**SOFTWARE TESTING**

**Adaptabilidad** (Adaptability) Es la capacidad del producto software de ser adaptado a diferentes entornos sin la aplicación de acciones o medios distintos de los aportados para este propósito por el software considerado.

**Análisis de Árbol de Faltas** (Fault Tree Analysis) Método utilizado para analizar las causas de faltas (defectos).

**Análisis de Impacto** (Impact Analysis) Valoración del cambio en las capas de documentación de desarrollo, documentación de pruebas y componentes, con el objeto de implementar un cambio dado en requisitos especificados.

**Análisis de Punto de Prueba APP** (Test Point Analysis TPA) Fórmula basada en un método de estimación de pruebas basada en el análisis de punto de función.

**Analisis de Riesgos** (Risk Analysis) Proceso de evaluación de los riesgos identificados para estimar su probabilidad e impacto.

**Análisis de Valores Límite** (Boundary Value Analysis) Técnica de diseño de pruebas de caja negra en la cual los casos de prueba son diseñados basándose en los valores límite. Véase también valor límite.

**Análisis del Flujo de Datos** (Data Flow Analysis) Forma de análisis estático basado en la definición y uso de variables.

**Análisis del Modo de Fallo y Efecto AMFE** (Failure Mode and Effect Analysis FMEA) Método sistemático para la identificación y análisis de riesgos identificando posibles modos de fallo e intentando impedir su ocurrencia.

**Análisis Dinámico** (Dynamic Analysis) Proceso de evaluación del comportamiento, por ejemplo rendimiento de memoria, uso de CPU, de un sistema o componente durante su ejecