact analysis)

Avaliação de alterações ocorridas nas camadas de documentação de desenvolvimento e na documentação e componentes de testes a fim de implementar uma dada alteração nos requisitos especificados.

Análise de mutação

(Mutation analysis)

Método que determina a acuidade da suite de teste medindo a extensão até a qual uma suite de teste pode discernir entre o programa e suas pequenas variantes (mutantes).

Análise de Ponto de Função

(Function Point Analysis (FPA))

Método que visa medir o tamanho da funcionalidade de um sistema de informações. A medição independe de tecnologia. Ela pode ser utilizada como base para medição de produtividade, para a estimação de recursos necessários e para controle de projeto.

Análise de Ponto de Teste

(Test Point Analysis - TPA)

Método de estimação de teste que usa fórmula baseada na Análise de Ponto de Função. [TMap]

Análise de risco

(Risk analysis)

Processo pelo qual se averigua os riscos identificados para estimar impactos e probabilidade de ocorrência (possibilidade).

Análise de Valor Limite

(Boundary Value Analysis)

Técnica de modelagem de teste caixa preta na qual os casos de teste são projetados com base nos valores limite.

Análise dinâmica

(Dynamic analysis)

Processo que avalia o comportamento, por exemplo, o desempenho da memória ou o uso da CPU, de um sistema ou componente durante sua execução. [Subsequente a IEEE 610]

Análise estática

(Static analysis)

Análise de artefatos de software, por exemplo, requisitos ou código, realizada sem a execução destes artefatos.

Analísibilidade

(Analyzability)

Capacidade do produto de software de permitir o diagnóstico de deficiências ou causas de falhas no software, ou a identificação de partes a serem modificadas. [ISO 9126] Ver também manutenibilidade.

Analista de código

(Code analyzer)

Ver analista de código estático.

Analista de código estático

(Static code analyser)

Ferramenta que realiza a análise de código estático. A ferramenta verifica o código de fonte em busca de certas propriedades, tais como conformidade para com os padrões de codificação, qualidade da métrica ou anomalia de fluxos de dados.

Analista estático

(Static analyser)

Ferramenta que realiza a análise estática.

Anomalia

(Anomaly)

Qualquer condição que se desvie da expectativa proposta pelas especificações dos requisitos para uma modelagem dos documentos, documentos de modelagem, padrões, etc. ou da percepção ou experiência de uma determinada pessoa. Dentre outros motivos, as anomalias podem ser encontradas durante as revisões, os testes, as análises, as compilações ou a aplicação do uso dos produtos de software ou da documentação. [IEEE 1044]. Ver também defeito, desvio, erro, dano, falha, incidente e problema.

Apreensibilidade

(Learnability)

Capacidade que um produto de software tem de possibilitar ao usuário aprender suas aplicações. [ISO 9126]. Ver também usabilidade.

Aprovação

(Pass)

Um teste será aprovado se o seu resultado real for igual ao resultado esperado.

Aprovação de teste

(Test pass)

Ver aprovação.

Armazenamento

(Storage)

Ver utilização de recurso.

Assistente de instalação

(Installation wizard)

Software fornecido por qualquer mídia adequada para guiar uma pessoa durante o processo de instalação. Normalmente executa o processo de instalação, fornece feedback sobre os resultados da instalação e dá “dicas” de opções.

Atratividade

(Attractiveness)

Capacidade do produto de software de ser atraente ao usuário. [ISO 9126] Ver também usabilidade.

Atributo de qualidade

(Quality attribute)

Característica que afeta a qualidade de um item. [IEEE 610]

Auditoria

(Audit)

Avaliação independente feita nos produtos de software, ou nos processos, a fim de averiguar com segurança a conformidade aos padrões, às diretrizes, às especificações e/ou aos procedimentos baseados em critérios objetivos, incluindo documentos que especifiquem: (1) A forma ou o conteúdo dos produtos a serem produzidos. (2) O processo pelo qual os produtos deverão ser produzidos. (3) Como a conformidade aos padrões e às diretrizes deverá ser medida [IEEE 1028].

Auditoria de configuração

(Configuration auditing)

Função que verifica os conteúdos das bibliotecas de itens de configuração, por exemplo, verificação do atendimento aos padrões. [IEEE 610]

Automação da execução de teste

(Test execution automation)

Utilização de um software, ex. ferramentas de captura/recuperação, para controlar a execução de testes, a comparação entre os resultados reais e os esperados, o estabelecimento de precondições e outras funções de controle de teste e de relato.

Automação de teste

(Test automation)

Utilização de software para desempenhar ou dar suporte às atividades de teste, por exemplo, gerenciamento de teste, modelagem de teste, execução de teste e verificação de resultados.

Avaliação

(Evaluation)

Ver testing (teste).

Avaliação heurística

(Heuristic evaluation)

Técnica estática de teste de usabilidade que determina o atendimento da interface de um usuário aos princípios de uso reconhecidos (os assim chamados princípios "heurísticos").

B

Base de teste

(Test basis)

Todos os documentos a partir dos quais os requisitos de um determinado componente ou sistema podem ser inferidos. Documentação na qual os casos de testes estão baseados. Se um documento pode ser alterado somente por meio de procedimento formal, então a base de teste passa a se chamar base de teste congelada. [Subsequente a TMap]

Base de Teste congelada

(Frozen test basis)

Documento para base de teste que só pode ser alterado por um processo formal de controle de alteração. Ver também linha de base.

Bebugging

(Bebugging)

Ver envio de erro. [Abbott]

Beta Teste

(Beta testing)

Teste operacional realizado por usuários/consumidores existentes/potenciais em um local externo, sem envolvimento dos desenvolvedores, a fim de determinar se um componente ou sistema satisfaz, ou não, as necessidades dos usuários/consumidores e se encaixa dentro dos processos dos negócios. O teste beta é freqüentemente utilizado como uma forma de teste de aceitação externo para o off-the-shelf software (software de prateleira) e para assim obter feedback do mercado.

Bloco básico

(Basic block)

Seqüência de uma ou mais sentenças consecutivas executáveis que não contenham desvios.

Bug

(Bug)

Ver defeito.

C

Caminho

(Path)

Seqüência de eventos, por exemplo, sentenças executáveis, de um componente ou sistema a partir de um ponto de entrada até um ponto de saída.

Caminho inviável

(Infeasible path)

Caminho que não pode ser exercido por nenhum conjunto de valores de entrada/input possível.

Caminho possível

(Feasible path)

Caminho para o qual existe um conjunto de valores de entrada/input e de precondições que faz com que ele seja executado.

Característica

(Feature)

Atributo de um componente ou sistema especificado ou implícito na documentação de requisitos (por exemplo, restrições de confiabilidade, de uso ou de modelagem). [Subsequente ao IEEE 610]

Característica de qualidade

(Quality characteristic)

Ver atributo de qualidade.

Característica de qualidade de software

(Software quality characteristic)

Ver atributo de qualidade.

Característica de software

(Software feature)

Ver característica.

CASE

(CASE)

Acrônimo para Computer Aided Software Engineering (Auxílio à Engenharia de Software por Computador).

Caso de teste

(Test case)

Conjunto de valores de entrada/inputs, condições de execução, resultados esperados e pós-condições de execução desenvolvidas para um determinado objetivo ou condição de teste, tais como para exercitar o caminho de um determinado programa ou verificar o atendimento a um requisito específico. [Subsequente a IEEE 610]

Caso de teste abstrato

(Abstract test case)

Ver Caso de teste de alto nível.

Caso de teste bloqueado

(Blocked test case)

Um caso de teste que não pode ser executado, pois as condições para sua execução não foram atendidas.

Caso de teste concreto

(Concrete test case)

Ver caso de teste de nível baixo.

Caso de teste de alto nível

(High level test case)

Caso de teste sem valores concretos (nível de implementação) para os dados de entrada/inputs e para resultados esperados. Utilizam operadores e as instancias dos valores reais ainda não estão definidas e/ou disponíveis. Ver também caso de teste de nível baixo.

Caso de teste de baixo nível

(Low level teste case)

Caso de teste com valores concretos (nível de implementação) para os dados de entrada/inputs e resultados esperados. Os operadores lógicos de casos de teste de alto nível são substituídos por valores reais que correspondem aos objetivos dos operadores lógicos. Ver também caso de teste de nível alto.

Caso de teste lógico

(Logical test case)

Ver caso de teste de nível alto.

Caso de uso

(Use case)

Seqüência de transações em um diálogo, entre um usuário e um sistema, com um resultado tangível.

CAST

(CAST)

Acrônimo para Computer Aided Software Testing (Auxílio a Testes por Computador). Ver também automatização de teste.

Causa raiz

(Root cause)

Fator subjacente responsável por causar a não conformidade e que, possivelmente, deva ser permanentemente eliminado por meio de melhoria do processo.

Cenário de teste

(Test scenario)

Ver especificação de procedimento de teste.

Certificação

(Certification)

Processo que confirma se um determinado componente, sistema ou pessoa atende aos requisitos especificados, ex. aprovação em um exame.

Ciclo de teste

(Test cycle)

Execução do processo de teste contra um único release identificável do objeto de teste.

Classe de equivalência

(Equivalence class)

Ver partição de equivalência.

Cobertura

(Coverage)

Grau, expresso em porcentagem, até o qual um item de cobertura específico foi exercitado por uma suite de teste.

Cobertura da condição de decisão

(Decision condition coverage)

Porcentagem de todos os resultados de condições e resultados de decisões, exercitados por uma suite de teste. 100% de cobertura de condição de decisão implicam tanto em 100% de cobertura de condição como em 100% de cobertura de decisão.

Cobertura de caminho

(Path coverage)

Porcentagem de caminhos exercitada por uma suite de teste. 100% de cobertura de caminho implicam em 100% de cobertura LCSAJ.

Cobertura de código

(Code coverage)

Método de análise que determina quais parte de um determinado software foram executadas (cobertas) por uma suite de teste e quais não, por exemplo, cobertura das sentenças, cobertura da decisão e cobertura da condição.

Cobertura de combinação de condição

(Condition combination coverage)

Ver cobertura de condições múltiplas.

Cobertura de condição

(Condition coverage)

Porcentagem de conseqüências/resultados de condições exercitada por uma suíte de teste. 100% de cobertura de condição requerem que cada condição única em toda sentença de decisão seja testada em termos de Verdadeiro ou Falso.

Cobertura de condição de desvio

(Branch condition coverage)

Ver cobertura de condição.

Cobertura de condição múltipla

(Multiple condition coverage)

Porcentagem de combinações de todos os resultados de condições únicas dentro de uma sentença exercitada por uma suíte de teste. 100% de cobertura de condição múltipla implicam em 100% de cobertura de determinação de condição.

Cobertura de condição múltipla modificada

(Modified multiple condition coverage)

Ver cobertura de determinação de condição.

Cobertura de decisão

(Decision coverage)

Porcentagem de resultados de decisão exercitados por uma suíte de teste. 100% de cobertura de decisão implica tanto em 100% de cobertura de desvio como em 100% de cobertura de sentença.

Cobertura de decisão de condição modificada

(Modified condition decision coverage)

Ver cobertura de determinação de condição.

Cobertura de desvio

(Branch coverage)

Porcentagem de desvios exercitados por uma suíte de teste. 100% de cobertura de desvio implicam em 100% de cobertura de decisão e em 100% de cobertura de sentença.

Cobertura de determinação de condição

(Condition determination coverage)

Porcentagem de todos os resultados de condições únicas que afeta de modo independente um resultado de decisão exercitado por uma suíte de caso de teste. 100% de cobertura de determinação de condição implicam em 100% de cobertura de condição de decisão.

Cobertura de fluxo de dados

(Data flow coverage)

Porcentagem de pares de definição-uso exercitada por uma suíte de teste.

Cobertura de partição de equivalência

(Equivalence partition coverage)

Porcentagem de partições de equivalência exercitada por uma suíte de teste.

Cobertura de sentença (comando)

(Statement coverage)

Porcentagem de sentenças executáveis exercitadas por uma suíte de teste.

Cobertura de Seqüência de Código Linear e Salto

(LCSAJ coverage)

Porcentagem de LCSAJs de um componente exercitado por um suíte de teste. 100% de cobertura LCSAJ implica em 100% de cobertura de decisão.

Cobertura de teste

(Test coverage)

Ver cobertura.

Cobertura do valor limite

(Boundary value coverage)

Porcentagem de valores limite exercitados por uma suite de teste.

Cobertura estrutural

(Structural coverage)

Medidas de cobertura baseadas na estrutura interna de um componente ou sistema.

Cobertura N-switch

(N-switch coverage)

Porcentagem de seqüências de transições N+1 exercitadas por uma suite de teste. [Chow]

Código

(Code)

Instruções de computador e definições de dados expressos por um montador, compilador ou outro tradutor em uma linguagem de programação ou em forma de resultado/saída/output. [IEEE 610]

Código inatingível

(Unreachable code)

Código que não pode ser alcançado e que, portanto, não pode ser executado.

Código morto

(Dead code)

Ver código inatingível.

Coexistência

(Co-existence)

Capacidade que um produto de software tem para coexistir com outro software independente em um ambiente comum compartilhando de recursos comuns. [ISO 9126]. Ver também portabilidade.

Combinação de cobertura de condição de desvio

(Branch condition combination coverage)

Ver cobertura de condições múltiplas.

Comitê de controle de alteração

(Change control board)

Ver comitê de controle de configuração.

Comitê de controle de configuração

(Configuration control board (CCB))

Grupo de pessoas responsável pela avaliação e aprovação, ou desaprovação, das alterações propostas para os itens de configuração e também por assegurar a implementação das alterações aprovadas. [IEEE 610]

Comparação de teste

(Test comparison)

Processo que identifica diferenças entre os resultados reais produzidos pelo componente ou sistema sendo testado e os resultados esperados para o teste. A comparação de teste pode ser desempenhada durante a execução do teste (comparação dinâmica) ou após sua execução.

Comparação dinâmica

(Dynamic comparison)

Comparação entre os resultados reais e esperados realizada durante a execução do software, por exemplo, por uma ferramenta de execução de teste.

Comparação pós-execução

(Post-execution comparison)

Comparação entre os resultados reais e os esperados, desempenhada após o software ser executado.

Comparador

(Comparator)

Ver comparador de teste.

Comparador de teste

(Test comparator)

Ferramenta de teste que faz a comparação automatizada de testes.

Compatibilidade

(Compliance)

Capacidade de um produto de software de atender aos padrões, convenções ou regulamentações da lei e de prescrições similares. [ISO 9126]

Compilador

(Compiler)

Ferramenta de software que traduz programas expressos em uma linguagem de alto-nível para uma linguagem de máquina. [IEEE 610]

Complexidade

(Complexity)

Grau de dificuldade de entendimento, manutenção e verificação que uma modelagem e/ou estrutura interna apresenta. Ver também complexidade ciclomática.

Complexidade ciclomática

(Cyclomatic complexity)

Número de caminhos independentes através de um programa. A complexidade ciclomática é definida por $(L - N + 2P)$, onde: (1) L = ao número de edges/links em um grafo, (2) N = ao número de nós em um grafo, (3) P = número de partes desconectadas em um grafo (por exemplo, um grafo chamado/called graph e uma sub-rotina). [Subseqüente a McCabe]

Componente

(Component)

Item mínimo de software que pode ser testado isoladamente.

Comportamento

(Behavior)

Resposta que um componente ou sistema dá a um conjunto de valores de entrada/inputs e de precondições.

Comportamento relacionado a tempo

(Time behavior)

Ver desempenho.

Condição

(Condition)

Expressão lógica que pode ser avaliada em termos de Verdadeiro ou Falso, por exemplo, $A > B$. Ver também condição de teste.

Condição composta

(Compound condition)

Duas ou mais condições únicas reunidas por meio de um operador lógico (AND, OR, ou XOR-E, OU, ou XOR), por exemplo, ' $A > B$ AND/E $C > 1000$ '.

Condição de desvio

(Branch condition)

Ver Condição.

Condição de teste

(Test condition)

Item ou evento de um componente ou sistema que pode ser verificado por um ou mais casos de teste, por exemplo: função, transação, característica, atributo de qualidade ou elemento estrutural.

Condição múltipla

(Multiple condition)

Ver condição composta.

Confiabilidade

(Reliability)

Habilidade do produto de software para desempenhar suas funções sob certas condições – período de tempo específico ou número de operações específico. [ISO 9126]

Configuração

(Configuration)

Composição de um componente ou sistema conforme definido pelo número, natureza e interligação de suas partes constituintes.

Conjunto de teste

(Test set)

Ver suite de teste.

Conjunto de testes de base

(Basis test set)

Conjunto de casos de testes derivados da estrutura interna de um componente ou especificação com o fim de assegurar que 100% de um determinado critério de cobertura seja alcançado.

Conseqüência

(Outcome)

Ver resultado.

Conseqüência do teste

(Test outcome)

Ver resultado.

Conseqüência prevista

(Predicted outcome)

Ver resultado esperado.

Consistência

(Consistency)

Grau de uniformidade, padronização e isenção de contradição entre os documentos ou as partes de um componente ou sistema. [IEEE 610]

Contabilização da condição

(Status accounting)

Elemento do gerenciamento de configuração que consiste na gravação e relato das informações necessárias para gerenciar a configuração eficazmente. Estas informações incluem uma lista da identificação de configuração aprovada, a condição das alterações de configuração propostas e o estado de implementação das alterações aprovadas. [IEEE 610]

Controlador

(Driver)

Componente de software ou ferramenta de teste que substitui um componente que cuida do controle e/ou a chamada de um componente ou sistema. [Subsequente a TMap]

Controlador de teste

(Test driver)

Ver controlador.

Controle de alteração

(Change control)

Ver controle de configuração.

Controle de configuração

(Configuration control)

Elemento de gerenciamento de configuração que consiste na avaliação, coordenação, aprovação ou desaprovação e na implementação de alterações nos itens de configuração após o estabelecimento formal de suas identificações de configuração. [IEEE 610]

Controle de risco

(Risk control)

Processo que toma decisões e implementa medidas de proteção para reduzir riscos ou para mantê-los em níveis específicos.

Controle de teste

(Test control)

Tarefa do gerenciamento de teste que lida como desenvolvimento e aplicação de um conjunto de ações corretivas quando o monitoramento mostra qualquer desvio do originalmente planejado. Ver também gerenciamento de teste.

Controle de versão

(Version control)

Ver controle de configuração.

COTS

(COTS)

Acrônimo para Commercial Off-the-Shelf Software (Software Comercial de Prateleira). Ver software de prateleira.

Critério de aceite

(Acceptance criteria)

Critério de saída que um componente ou sistema deve satisfazer para ser aceita por um usuário, cliente ou outra entidade autorizada. [IEEE 610].

Critérios de conclusão

(Completion criteria)

Ver critérios de saída.

Critérios de conclusão de teste

(Test completion criteria)

Ver critérios de saída.

Critérios de entrada

(Entry criteria)

Conjunto de condições genéricas e específicas que permite a um processo prosseguir com uma determinada tarefa, por exemplo, a fase de teste. A finalidade dos critérios de entrada é evitar que uma tarefa que significaria mais esforço (perdido) comparado ao esforço necessário para remover os critérios de entrada falhos seja iniciada. [Gilb e Graham]

Critérios de reinício

(Resumption criteria)

Atividades que devem ser repetidas quando um teste for reiniciado após sua suspensão. [Subsequente a IEEE 829]

Critérios de saída

(Exit criteria)

Conjunto de condições genéricas e específicas, acordadas pelos stakeholders/interessados, que permite que um processo seja oficialmente considerado completado. A finalidade dos critérios de saída é evitar que uma tarefa seja considerada completa quando ainda existirem partes importantes dela que ainda não tenham sido terminadas. Os critérios de saída são utilizados para relatar e para planejar o momento de interromper os testes. [Subsequente a Gilb e Graham]

Critérios de suspensão

(Suspension criteria)

São os critérios utilizados para interromper (temporariamente) todas ou parte das atividades de teste nos itens de teste. [Subsequente a IEEE 829]

Critérios para aprovação/reprovação

(Pass/fail criteria)

Regras de decisão utilizadas para determinar se um item de teste (função) ou uma característica foi aprovada ou reprovada no teste. [IEEE 829]

Cronograma de execução de teste

(Test execution schedule)

Esquema para a execução dos procedimentos de teste. Os procedimentos de teste são incluídos no cronograma do contexto de execução do teste na ordem em que deverão ser executados.

D

Dados de teste

(Test data)

Dados existentes (ex. em um banco de dados) antes do início da execução de um teste e que afetam ou são afetados pelo componente ou sistema sendo testado.

Debugger

(Debugger)

Ver ferramenta debugging.

Debugging

(Debugging)

Nome dado ao processo que encontra, analisa e remove as causas de falhas em um software.

Decisão

(Decision)

Nome dado ao ponto de um programa no qual o fluxo de controle tem duas ou mais rotas alternativas. Um nó com dois ou mais links separa os desvios.

Defeito

(Defect)

Falha em um componente ou sistema que pode fazer com que o componente ou sistema falhe ao desempenhar sua devida função, por exemplo, uma sentença ou definição de dados incorretos. Um defeito, se descoberto durante a execução, pode causar uma falha no componente ou sistema.

Definição dos dados

(Data definition)

Sentença executável na qual uma variável recebe um determinado valor.

Densidade de falta

(Fault density)

Ver densidade de defeito.

Densidade do defeito

(Defect density)

Trata-se do número de defeitos identificados em um componente ou sistema dividido pelo tamanho do componente ou sistema (expresso em termos de medida padrão, por exemplo: linhas de código, número de classes ou de pontos de função).

Desempenho

(Performance)

Grau até o qual um sistema ou componente cumpre com as funções a ele destinadas de acordo com restrições de tempo de processo e de taxa de rendimento/throughput. [Subseqüente a IEEE 610]. Ver também eficiência.

Desenvolvimento orientado ao teste

(Test driven development)

Modo de desenvolvimento de software no qual os casos de teste são desenvolvidos, e, freqüentemente automatizados, antes que o software seja desenvolvido para rodar esses casos de teste.

Desvio

(Deviation)

Ver incidente.

Desvio/Ramo

(Branch)

Bloco básico que pode ser selecionado para execução baseada na construção de um programa no qual um ou mais caminhos de programas interativos estão disponíveis, por exemplo: caso, salto, ir para, "if-then-else".

Diagrama de estado

(State diagram)

Diagrama que descreve os estados que um componente ou sistema pode assumir. Mostra também os eventos e circunstâncias que causam e/ou resultam da alteração de um estado para outro. [IEEE 610]

Disponibilidade

(Availability)

É a medida pela qual um componente ou sistema está operacional ou acessível quando seu uso é requisitado. Freqüentemente expresso em porcentagem. [IEEE 610].

Domínio

(Domain)

Conjunto a partir do qual valores válidos de entrada/input e/ou de saída/output podem ser selecionados.

Domínio de entrada

(Input domain)

Conjunto a partir do qual os valores de entrada/input válidos podem ser selecionados. Ver também domínio.

Domínio de saída

(Output domain)

Conjunto a partir do qual valores de saída/output válidos podem ser selecionados. Ver também domínio.

E

Efeito da monitoração

(Probe effect)

Efeito causado no componente ou sistema pelo instrumento de medição quando o componente ou sistema está sendo medido, ex. por uma ferramenta de teste de desempenho ou por um monitor. Por exemplo, o desempenho poderá ser um pouco pior quando as ferramentas de teste de desempenho forem utilizadas.

Eficiência*(Efficiency)*

Capacidade que um produto de software tem de fornecer um desempenho apropriado com relação à quantidade de recursos utilizados sob determinadas condições. [ISO 9126]

Emulador*(Emulator)*

Dispositivo, programa de computador ou sistema que aceita as mesmas entradas/inputs e produz as mesmas saídas/saídas de um dado sistema, [IEEE 610] Ver também simulador.

Engano*(Mistake)*

Ver erro.

Entendibilidade*(Understandability)*

Capacidade que um produto de software tem de possibilitar ao usuário entender se o software é adequado para uso, e como ele pode ser utilizado em determinadas tarefas e condições de uso. [ISO 9126]. Ver também usabilidade.

Entrada*(Input)*

Variável (armazenada dentro ou fora de um componente) lida por um componente.

Entrada de teste*(Test input)*

Dados recebidos pelo objeto do teste de uma fonte externa durante a execução do teste. A fonte externa pode ser um hardware, um software ou uma pessoa.

Entrada especificada*(Specified input)*

Uma entrada/input para a qual a especificação prevê um resultado.

Entregável*(Deliverable)*

Qualquer produto (trabalho) que deva ser entregue a alguém que não o autor do produto (trabalho).

Erro*(Error)*

Ação humana que produz resultados incorretos. [Subsequente a IEEE 610]

Escala de medição*(Measurement scale)*

Escala que restringe o tipo de análise de dados que pode ser desempenhada nela. [ISO 14598]

Escalabilidade*(Scalability)*

Capacidade que um produto de software tem para sofrer um upgrade ou para acomodar aumento de cargas. [Subsequente a Gerrard]

Especificação*(Specification)*

Documento que especifica, idealmente de modo completo, preciso e passível de verificação, os requisitos, a modelagem, o comportamento ou outras características de um componente ou sistema, e freqüentemente também os procedimentos que determinam se essas disposições foram satisfeitas. [Subsequente a IEEE 610]

Especificação de caso de teste

(Test case specification)

Documento que especifica um conjunto de casos de teste (objetivos, entradas/entradas, ações do teste, resultados esperados e precondições para execução) para um item de teste. [Subsequente a IEEE 829]

Especificação de componente

(Component specification)

Descrição da função de um componente em termos de seus valores de saída/outputs para valores de entrada/input específicos, sob situações específicas e descrição do comportamento não funcional requerido (ex. recurso-utilização).

Especificação de modelagem de teste

(Test design specification)

Documento que especifica as condições de teste (cobertura de itens) para um item de teste. Detalha a abordagem de teste e identifica os casos de teste de alto nível associados. [Subsequente a IEEE 829]

Especificação de procedimento (roteiro) de teste

(Test procedure specification)

Documento que especifica uma seqüência de ações para a execução de um teste. Também conhecido como roteiro de teste ou roteiro de teste manual. [Subsequente a IEEE 829]

Especificação de teste

(Test specification)

Documento que consiste em uma especificação da modelagem de teste, em uma especificação de caso de teste e/ou em uma especificação de procedimento de teste.

Estabilidade

(Stability)

Capacidade que um produto de software tem para evitar efeitos inesperados resultantes de modificações feitas em um software. [ISO 9126]. Ver também manutenibilidade.

Estágio de teste

(Test stage)

Ver nível de teste.

Estratégia de teste

(Test strategy)

Descrição de alto nível dos níveis de teste a serem realizados e do teste dentro desses níveis feitos para uma organização ou programa (um ou mais projetos).

Execução de teste

(Test execution)

Processo de executar um teste em um componente ou sistema sendo testado e que produz resultado(s) real(ais).

Execução de teste

(Test run)

Execução de um teste em uma versão específica de um objeto de teste.

Exercitado

(Exercised)

Diz-se que um elemento de um programa foi exercitado por um caso de teste quando o valor de entrada/input faz com que este elemento seja executado, por exemplo, uma sentença, uma decisão ou outro elemento estrutural.

F

Falha

(Failure)

Desvio do componente ou sistema da entrega, resultado ou serviço esperado. [Subsequente a Fenton]

Falta

(Fault)

Ver defeito.

Fase de execução de teste

(Test execution phase)

Período de tempo do ciclo de vida de desenvolvimento de um software durante o qual os componentes de um produto de software são executados, e o produto de software é avaliado para determinar se os requisitos foram ou não satisfeitos. [IEEE 610]

Fase de requisito

(Requirement phase)

Período de tempo no ciclo de vida de um software durante o qual os requisitos para um produto de software são definidos e documentados. [IEEE 610]

Fase de teste

(Test phase)

Conjunto distinto de atividades de teste coletadas em uma fase gerenciável do projeto, ex. durante a execução das atividades de um nível de teste. [Subsequente a Gerrard]

Fechamento de teste

(Test closure)

Durante a fase de fechamento de um processo de teste, coletam-se dados das atividades já completadas a fim de consolidar a experiência, o testware, os fatos e os números. A fase de fechamento consiste em finalizar e arquivar o testware e em avaliar o processo de teste, inclusive com a preparação de um relatório de avaliação de teste. Ver também processo de teste.

Ferramenta de análise dinâmica

(Dynamic analysis tool)

Ferramenta que fornece informação sobre o tempo de execução do estado do código de software. Essas ferramentas são freqüentemente utilizadas para identificar indicadores não declarados, verificar a função aritmética e para monitorar a alocação, a utilização, e a desalocação de memória e para sinalizar vazamentos de memória.

Ferramenta de análise estática

(Static analysis tool)

Ver analista estático.

Ferramenta de captura e reprodução

(Capture/playback tool)

Tipo de ferramenta de execução de teste onde as entradas/inputs são gravadas durante o teste manual a fim de gerar roteiros de testes automatizados que possam ser executados mais tarde (ou seja, reproduzidos novamente). Essas ferramentas são freqüentemente utilizadas para dar suporte aos testes de regressão automatizada.

Ferramenta de captura/replay

(Capture/replay tool)

Ver ferramenta de captura e reprodução.

Ferramenta de cobertura

(Coverage tool)

Ferramenta que fornece medidas objetivas de quais elementos estruturais, por exemplo, sentenças ou desvios, foram exercitados por uma suíte de teste.

Ferramenta de debug

(Debugging tool)

Ferramenta utilizada por programadores para reproduzir falhas, investigar o estado dos programas e encontrar o defeito correspondente. Os debuggers possibilitam aos programadores executar os programas passo a passo, parar um programa, qualquer que seja a sentença, e estabelecer e examinar variáveis de programas.

Ferramenta de execução de teste

(Test execution tool)

Tipo de ferramenta de teste que pode executar outro software utilizando um roteiro de teste automatizado, ex. captura/recuperação. [Fewster e Graham]

Ferramenta de gerenciamento de configuração

(Configuration management tool)

Ferramenta que dá suporte à identificação e ao controle dos itens de configuração, ao seu estado durante alterações e versões e à liberação/release de linhas de base que consistem de itens de configuração.

Ferramenta de gerenciamento de defeito

(Defect management tool)

Ferramenta que facilita a gravação e o rastreamento de condição de defeitos. Possuem freqüentemente recursos orientados para o fluxo de trabalho a fim de rastrear e controlar a alocação, a correção e a nova realização de testes de defeitos, além de fornecerem recursos para a elaboração de relatórios. Ver também incident management tool (ferramenta de gerenciamento de incidentes).

Ferramenta de gerenciamento de incidente

(Incident management tool)

Ferramenta que facilita o registro e o rastreamento de condição de incidentes. Freqüentemente possui recursos orientados para o fluxo de trabalho para rastrear e controlar a alocação, correção e nova realização de testes de incidentes, além de fornecer recursos para relatório. Ver também ferramenta para gerenciamento de defeito.

Ferramenta de gerenciamento de teste

(Test management tool)

Ferramenta que dá suporte ao gerenciamento de teste e que controla parte deste processo. Freqüentemente possui várias capacidades, tais como, gerenciamento de testware, estabelecimento de um cronograma de testes, registro dos resultados, rastreamento do progresso, gerenciamento de incidentes e relato de teste.

Ferramenta de gravação/recuperação

(Record/playback tool)

Ver ferramenta de captura e recuperação.

Ferramenta de modelagem de teste

(Test design tool)

Ferramenta que dá suporte à atividade de modelagem de teste por meio da geração de entradas/inputs de teste a partir de uma especificação que pode estar armazenada em um repositório de ferramenta CASE, por exemplo: ferramenta de gerenciamento de requisitos a partir de condições de teste especificadas armazenadas na ferramenta em si ou em um código.

Ferramenta de monitoramento

(Monitoring tool)

Ver monitor.

Ferramenta de preparação de dados de teste

(Test data preparation tool)

Tipo de ferramenta de teste que possibilita que os dados sejam selecionados dos bancos de dados existentes ou que sejam criados, gerados, manipulados e editados para uso no teste.

Ferramenta de rastreamento de defeito

(Defect tracking tool)

Ver ferramenta de gerenciamento de defeito.

Ferramenta de revisão

(Review tool)

Ferramenta que dá suporte ao processo de revisão. Suas características normalmente incluem o planejamento da revisão e o suporte ao rastreamento, assim como suporte às comunicações, revisões colaborativas e um repositório para coletar e relatar as métricas.

Ferramenta de teste

(Test tool)

Produto de software que dá suporte a uma ou mais atividades de um teste, entre elas, planejamento e controle, especificação, construção de arquivos iniciais e dados, execução e análise de testes. [TMap]. Ver também CAST.

Ferramenta de teste de desempenho

(Performance testing tool)

Ferramenta que dá suporte ao teste de desempenho e que, normalmente, tem dois recursos principais de medição de geração de carga e de transação de teste. A geração de carga pode simular tanto os usuários múltiplos como os altos volumes de dados de entrada/input. Durante a execução, as medições dos tempos de resposta são feitas a partir de transações selecionadas e depois registradas. Normalmente, as ferramentas de teste de desempenho fornecem relatórios baseados nos registros e grafos de testes da carga em relação aos tempos de resposta.

Ferramenta de teste de segurança - de acesso

(Security testing tool)

Ferramenta que dá suporte para o teste das características e vulnerabilidades de segurança.

Ferramenta para gerenciamento dos requisitos

(Requirements management tool)

Ferramenta que dá suporte à gravação de requisitos e de atributos de requisitos (ex. prioridade e responsável pelo conhecimento, entre outros) e à anotação. Ela também facilita o rastreamento por meio de camadas de requisitos e de gerenciamento de alteração de requisitos. Algumas ferramentas de gerenciamento de requisitos também oferecem recursos para análise estática, tais como verificação de consistência e violações de regras de requisitos predefinidas.

Fluxo de controle

(Control flow)

Seqüência de eventos (caminhos) na execução de um componente ou sistema.

Fluxo de dados

(Data flow)

Representação abstrata da seqüência e das possíveis alterações do estado dos objetos de dados, na qual o estado de um objeto pode ser de criação, utilização ou destruição. [Beizer]

Funcionalidade

(Functionality)

Capacidade do produto de software de oferecer funções que atendam às necessidades declaradas ou implícitas quando utilizado sob condições específicas. [ISO 9126]

G

Garantia de qualidade

(Quality assurance)

Parte do gerenciamento de qualidade que garante que os requisitos de qualidade sejam atendidos. [ISO 9000]

Gerador de teste

(Test generator)

Ver ferramenta de preparação de dados de teste.

Gerenciamento de configuração

(Configuration management)

Disciplina que aplica direção e vigilância técnica e administrativa para identificar e documentar as características funcionais e físicas de um item de configuração, controlar as alterações feitas a essas características, gravar e relatar o estado do processo de alteração e implementação e verificar o atendimento aos requisitos especificados. [IEEE 610]

Gerenciamento de defeito

(Defect management)

Processo que reconhece, investiga, toma medidas e resolve defeitos. Envolve a gravação dos defeitos, sua classificação e identificação de impactos. [Subsequente IEEE 1044]

Gerenciamento de incidente

(Incident management)

Processo que reconhece, investiga, toma medidas e soluciona incidentes. Envolve a gravação de incidentes, sua classificação e identificação do impacto. [Subsequente a IEEE 1044]

Gerenciamento de problema

(Problem management)

Ver gerenciamento de defeito.

Gerenciamento de qualidade

(Quality management)

Atividades coordenadas para dirigir e controlar a qualidade em uma organização. Direção e controle de qualidade geralmente incluem o estabelecimento da política e dos objetivos de qualidade, assim como o planejamento, o controle, a garantia e a melhoria da qualidade. [ISO 9000]

Gerenciamento de risco

(Risk management)

Aplicação sistemática de procedimentos e práticas à tarefa de identificar, analisar, priorizar e controlar riscos.

Gerenciamento de teste

(Test management)

Planejamento, estimativa, monitoramento e controle das atividades de teste, normalmente conduzidos pelo gerente de teste.

Gerente de teste

(Test manager)

Pessoa responsável pelo gerenciamento do projeto, pelas atividades e recursos de teste e por avaliar o objeto de teste. É o indivíduo que dirige, controla, administra, planeja e regula a avaliação de um objeto de teste.

Gráfico de causa e efeito

(Cause-effect graphing)

Técnica de modelagem de teste de caixa preta na qual os casos de teste são modelados a partir de grafos de causa e efeito. [BS 7925/2]

Grafo de causa e efeito

(Cause-effect graph)

Representação gráfica das entradas/inputs e/ou dos estímulos (causas) com seus resultados/saídas/outputs (efeitos) associados, os quais podem ser utilizados para projetar casos de teste.

Grafo do fluxo de controle

(Control flow graph)

Ver caminho.

Gravação de teste

(Test recording)

Ver registrar teste.

Gravar teste

(Test record)

Ver registro de teste.

Guia de instalação

(Installation guide)

Instruções fornecidas por qualquer mídia adequada que guiam uma pessoa durante o processo de instalação. Pode ser um guia manual, um procedimento passo a passo, um assistente de instalação ou qualquer outro processo similar.

I**Identificação de configuração**

(Configuration identification)

Elemento de gerenciamento de configuração que consiste em selecionar os itens de configuração para um sistema e em gravar suas características funcionais e físicas em documentação técnica. [IEEE 610]

Identificação de risco

(Risk identification)

Processo que identifica os riscos por meio de técnicas como brainstorming, listas de verificação ou histórico de falhas.

Incidente

(Incident)

Qualquer evento que requeira investigação. [Subsequente a IEEE 1008]

Incidente de teste

(Test incident)

Ver incidente.

Incidente de teste de software

(Software test incident)

Ver incidente.

Independência

(Independence)

Separação de responsabilidades que estimula o alcance do teste de objetivo. [Subseqüente a DO-178b]

Indicador de desempenho

(Performance indicator)

Métrica de nível alto de eficácia e/ou eficiência utilizada para guiar e controlar o desenvolvimento progressivo, por exemplo, deslizamentos no acompanhamento da linha do tempo do projeto no desenvolvimento de software. [CMMI]

Indicador de desempenho de teste

(Test performance indicator)

Métrica de alto nível de eficácia e/ou eficiência utilizada para guiar e controlar o desenvolvimento progressivo de teste, ex. Defect Detection Percentage - DDP (Porcentagem de Detecção de Defeito).

Infra-estrutura de teste

(Test infrastructure)

Artefatos organizacionais necessário para realizar os testes. Eles consistem em ambientes de teste, ferramentas de teste, ambiente de escritório e procedimentos.

Injeção de erros

(Error seeding)

Nome dado ao processo que adiciona, intencionalmente, defeitos conhecidos à aqueles já presentes no componente ou sistema com a finalidade de monitorar a taxa de detecção ou remoção e de estimar o número de defeitos remanescentes. [IEEE 610]

Inspeção

(Inspection)

Revisão realizada pelos pares para detectar defeitos e baseada no exame visual de documentos, por exemplo, violações dos padrões de desenvolvimento e não conformidade da documentação de nível mais alto. Trata-se da técnica de revisão mais formal e, portanto, está sempre baseada em um procedimento documentado. [Subseqüente a IEEE 610 e a IEEE 1028]. Ver também revisão pelos pares.

Inspetor

(Inspector)

Ver revisor.

Instalabilidade

(Installability)

Capacidade que um produto de software tem para ser instalado em um ambiente específico [ISO 9126]. Ver também portabilidade.

Instrumentação

(Instrumentation)

Inserção de um código adicional no programa a fim de coletar informações sobre o comportamento do programa durante sua execução, por exemplo, para medir a cobertura de código.

Instrumentador

(Instrumenter)

Ferramenta de software utilizada para proceder a instrumentação.

Instrumentador de programa

(Program instrumenter)

Ver instrumentador.

Integração

(Integration)

Processo que combina componentes ou sistemas em componentes maiores.

Integração do Modelo de Maturidade de Capacidade

(Capability Maturity Model Integration (CMMI))

Estrutura que descreve os principais elementos de desenvolvimento eficaz de produto e do processo de manutenção. A Integração do Modelo de Maturidade de Capacidade cobre as melhores práticas para planejamento, engenharia e gerenciamento de desenvolvimento e manutenção de produto. O CMMI é o sucessor designado do CMM. [CMMI]

Integração funcional

(Functional integration)

Abordagem de integração que combina os componentes ou sistemas com a finalidade de fazer com que a funcionalidade básica funcione o mais rápido possível. Ver também teste de integração.

Interoperabilidade

(Interoperability)

Capacidade de um produto de software para interagir com um ou mais componentes ou sistemas específicos. [Subsequente a ISO 9126]. Ver também funcionalidade.

Inventário de Medição de Usabilidade de Software

(Software Usability Measurement Inventory (SUMI))

Questionário baseado em uma técnica de teste de usabilidade que avalia a usabilidade, ex. satisfação do usuário, de um componente ou sistema. [Veenendaal]

Item de cobertura

(Coverage item)

Entidade ou propriedade utilizada como base para a cobertura de teste, por exemplo, partições de equivalência ou sentenças de código.

Item de configuração

(Configuration item)

Trata-se de uma agregação de hardware, software, ou de ambos, modelada para o gerenciamento de configuração e tratada como uma única entidade do processo de gerenciamento de configuração. [IEEE 610]

Item de teste

(Test item)

Elemento individual a ser testado. Normalmente, há um objeto de teste e vários itens de teste. Ver também objeto de teste.

L

Líder de inspeção

(Inspection leader)

Ver moderador.

Líder de teste

(Test leader)

Ver gerente de teste.

Linguagem de script

(Scripting language)

Linguagem de programação nas quais os scripts de testes executáveis são escritos e utilizados por uma ferramenta de execução de testes (ex. ferramenta de captura/recuperação).

Linha de Base

(Baseline)

Especificação ou produto de software formalmente revisto ou acordado que servirá de base para futuros desenvolvimentos, e que só poderá ser alterado por meio de um processo formal de controle de alteração.[Subsequente a IEEE 610].

M

Mantenabilidade

(Maintainability)

Facilidade com que um produto de software pode ser modificado para corrigir defeitos, atender a novos requisitos, facilitar manutenções futuras ou adaptar-se a um ambiente alterado. [ISO 9126]

Manutenção

(Maintenance)

Modificação feita em um produto de software após sua entrega. Tem por finalidade corrigir defeitos, melhorar o desempenho ou outros atributos, ou adaptar o produto ao ambiente modificado. [IEEE 1219]

Máquina de estado finito

(Finite state machine)

Modelo computacional que consiste em um número finito de estados e de transições entre esses estados, possivelmente com ações de acompanhamento. [IEEE 610]

Marco

(Milestone)

Determinado ponto de um projeto no qual os “entregáveis” definidos (intermediários) e os resultados devem estar prontos.

Mascaramento de defeito

(Defect masking)

Ocorrência na qual um defeito evita a detecção de outros. [Subsequente a IEEE 610]

Mascaramento de faltas

(Fault masking)

Ver mascaramento de defeito.

Maturidade

(Maturity)

(1) Grau de eficácia e eficiência de uma organização com relação a seus processos e práticas de trabalho. Ver também Modelo de Maturidade de Capacidade e Modelo de Maturidade de Teste. (2) Capacidade do produto de software de evitar falhas resultantes de defeitos no software. [ISO 9126]. Ver também confiabilidade.

Medição

(Measurement)

Processo que designa um número ou categoria a uma entidade a fim de descrever determinado atributo de tal entidade. [ISO 14598]

Medida

(Measure)

Número ou categoria designada a um atributo de uma entidade por meio de medição. [ISO 14598]

Melhor prática

(Best practice)

Método superior ou prática inovadora que contribui para o aprimoramento do desempenho de uma organização em um determinado contexto, normalmente, reconhecido como o “melhor” por outras organizações similares.

Melhoria do Processo de Teste

(Test Process Improvement - TPI)

Estrutura contínua para a melhoria do processo de teste que descreve os principais elementos de um processo de teste eficaz. Especialmente voltada para o teste de sistema e para o teste de aceitação.

Método de classificação por árvore

(Classification tree method)

Técnica de modelagem de teste de caixa preta na qual os casos de teste, descritos por uma árvore de classificação, são modelados para executar combinações de domínios de entrada/input e/ou de saída/output representativos. [Grochtmann]

Métrica

(Metric)

Escala e método utilizados para medição. [ISO 14598]

Métrica de cobertura de Chow

(Chow's coverage metrics)

Ver cobertura N-switch. [Chow]

Mitigação de risco

(Risk mitigation)

Ver controle de risco.

Mode de falha e Análise de Efeito

(Failure Mode and Effect Analysis (FMEA))

Abordagem sistemática da identificação de risco e da análise da identificação de possíveis modos de falha e das tentativas de prevenção de sua ocorrência.

Modelagem de teste

(Test design)

Ver especificação de modelagem de teste.

Modelo de desenvolvimento incremental

(Incremental development model)

Ciclo de vida de desenvolvimento que divide um projeto em uma série de incrementos, cada um dos quais entrega uma porção da funcionalidade dos requisitos gerais do projeto. Os requisitos são priorizados e entregues de acordo com a ordem de prioridade no incremento apropriado. Em algumas (mas, não em todas) versões deste modelo de ciclo de vida, cada subprojeto segue um “mini modelo V”, com suas próprias fases de modelagem, codificação e teste.

Modelo de desenvolvimento iterativo

(Iterative development model)

Ciclo de vida de desenvolvimento no qual um projeto é normalmente dividido em iterações. Dá-se o nome de interação a um laço (loop) de desenvolvimento completo que resulta em um release (liberação) (interna ou externa) de um produto executável, num subconjunto do produto final em desenvolvimento, o qual cresce de interação em interação até tornar-se o produto final.

Modelo de Maturidade de Capacidade

(Capability Maturity Model (CMM))

Estrutura de cinco níveis que descreve os principais elementos de um processo de software eficaz. O Modelo de Maturidade de Capacidade cobre as melhores práticas para planejamento, engenharia e gerenciamento da manutenção e desenvolvimento de software. [CMM]

Modelo de Maturidade de Teste

(Test Maturity Model - TMM)

Estrutura de cinco níveis utilizada para a melhoria do processo de teste relaciona-se ao Modelo de Maturidade de Capacidade (CMM) que descreve os principais elementos de um processo eficaz de teste.

Modelo V

(V-model)

Estrutura que descreve as atividades do ciclo de vida do desenvolvimento de um software, desde a especificação de requisitos até a manutenção. O modelo V ilustra como as atividades de teste podem ser integradas em cada fase do ciclo de vida do desenvolvimento de um software.

Moderador

(Moderator)

É o líder ou o principal responsável pela inspeção ou por outro processo de revisão.

Modified multiple condition testing

(Modified multiple condition testing)

Ver teste de cobertura de determinação de condição.

Modo de falha

(Failure mode)

Manifestação física ou funcional de uma falha. Por exemplo, um sistema em modo de falha pode ser caracterizado pela operação lenta, saída/output incorreto ou pelo término completo da execução.

Módulo

(Module)

Ver componente.

Monitor

(Monitor)

Ferramenta de software ou dispositivo de hardware, executado simultaneamente com o componente/sistema sendo testado, e que supervisiona, grava e/ou analisa o comportamento do componente/sistema. [Subsequente a IEEE 610]

Monitoramento de teste

(Test monitoring)

Tarefa do gerenciamento de testes que lida com as atividades relacionadas às verificações periódicas da condição de um projeto de teste. São preparados relatórios para comparar os resultados reais e os planejados. Ver também gerenciamento de teste.

N

Não conformidade

(Non-conformity)

Trata-se do não atendimento a requisito especificado. [ISO 9000]

Nível de teste

(Test level)

Grupo de atividades de teste organizadas e gerenciadas conjuntamente. Um nível de teste está ligado às responsabilidades do projeto. Podemos citar como exemplos teste de componente, teste de integração, teste de sistema e teste de aceitação. [Subsequente a TMap]

Nota de lançamento

(Release note)

Documento que identifica os itens de um teste, suas configurações, estado atual e outras informações de entrega entregues pelo desenvolvimento para serem testados, e, possivelmente outras informações de interessados/stakeholders, no início da fase de execução do teste. [Subsequente a IEEE 829]

Número ciclomático

(Cyclomatic number)

Ver complexidade ciclomática.

O

Objetivo de teste

(Test objective)

Razão ou finalidade por trás da modelagem e da execução de um teste.

Objeto de teste

(Test object)

Componente ou sistema a ser testado. Ver também item de teste.

Operabilidade

(Operability)

Capacidade que um produto de software tem para possibilitar ao usuário operá-lo e controlá-lo. [ISO 9126]. Ver também usabilidade.

Oráculo

(Oracle)

Ver oráculo de teste.

Oráculo de teste

(Test oracle)

Fonte utilizada para determinar os resultados esperados e compara-los com os resultados reais produzidos pelo software em teste. Um oráculo pode ser um sistema existente (para um benchmark), um manual de usuário ou o conhecimento especializado de um indivíduo, porém, não deve ser o código. [Subsequente a Adrion]

P

Pacote diário

(Daily build)

Atividade de desenvolvimento na qual um sistema completo é compilado e “linkado” (link) diariamente (normalmente durante a noite), para que um sistema consistente, contendo as últimas alterações, esteja disponível a qualquer momento.

Painel de teste

(Test charter)

Sentença dos objetivos de um teste e de possíveis idéias sobre como realizar o teste. Os painéis de teste são freqüentemente utilizados em testes exploratórios. Ver também teste exploratório.

Par de definição-utilização

(Definition-use pair)

Associação da definição de uma variável com a utilização desta variável. Os usos das variáveis incluem computacional (ex. multiplicação) ou para dirigir a execução de um caminho (uso “predicado”).

Partição de equivalência

(Equivalence partition)

Parte de um domínio de entrada/input ou de saída/output para o qual se presume, com base na especificação, que o comportamento de um componente ou sistema seja o mesmo.

Particionamento de equivalência

(Equivalence partitioning)

Técnica de modelagem de teste de caixa preta na qual os casos de testes são modelados para executar a partir de partições de equivalência. Em princípio, os casos de teste são modelados para cobrir cada partição pelo menos uma vez.

Planejamento de teste

(Test planning)

Atividade relacionada ao estabelecimento ou à atualização de um plano de teste.

Plano de projeto de teste

(Project teste plan)

Ver plano de teste mestre.

Plano de teste

(Test plan)

Documento que descreve o escopo, a abordagem, os recursos e o cronograma das atividades de testes. Identificam, entre outros itens de teste, as características a serem testadas, as tarefas e seus executores, o grau de independência do testador, o ambiente, as técnicas de modelagem de teste, critérios de entrada e saída a serem utilizados, assim como os fundamentos lógicos de sua escolha e quaisquer riscos que requeiram planejamento de contingência. É o registro do processo de planejamento. [Subsequente a IEEE 829]

Plano de teste de fase

(Phase test plan)

Plano de teste que normalmente aborda uma fase de teste. Ver também plano de teste.

Plano de teste de nível

(Level test plan)

Plano de teste que aborda um nível de teste. Ver também plano de teste.

Plano de teste mestre

(Master test plan)

Plano de teste que aborda múltiplos níveis de teste. Ver também plano de teste.

Política de teste

(Test policy)

Documento de alto nível que descreve os princípios, a abordagem e os principais objetivos da organização de um teste.

Ponto de entrada

(Entry point)

Primeira sentença executável dentro de um componente.

Ponto de saída

(Exit point)

Última sentença executável dentro de um componente.

Porcentagem de Detecção de Defeitos

(Defect Detection Percentage (DDP))

Número de defeitos encontrados por uma fase de teste, dividido pelo número encontrado pela fase de teste e por qualquer outro meio depois disto.

Porcentagem de Detecção de Falhas

(Fault Detection Percentage (FDP))

Ver Porcentagem de Detecção de Defeitos.

Portabilidade

(Portability)

Facilidade com que o produto de software pode ser transferido de um ambiente de hardware ou software para outro. [ISO 9126]

Pós-condição

(Postcondition)

Condições de ambiente e de estado que devem ser atendidas após a execução de um teste ou de um procedimento de teste.

Precondição

(Precondition)

Condições de ambiente e de estado que devem ser atendidas antes que um componente ou sistema possa ser executado com um determinado teste ou procedimento de teste.

Pré-teste

(Pretest)

Ver teste de intake.

Principal indicador de desempenho

(Key performance indicator)

Ver indicador de desempenho.

Prioridade

(Priority)

Nível de importância (do negócio) designado a um item, por exemplo, defeito.

Problema

(Problem)

Ver defeito.

Procedimento (roteiro) de teste

(Test procedure)

Ver especificação de procedimento de teste.

Processo

(Process)

Conjunto de atividades inter-relacionadas que transforma entradas/inputs em saídas/outputs. [ISO 12207]

Processo de teste

(Test process)

O processo de teste compreende fundamentalmente o planejamento, a especificação, a execução, o registro, a verificação da finalização e as atividades de fechamento. [Subsequente a BS 7925/2]

Programação em pares

(Pair programming)

Abordagem de desenvolvimento de software pela qual as linhas de código (produção e/ou teste) de um componente são escritas por dois programadores trabalhando em um único computador. Implicitamente, isto significa que revisões de código em tempo real são realizadas.

Projeto

(Project)

Conjunto único de atividades coordenadas e controladas, com datas para início e término, realizado com a finalidade de alcançar um objetivo que atenda aos requisitos específicos, inclusive os de restrição de tempo, custos e recursos. [ISO 9000]

Pseudo aleatório

(Pseudo-random)

Série que parece ser aleatória, mas que, de fato, foi gerada de acordo com alguma seqüência preestabelecida.

Q

Qualidade

(Quality)

Grau até o qual um componente, sistema ou processo atende aos requisitos especificados e/ou às necessidades e expectativas do usuário/consumidor. [Subsequente IEEE 610]

Qualidade do software

(Software quality)

Totalidade das funcionalidades e características de um produto de software que se baseia na sua habilidade de satisfazer as necessidades declaradas ou implícitas. [Subsequente a ISO 9126]

R

Rastreabilidade

(Traceability)

Habilidade de identificar itens relacionados em documentos e em software, ex. requisitos e testes associados. Ver também rastreabilidade horizontal e rastreabilidade vertical.

Rastreabilidade horizontal

(Horizontal traceability)

Rastreamento dos requisitos para um nível de teste por meio de camadas de documentação de testes (por exemplo, plano de teste e especificação de modelagem, caso de teste, procedimento ou para roteiro de teste).

Rastreabilidade vertical

(Vertical traceability)

Rastreamento de requisitos por meio de camadas de documentação de desenvolvimento dos componentes.

Recuperabilidade

(Recoverability)

Capacidade de um produto de software para estabelecer novamente um nível específico de desempenho e de recuperar os dados diretamente afetados em caso de falha. [ISO 9126]. Ver também confiabilidade.

Redator

(Scribe)

Nome dado à pessoa que registra cada defeito mencionado ou sugestão dada para a melhoria do processo durante uma reunião de revisão, em um formulário de registro. O redator deve assegurar que o formulário de registro possa ser lido e entendido.

Registrador

(Recorder)

Ver redator

Registrar incidente

(Incident logging)

Gravação dos detalhes de qualquer incidente ocorrido, por exemplo, durante o teste.

Registrar teste

(Test logging)

Processo pelo qual as informações sobre o teste executado são registradas em um registro de teste.

Registro da execução do teste

(Test run log)

Ver test log (registro do teste).

Registro de teste

(Test log)

Registro cronológico dos detalhes relevantes da execução dos testes. [IEEE 829]

Relato de incidente

(Incident report)

Documento que relata qualquer evento ocorrido, por exemplo, durante o teste, que requeira investigação. [Subsequente a IEEE 829]

Relatório de avaliação de teste

(Test evaluation report)

Documento produzido ao final do processo de teste e que resume todas as atividades de teste e seus resultados. Contém também uma avaliação do processo de teste e as lições aprendidas no referido processo.

Relatório de defeito

(Defect report)

Documento que relata qualquer falha em um componente ou sistema que possa fazer com este componente ou sistema deixe de desempenhar sua função requisitada. [Subsequente a IEEE 829]

Relatório de desvio

(Deviation report)

Ver relatório de incidente.

Relatório de incidente de teste

(Test incident report)

Ver relatório de incidente.

Relatório de incidente de teste de software

(Software test incident report)

Ver relatório de incidente.

Relatório de problema

(Problem report)

Ver relatório de defeito.

Relatório de resumo de teste

(Test summary report)

Documento que resume as atividades e os resultados de um teste. Contém também uma avaliação dos itens de teste correspondentes versus os critérios de saída. [Subsequente a IEEE 829]

Relatório de teste

(Test report)

Ver relatório do resumo de teste.

Relatório de transmissão de item

(Item transmittal report)

Ver nota de liberação.

Relatório de transmissão de item

(Test item transmittal report)

Ver nota de lançamento.

Reproducibilidade de teste

(Test reproduceability)

Atributo de um teste que indica se os mesmos resultados são produzidos a cada vez que o teste é executado.

Reprovação

(Fail)

Um teste será reprovado se seu resultado real não for o resultado esperado.

Reprovação de teste

(Test fail)

Ver reprovação.

Requisito

(Requirement)

Condição ou capacidade que um usuário precisa ter para resolver um problema ou alcançar um objetivo que deve ser necessariamente alcançado ou possuído por um sistema, ou por um componente de sistema, a fim de satisfazer um contrato, padrão, especificação ou um outro documento formalmente exigido. [Subseqüente a IEEE 610]

Requisito de teste

(Test requirement)

Ver condição de teste.

Requisito funcional

(Functional requirement)

Especifica determinada função que um componente ou sistema deve desempenhar. [IEEE 610]

Requisito não funcional

(Non-functional requirement)

Requisito que não se refere à funcionalidade, mas sim aos atributos como confiabilidade, eficiência, usabilidade, manutenibilidade e portabilidade.

Requisitos testáveis

(Testable requirements)

Grau até o qual um requisito se declara em termos que permitam o estabelecimento da modelagem de teste (e subseqüentemente de casos de teste) e a execução de testes para determinar se os requisitos foram atendidos. [Subseqüente a IEEE 610]

Resultado

(Result)

Conseqüência da execução de um teste. Inclui saídas/outputs para telas, alterações de dados, relatórios e comunicações enviadas. Ver também resultado real e resultado esperado.

Resultado de decisão

(Decision outcome)

Resultado de uma decisão (o qual, por sua vez, determina os branches a serem tomados).

Resultado de teste

(Test result)

Ver resultado.

Resultado esperado

(Expected result)

Comportamento previsto pela especificação, ou por outra fonte, de um componente ou sistema sob determinadas condições.

Resultado real

(Actual result)

Comportamento produzido/observado quando um componente ou sistema é testado.

Re-teste

(Re-testing)

Teste que executa casos de teste reprovados durante sua última execução. Este procedimento é feito para verificar o sucesso das ações corretivas.

Revisão

(Review)

Avaliação das condições de um produto ou projeto para averiguar discrepância em relação aos resultados planejados e para recomendar melhorias. Como exemplos de revisão, podemos citar as revisões de gerenciamento, as revisões informais, revisões técnicas, as inspeções e os acompanhamentos (walkthroughs). [Subseqüente a IEEE 1028]

Revisão ad hoc

(Ad hoc review)

Ver revisão informal

Revisão de gerenciamento

(Management review)

Avaliação sistemática da aquisição de software, do fornecimento, do desenvolvimento, da operação ou do processo de manutenção. Tal avaliação pode ser feita pelo gerenciamento, ou em seu nome, a fim de monitorar o progresso, determinar o estado dos planos e dos cronogramas, confirmarem os requisitos e seus sistemas de alocação ou de avaliar a eficácia da abordagem de gerenciamento para fins de otimização. [Subseqüente a IEEE 610 e IEEE 1028]

Revisão de testabilidade

(Testability review)

Verificação detalhada das bases de um teste a fim de determinar se a base de teste está num nível adequado de qualidade para agir como documento de entrada/input do processo de teste. [Subsequente a TMap]

Revisão formal

(Formal review)

Revisão caracterizada por procedimentos e requisitos documentados, por exemplo, inspeção.

Revisão informal

(Informal review)

Revisão que não é baseada em um procedimento formal (documentado).

Revisão pelo pares

(Peer review)

Revisão do trabalho de um produto de software feita por colegas do produtor do produto, com a finalidade de identificar defeitos e apontar melhorias. Como exemplo, podemos citar a inspeção, a revisão técnica e o acompanhamento/walkthrough.

Revisão técnica

(Technical review)

Discussão realizada entre pares e que busca o consenso sobre o tipo de abordagem técnica a ser utilizada. [Gilb e Graham, IEEE 1028]. Ver também revisão pelos pares.

Revisor

(Reviewer)

Pessoa envolvida no processo de revisão e que identifica e descreve as anomalias encontradas no produto ou projeto sendo revisto. Os revisores podem ser escolhidos para representar diferentes pontos de vista e papéis neste processo.

Risco

(Risk)

Fator que pode resultar em conseqüências futuras negativas; normalmente expresso em termos de impacto e possibilidade.

Risco de produto

(Product risk)

Risco diretamente relacionado ao objeto do teste. Ver também risco.

Risco de projeto

(Project risk)

Risco relacionado ao gerenciamento e ao controle do projeto (teste). Ver também risco.

Robustez

(Robustness)

Grau até o qual um componente ou sistema pode funcionar corretamente na presença de entradas inválidas ou de condições ambientais estressantes. [IEEE 610]. Ver também tolerância ao erro e tolerância à falta.

S

Saída

(Output)

Variável (armazenada dentro ou fora de um componente) escrita por um componente.

Saída da condição

(Condition outcome)

Avaliação de uma condição em termos de Verdadeiro ou Falso.

Saída esperada

(Expected outcome)

Ver resultado esperado.

Saída real

(Actual outcome)

Ver resultado real.

Script de teste

(Test script)

Termo normalmente utilizado para se referir à uma especificação de procedimento de teste, especialmente em testes automatizados.

Segurança

(Safety)

Capacidade que um produto de software tem para alcançar níveis de risco aceitáveis tanto para pessoas como negócios, software, propriedade ou meio ambiente dentro de um contexto de utilização específico. [ISO 9126]

Segurança de acesso

(Security)

Atributos dos produtos de software que se baseiam em sua habilidade de evitar o acesso não autorizado, acidental ou deliberado, aos programas e dados. [ISO 9126]. Ver também funcionalidade.

Sensibilização de caminho

(Path sensitizing)

Escolha de um conjunto de valores de entrada/input para forçar a execução de um dado caminho.

Sentença (comando)

(Statement)

Entidade de uma linguagem de computador que é, normalmente, a menor unidade indivisível da execução.

Sentença de fonte

(Source statement)

Ver sentença (comando).

Sentença executável

(Executable statement)

Sentença que, quando compilada, é traduzida em um código de objeto e que será executada por meio de procedimentos quando o programa estiver sendo executado, também poderá desempenhar uma ação nos dados.

Seqüência de Código Linear e Salto

(LCSAJ - Linear Code Sequence and Jump)

Consiste nos três itens seguintes (convencionalmente identificados por número de linhas em uma lista de código fonte) início da seqüência linear em sentenças executáveis, fim da seqüência linear e linha alvo para a qual o fluxo de controle é transferido ao final da seqüência linear.

Severidade

(Severity)

Grau de impacto que um defeito tem sobre o desenvolvimento ou operação de um componente ou sistema. [Subsequente a IEEE 610]

Simulação

(Simulation)

Representação de características comportamentais selecionadas de um sistema físico ou abstrato por outro sistema. [ISO 2382/1]

Simulador

(Simulator)

Dispositivo, programa de computador ou sistema utilizado durante o teste e que se comporta ou opera como um dado sistema quando recebe um conjunto de entradas/inputs controladas. [Subsequente a IEEE 610 e a DO178b]. Ver também emulador.

Simulador

(Stub)

Implementação estrutural ou com um propósito especial de um componente de software utilizada para desenvolver ou testar um componente que chama ou é de outra forma dependente de ele. Substitui um componente chamado. [Subseqüente a IEEE 610]

Sistema

(System)

Coleção de componentes organizada para alcançar uma função específica ou um conjunto de funções. [IEEE 610]

Situação de teste

(Test situation)

Ver condição de teste.

Software

(Software)

Programas e procedimentos de computação, e possível documentação associada e dados pertinentes à operação de um sistema de computador. [IEEE 610]

Software bespoke

(Bespoke software)

Software desenvolvido especificamente para um conjunto de usuários ou clientes. Seu oposto é o software de prateleira.

Software comercial de prateleira

(Commercial off-the-shelf software)

Ver software de prateleira.

Software de prateleira

(Off-the-shelf-software)

Produto de software desenvolvido para o mercado em geral, ou seja, para um grande número de consumidores, e entregue a vários consumidores num mesmo formato.

Software de sob medida

(Custom software)

Ver software bespoke.

Software padrão

(Standard software)

Ver software de prateleira.

Subcaminho

(Subpath)

Seqüência de sentenças executáveis dentro de um componente.

Substitutibilidade

(Replaceability)

Capacidade que um produto de software tem para ser utilizado no lugar de outro produto de software específico para os mesmos fins e no mesmo ambiente. [ISO 9126]. Ver também portabilidade.

Suite de caso de teste

(Test case suite)

Ver suite de teste.

Suíte de teste

(Test suite)

Conjunto de vários casos de teste para um componente ou sistema sendo testado, no qual a pós-condição de um teste é freqüentemente utilizada como pré-condição para o próximo.

Suposição de erro

(Error guessing)

Técnica de modelagem de teste na qual a experiência do testador é utilizada para antecipar possíveis defeitos em um componente ou sistema em teste como resultado de erros feitos. Também utilizada para projetar testes específicos para expor tais defeitos.

T

Tabela de decisão

(Decision table)

Tabela que mostra as combinações de entradas/inputs e/ou de estímulos (causas) com os outputs/saídas e/ou ações (efeitos) associados, os quais podem ser utilizados para projetar casos de teste.

Tabela de decisão de causa e efeito

(Cause-effect decision table)

Ver tabela de decisão.

Tabela de estado

(State table)

Grade que mostra as transições resultantes para cada estado combinado versus evento possível, exhibe tanto as transições válidas como as inválidas.

Taxa de falha

(Failure rate)

Razão do número de falhas de uma dada categoria para uma dada unidade de medida, por exemplo, falhas por unidade de tempo, número de transações, número de execução de computadores. [IEEE 610]

Técnica de caixa-preta

(Black-box technique)

Ver técnica de modelagem de teste de caixa preta.

Técnica de especificação de teste

(Test specification technique)

Ver técnica de modelagem de teste.

Técnica de execução de teste

(Test execution technique)

Método utilizado para desempenhar a real execução do teste, seja manual ou automaticamente.

Técnica de modelagem de caso de teste

(Test case design technique)

Ver técnica de modelagem de teste.

Técnica de modelagem de teste

(Test design technique)

Procedimento utilizado para derivar e/ou selecionar casos de teste.

Técnica de modelagem de teste baseado em especificação

(Specification-based test design technique)

Ver técnica de modelagem de teste de caixa preta.

Técnica de modelagem de teste baseado em experiência

(Experienced-based test design technique)

Procedimento utilizado para derivar e/ou selecionar casos de testes baseado na experiência, no conhecimento e na intuição do testador.

Técnica de modelagem de teste de caixa branca

(White-box test design technique)

Procedimento para derivar e/ou selecionar casos de teste baseado em uma análise da estrutura interna de um componente ou sistema.

Técnica de modelagem de teste de caixa preta

(Black-box testing design technique)

Procedimento para derivar e/ou selecionar casos de teste baseado em uma análise de especificação, funcional ou não funcional, de um componente ou de um sistema sem referência à sua estrutura interna.

Técnica de modelagem de teste estrutural

(Structural test design technique)

Ver técnica de modelagem de teste de caixa branca.

Técnica de modelagem de teste funcional

(Functional test design technique)

Procedimento que deriva e/ou seleciona casos de testes com base em uma análise da especificação da funcionalidade de um componente ou sistema sem fazer referência à sua estrutura interna. Ver também técnica de modelagem de teste de caixa preta.

Técnica de modelagem de teste não funcional

(Non-functional test design technique)

Procedimento que deriva e/ou seleciona os casos de teste para teste não funcional. Baseia-se na análise da especificação de um componente ou sistema sem referir-se à sua estrutura interna. Ver também técnica de modelagem de teste de caixa preta.

Técnica de teste

(Test technique)

Ver técnica de modelagem de teste.

Técnicas baseadas em estrutura

(Structure-based techniques)

Ver técnica de modelagem de teste de caixa branca.

Testabilidade

(Testability)

Capacidade de um produto de software que possibilita testar um software modificado. [ISO 9126]. Ver também manutenibilidade.

Testador

(Tester)

Profissional habilitado e envolvido no teste de um componente ou sistema.

Teste

(Test)

Conjunto de um ou mais casos de teste. [IEEE 829]

Testar

(Testing)

Processo que consiste em todas as atividades do ciclo de vida, tanto estáticas quanto dinâmicas, voltadas para o planejamento, preparação e avaliação de produtos de software e produtos de trabalho relacionados a fim de determinar se elas satisfazem os requisitos especificados e demonstrar que estão aptas para sua finalidade e para a detecção de defeitos.

Teste ad hoc

(Ad hoc testing)

Teste conduzido informalmente; não há qualquer tipo de preparação formal nem utilização de qualquer técnica de modelagem de teste. Não há expectativas com relação aos resultados e a arbitrariedade guia a execução do teste.

Teste ágil

(Agile testing)

Prática de teste para um projeto que utiliza metodologias ágeis, tais como a Extreme Programming (XP), e que trata o processo de desenvolvimento como o cliente do processo de teste e enfatiza o paradigma do "test-first design". Ver também desenvolvimento orientado ao teste.

Teste aleatório

(Random testing)

Técnica de modelagem de teste de caixa preta na qual os casos de teste são selecionados, possivelmente por meio de um algoritmo de geração pseudo-aleatória para atender um perfil operacional. Esta técnica pode ser utilizada para testar atributos não funcionais, tais como confiabilidade e desempenho.

Teste baseado em código

(Code-based testing)

Ver teste da caixa branca.

Teste baseado em especificação

(Specification-based testing)

Ver teste de caixa preta.

Teste baseado em modelagem

(Design-base testing)

Abordagem de testes na qual os casos de testes são modelados com base na arquitetura e/ou modelagem detalhada de um componente ou sistema (ex. testes de interfaces entre componentes e sistemas).

Teste baseado em processos de negócios

(Business process-based testing)

Abordagem na qual os casos de teste são modelados com base em descrições e/ou no conhecimento dos processos dos negócios.

Teste baseado em requisitos

(Requirements-based testing)

Abordagem de teste na qual os casos de testes são modelados com base nos objetivos e nas condições de teste derivados dos requisitos, ex. testes que exercitam funções específicas ou investigam atributos não funcionais, tais como confiabilidade ou usabilidade.

Teste baseado em riscos

(Risk-based testing)

Teste orientado para explorar e fornecer informações sobre os riscos de um produto. [Subsequente a Gerrard]

Teste big-bang

(Big-bang test)

Tipo de teste de integração no qual os elementos de um software ou de um hardware, ou ainda de ambos, são combinados todos de uma vez, ao invés de em estágios, em um componente ou sistema geral. [Subsequente ao IEEE 610]. Ver também teste de integração.

Teste bottom-up

(Bottom-up testing)

Abordagem incremental do teste de integração, na qual os componentes de nível mais baixo são testados em primeiro lugar, e, então utilizados para facilitar o teste de componentes de nível mais alto. Este processo é repetido até que o componente no topo da hierarquia seja testado. Ver também integration testing (teste de integração).

Teste completo

(Complete testing)

Ver teste exaustivo.

Teste da eficiência

(Efficiency testing)

Teste que determina a eficiência de um produto de software.

Teste de aceitação de site

(Site acceptance testing)

Teste de aceitação realizado por usuários/consumidores em seu próprio local a fim de determinar se um componente ou sistema satisfaz ou não as necessidades dos usuários/consumidores e se enquadra dentro dos processos de negócios, normalmente inclui hardware e software.

Teste de aceitação de usuário

(User acceptance testing)

Ver teste de aceitação.

Teste de aceite

(Acceptance testing)

Teste formal relacionado às necessidades dos usuários e aos requisitos e processos de negócios. Este teste é realizado para estabelecer se um determinado sistema satisfaz ou não os critérios de aceite e para possibilitar aos usuários, aos clientes e às outras entidades autorizadas decidir aceitar ou não determinado sistema. [Subsequente ao IEEE 610]

Teste de acessibilidade

(Accessibility testing)

Teste que determina a facilidade com a qual usuários portadores de deficiências podem utilizar determinado componente ou sistema. [Gerrard]

Teste de algoritmo

(Algorithm test [TMap])

Ver teste de desvio.

Teste de arco

(Arc testing)

Ver teste de desvio.

Teste de armazenamento

(Storage testing)

Ver teste de utilização de recurso.

Teste de caixa branca

(White-box testing)

Teste baseado na análise da estrutura interna de um componente ou sistema.

Teste de caixa de vidro

(Glass box testing)

Ver teste de caixa branca.

Teste de caixa-preta

(Black-box testing)

Teste, funcional ou não funcional, sem referência à estrutura interna do componente ou do sistema.

Teste de caminho

(Path testing)

Técnica de modelagem de teste de caixa branca na qual os casos de teste são modelados para executar caminhos.

Teste de carga

(Load testing)

Teste que mede o comportamento de um componente ou sistema por meio do aumento de carga, por exemplo, número de usuários paralelos e/ou número de transações para determinar qual o tamanho de carga que pode ser suportada pelo componente ou sistema. Ver também teste de esforço.

Teste de caso de uso

(Use case testing)

Técnica de modelagem de teste de caixa preta na qual os casos de teste são modelados para executar cenários de usuário.

Teste de cenário

(Scenario testing)

Ver teste de caso de uso.

Teste de cenário de usuário

(User scenario testing)

Ver teste de caso de uso.

Teste de ciclo de processo

(Process cycle test)

Técnica de modelagem de teste de caixa preta na qual os casos de teste são modelados para executar procedimentos e processos comerciais. [TMap]

Teste de cobertura lógica

(Logic-coverage testing)

Ver teste de caixa branca. [Myers]

Teste de combinação de condição de desvio

(Branch condition combination testing)

Ver teste de condições múltiplas.

Teste de combinação de condições

(Condition combination testing)

Ver teste de condições múltiplas.

Teste de comparação

(Back-to-back testing)

Teste em que duas ou mais variantes de um componente ou sistema são executados com as mesmas entradas/inputs, os resultados/saídas/outputs são comparados e analisados nos casos em que houver discrepância. [IEE 610].

Teste de comparação elementar

(Elementary comparison testing)

Técnicas de modelagem de teste de caixa preta nas quais os casos de testes são modelados para executar combinações de entradas/inputs utilizando o conceito de cobertura de determinação de condição. [TMap]

Teste de compatibilidade

(Compatibility testing)

Ver teste de interoperabilidade.

Teste de compatibilidade

(Compliance testing)

Processo pelo qual testes são realizados para determinar o atendimento de um componente ou sistema.

Teste de componente

(Component testing)

Teste realizado com os componentes individuais de um software. [Subsequente ao IEEE 610]

Teste de concorrência

(Concurrency testing)

Teste realizado para determinar como a ocorrência de duas ou mais atividades dentro de um mesmo intervalo de tempo, alcançada por entrelaçamento das atividades ou por execução simultânea, é tratada pelo componente ou sistema. [Subsequente a IEEE 610]

Teste de condição

(Condition testing)

Técnica de modelagem de teste de caixa branca na qual os casos de teste são modelados para executar os resultados das condições.

Teste de condição de decisão

(Decision condition testing)

Técnica de modelagem de teste de caixa branca na qual os casos de teste são modelados para executar resultados de condição e resultados de decisão.

Teste de condição múltipla

(Multiple condition testing)

Técnica de modelagem de teste de caixa branca na qual os casos de testes são modelados para executar combinações de resultados de condição única (dentro de uma sentença).

Teste de confiabilidade

(Reliability testing)

Processo que determina a confiabilidade de um produto de software.

Teste de confiança

(Confidence test)

Ver teste de fumaça.

Teste de configuração

(Configuration testing)

Ver teste de portabilidade.

Teste de confirmação

(Confirmation test)

Ver re-teste.

Teste de conformidade

(Conformance testing)

Ver teste de atendimento.

Teste de conversão

(Conversion testing)

Teste de software utilizado para converter dados de sistemas existentes para uso em sistemas de substituição.

Teste de decisão

(Decision testing)

Técnica de modelagem de teste de caixa branca na qual os casos de teste são modelados para executar os resultados de decisão.

Teste de decisão de condição modificada

(Modified condition decision testing)

Ver teste de cobertura de determinação de condição.

Teste de desempenho

(Performance testing)

Processo que determina o desempenho de um produto de software. Ver também teste de eficiência.

Teste de desenvolvimento

(Development testing)

Teste formal ou informal conduzido durante a implementação de um componente ou sistema, normalmente realizado no ambiente de desenvolvimento dos desenvolvedores. [Subsequente a IEEE 610]

Teste de desvio

(Branch testing)

Técnica de modelagem de teste de caixa branca na qual os casos de teste são modelados para executarem desvios.

Teste de determinação de condição

(Condition determination testing)

Técnica de modelagem de teste de caixa branca na qual os casos de testes são modelados para executar resultados de condições únicas que afetam de modo independente um resultado de decisão.

Teste de documentação

(Documentation testing)

Testa a qualidade da documentação, por exemplo, guia do usuário ou guia de instalação.

Teste de entrada

(Intake test)

Instância especial do teste de fumaça que decide se o componente ou sistema está pronto para testes mais detalhados. Este teste normalmente é realizado no início da fase de execução de teste. Ver também teste de fumaça.

Teste de escalabilidade

(Scalability testing)

Teste que determina a escalabilidade de um produto de software.

Teste de estado finito

(Finite state testing)

Ver teste de transição de estado.

Teste de estresse

(Stress testing)

Avalia um sistema ou componente em relação e além dos limites de seus requisitos especificados. [IEEE 610]. Ver também teste de carga.

Teste de fluxo de dados

(Data flow testing)

Técnica de modelagem de teste de caixa branca na qual, casos de teste são modelados para executar definições e para utilizar pares de variáveis.

Teste de fumaça

(Smoke test)

Subconjunto de todos os casos de testes definidos/planejados que cobre a principal funcionalidade de um componente ou sistema, para averiguar as principais funções de um programa em funcionamento sem se preocupar com maiores detalhes. A realização diária de testes de construção e fumaça está entre as melhores práticas do ramo. Ver também teste intake.

Teste de funcionalidade

(Functionality testing)

Realizado para determinar a funcionalidade de um produto de software.

Teste de instalabilidade

(Instalability testing)

Processo que testa a “instalabilidade” de um produto de software. Ver também teste de portabilidade.

Teste de integração

(Integration testing)

Teste realizado com a finalidade de expor defeitos nas interfaces e nas interações entre componentes ou sistemas integrados. Ver também teste de integração de componente e teste de integração de sistema.

Teste de integração de componente

(Component integration testing)

Teste realizado para expor defeitos nas interfaces e interações entre componentes integrados.

Teste de integração de sistema

(System integration testing)

Testa a integração de sistemas e pacotes e as interfaces com organizações externas, por exemplo, troca de dados eletrônicos pela Internet.

Teste de integração em larga escala

(Integration testing in the large)

Ver teste de integração de sistema.

Teste de integração em pequena escala

(Integration system in the small)

Ver teste de integração de componente.

Teste de integridade de banco de dados

(Database integrity testing)

Testa os métodos e processos utilizados para acessar e gerenciar o banco de dados a fim de assegurar que os métodos de acesso, os processos e as regras dos dados funcionem conforme esperado e que, durante o acesso ao banco de dados, estes não sejam corrompidos, removidos inesperadamente, atualizados ou criados.

Teste de integridade de dados

(Data integrity testing)

Ver teste de integridade de banco de dados.

Teste de interface

(Interface testing)

Tipo de teste de integração que testa interfaces entre componentes ou sistemas.

Teste de interoperabilidade

(Interoperability testing)

Nome dado ao processo que determina a interoperabilidade de um produto de software. Ver também teste de funcionalidade.

Teste de isolamento

(Isolation testing)

Testa componentes individuais, isolando-os dos componentes do meio. Se houver necessidade, podem-se simular os componentes do meio com simuladores (stubs) e controladores (drivers).

Teste de link

(Link testing)

Ver teste de integração de componente.

Teste de manutenibilidade

(Maintainability testing)

Processo que determina a manutenibilidade de um produto de software.

Teste de manutenção

(Maintenance testing)

Testa as alterações feitas em um sistema operacional ou o impacto de um ambiente alterado em um sistema operacional.

Teste de Mesa

(Desk checking)

Teste de software ou de especificação por meio da simulação manual de sua execução. Ver também análise estática.

Teste de migração

(Migration testing)

Ver teste de conversão.

Teste de módulo

(Module testing)

Ver teste de componente.

Teste de mutação

(Mutation testing)

Ver teste de comparação.

Teste de padrões

(Standards testing)

Ver teste de atendimento.

Teste de partição

(Partition testing)

Ver particionamento de equivalência [Beizer]

Teste de perfil operacional

(Operational profile testing)

Teste estatístico que utiliza um modelo de operações de sistema (para tarefas de curta duração) e da probabilidade de uso mais comum. [Musa]

Teste de portabilidade

(Portability testing)

Processo que determina a portabilidade de um produto de software.

Teste de programa

(Program testing)

Ver teste de componente.

Teste de recuperabilidade

(Recoverability testing)

Processo que determina a recuperabilidade de um produto de software. Ver também teste de confiabilidade.

Teste de recuperação

(Recovery testing)

Ver teste de recuperabilidade.

Teste de referência

(Benchmark test)

(1) Padrão de referência para se fazer medições e comparações. (2) Teste utilizado para comparar componentes ou sistemas entre si ou com relação a um padrão, tal como em (1). [Subsequente a IEEE 610].

Teste de regressão

(Regression testing)

Teste realizado em um programa previamente testado após alguma modificação feita e com a finalidade de assegurar que defeitos não tenham sido introduzidos ou mascarados nas áreas não alteradas do software como resultado da referida modificação. Este teste é realizado quando o software ou seu ambiente for alterado.

Teste de regulamentação

(Regulation testing)

Ver teste de atendimento.

Teste de robustez

(Robustness testing)

Teste que determina a robustez de um produto de software.

Teste de sanidade

(Sanity test)

Ver teste de fumaça.

Teste de segurança

(Safety testing)

Teste que determina a segurança de um produto de software.

Teste de segurança de acesso

(Security testing)

Teste que determina a segurança de um produto de software. Ver também teste de funcionalidade.

Teste de sentença (comando)

(Statement testing)

Técnica de modelagem de teste de caixa branca na qual os casos de teste são modelados para executar sentenças.

Teste de Seqüência de Código Linear e Salto

(LCSAJ testing)

Técnica de modelagem de teste de caixa branca na qual os casos de teste são modelados para executar LCSAJs.

Teste de servicibilidade

(Serviceability testing)

Ver teste de manutenibilidade.

Teste de sintaxe

(Syntax testing)

Técnica de modelagem de teste de caixa preta na qual os casos de teste são modelados com base nas definições do domínio de entrada/input e/ou no domínio de saída/output.

Teste de sistema

(System testing)

Testa um sistema integrado para verificar se ele atende aos requisitos especificados. [Hetzel]

Teste de tabela de decisão

(Decision table testing)

Técnicas de modelagem de teste de caixa preta nas quais os casos de teste são modelados para executar combinações de entradas/inputs e/ou estímulos (causas) apresentados em uma tabela de decisão. [Veenendaal]

Teste de transição de estados

(State transition testing)

Técnica de modelagem de caixa preta na qual os casos de testes são modelados para executar transições de estados válidas e inválidas. Ver também teste N-switch.

Teste de unidade

(Unit testing)

Ver teste de componente.

Teste de usabilidade

(Usability testing)

Teste que determina a extensão até a qual o produto de software é entendido, fácil de aprender, fácil de operar e atraente para os usuários sob condições específicas. [Subsequente a ISO 9126]

Teste de usuário

(User test)

Teste no qual os usuários da vida real se envolvem na avaliação da usabilidade de um componente ou sistema.

Teste de utilização de recurso

(Resource utilization testing)

Processo que determina a utilização de recursos por um dado produto de software. Ver também teste de eficiência.

Teste de valor limite

(Boundary value testing)

Ver análise de valor limite.

Teste de volume

(Volume testing)

Teste que submete o sistema a grandes volumes de dados. Ver também teste de recurso e utilização.

Teste dinâmico

(Dynamic testing)

Trata-se de um teste que envolve a execução do software de um componente ou sistema.

Teste dirigido a dados

(Data driven testing)

Técnica de roteiro que armazena entradas de testes e resultados esperados em uma tabela ou planilha para que um único roteiro de controle possa executar todos os testes na tabela. O teste dirigido para dados é frequentemente utilizado para dar suporte à aplicação de ferramentas de execução de teste tais como ferramentas de captura/playback. [Fewster e Graham] Ver também teste dirigido a palavra chave.

Teste dirigido a palavra-chave

(Keyword driven testing)

Técnica de desenvolvimento de scripts que utiliza arquivos de dados para conter não apenas os dados dos testes e seus resultados esperados, mas também palavras-chave relacionadas ao aplicativo sendo testado. As palavras-chave são interpretadas por scripts de apoio especiais chamados pelo script controlador para o teste. Ver também teste dirigido aos dados.

Teste dirigido para lógica

(Logic-driven testing)

Ver teste de caixa branca.

Teste em campo

(Field testing)

Ver beta teste.

Teste em pares

(Pair testing)

Duas pessoas, por exemplo, dois testadores, um desenvolvedor e um testador, ou um usuário final e um testador, trabalham juntos para descobrir defeitos. De modo geral, eles compartilham o mesmo computador e alternam-se no controle durante o teste.

Teste em threads

(Thread testing)

Versão do teste de integração de componente na qual a integração progressiva de componentes segue a implementação de subconjuntos de requisitos, ao contrário da integração de componentes por níveis de hierarquia.

Teste estático

(Static testing)

Teste de um componente ou sistema em nível de especificação ou implementação sem a execução daquele software, ex. revisões ou análise de códigos estáticos.

Teste estatístico

(Statistical testing)

Técnica de modelagem de teste na qual um modelo da distribuição estatística da entrada/input é utilizado para construir casos de testes representativos. Ver também teste de desempenho operacional.

Teste estrutural

(Structural testing)

Ver teste de caixa branca.

Teste exaustivo

(Exhaustive testing)

Abordagem na qual a suite de teste abarca todas as combinações de valores e condições de entrada/input.

Teste exploratório

(Exploratory testing)

Técnica de modelagem de teste informal na qual o testador controla ativamente a modelagem dos testes já que estes são realizados e utilizam as informações obtidas durante o teste para modelar testes novos e melhores. [Subsequente a Bach]

Teste funcional

(Functional testing)

Teste baseado em uma análise da especificação de funcionalidade de um componente ou sistema. Ver também teste de caixa preta.

Teste incremental

(Incremental testing)

Teste no qual os componentes ou sistemas são integrados e testados sozinhos ou em pequenos grupos por vez, até que todos os componentes ou sistemas sejam integrados e testados.

Teste inválido

(Invalid testing)

Utiliza valores de entrada/input que devem ser rejeitados pelo componente ou sistema. Ver também tolerância ao erro.

Teste não funcional

(Non-functional testing)

Testa os atributos de um componente ou sistema que não se relacionam à funcionalidade, por exemplo, confiabilidade, eficiência, usabilidade, manutenibilidade e portabilidade.

Teste sujo

(Dirty testing)

Ver teste de negativo.

Teste negativo

(Negative testing)

Visa mostrar que um componente ou sistema não funciona. O teste negativo se refere mais à atitude do testador do que a uma abordagem de teste específica ou uma técnica de modelagem de teste, por exemplo, o teste com valores de entrada/input inválidos ou com exceções. [Subseqüente a Beizer].

Teste N-switch

(N-switch testing)

Forma de teste de transição de estado na qual os casos de teste são modelados para executar todas as seqüências de N+1 válidas. [Chow]. Ver também teste de transição de estado.

Teste operacional

(Operational testing)

Realizado com a finalidade de avaliar um componente ou sistema em seu ambiente operacional. [IEEE 610]

Teste top-down

(Top-down testing)

Abordagem incremental do teste de integração na qual o componente na parte superior da hierarquia de componentes é testado primeiro, e os componentes nos níveis mais baixos são simulados por simuladores (stubs). Os componentes testados são então utilizados para testar componentes de níveis mais baixos. Este processo é repetido até que os componentes de níveis mais baixos tenham sido testados. Ver também teste de integração.

Testware

(Testware)

Artefatos produzidos durante o processo de teste e requeridos para planejar, projetar e executar testes, entre eles documentação, roteiros, entradas/inputs, resultados esperados, procedimentos de preparação e de esclarecimento, arquivos, bancos de dados, ambiente e qualquer software adicional ou utilitários utilizados no teste. [Subseqüente a Fewster e Graham]

Testware automatizado

(Automated testware)

Testware utilizado em teste automatizado, tal como scripts de ferramenta.

Tipo de teste

(Test type)

Grupo de atividades de teste que testa um componente ou sistema enfocando um objetivo de teste específico, ou seja, funcional, usabilidade, regressão, etc. Um tipo de teste pode acontecer em um ou mais níveis ou fases de teste. [Subseqüente a TMap]

Tolerância à falta

(Fault tolerance)

Capacidade que um produto de software tem para manter um nível específico de desempenho em casos de faltas (defeitos) de software ou de infração de sua interface específica. [ISO 9126]. Ver também confiabilidade.

Tolerância ao erro

(Error tolerance)

Habilidade que um sistema ou componente tem para continuar a operar normalmente a despeito da presença de entradas/inputs errôneas. [Subseqüente a IEEE 610]

Transição de estados

(State transition)

Transição entre dois estados de um componente ou sistema.

Tratamento de exceção

(Exception handling)

Comportamento apresentado por um componente, sistema ao responder a entradas/inputs erradas, sejam de um usuário humano, de outro componente ou sistema, ou a uma falha interna.

Trilha de auditoria

(Audit trail)

Caminho inverso pelo qual a entrada (input) original de um processo (ex. dados) pode ser rastreada, tomando como ponto de partida as saídas (outputs). Isto facilita a análise de defeitos e permite que se realize uma auditoria do processo. [Subsequente ao TMap].

U**Unidade**

(Unit)

Ver componente.

Usabilidade

(Usability)

Capacidade que um software tem de ser entendido, aprendido, utilizado e atraente para o usuário quando utilizado sob determinadas condições. [ISO 9126]

Utilização de recursos

(Resource utilization)

Capacidade que um produto de software tem de utilizar quantidades e tipos de recursos apropriados, ex. quantidades de memória principal e secundária utilizadas pelo programa e tamanhos de arquivos temporários ou de estouro/overflow necessários quando o software desempenha sua função sob condições dadas. [Subsequente a ISO 9126]. Ver também eficiência.

V**Validação**

(Validation)

Confirmação, por meio de exames e do fornecimento de evidências objetivas, que os requisitos de uso ou aplicação específica tencionada foram atendidos. [ISO 9000]

Valor de entrada

(Input value)

Instância de uma entrada/input. Ver também entrada.

Valor de saída

(Output value)

Instância de uma saída/output. Ver também saída.

Valor limite

(Boundary value)

Valor de entrada/input ou de saída/output que está no limite de um particionamento de equivalência ou na menor distância incremental em quaisquer dos lados de um edge, por exemplo, o valor mínimo ou máximo de uma amplitude.

Variável

(Variable)

Elemento de armazenamento em um computador que pode ser acessado por um programa de software referindo-se a ele por um nome.

Vazamento de memória

(Memory leak)

Defeito encontrado na lógica de alocação de armazenamento dinâmico de um dado programa e que faz com que o programa falhe ao recuperar a memória após tê-la utilizado, terminando por causar falha no programa em função da falta de memória.

Verificação

(Verification)

Confirmação por meio de exame e do fornecimento de evidências objetivas que os requisitos especificados foram atendidos. [ISO 9000]

Verificador

(Checker)

Ver revisor.

W

Wide Band Delphi

(Wide Band Delphi)

Técnica de estimação de teste baseada em especialista que visa produzir uma estimação precisa utilizando a sabedoria coletiva dos membros da equipe.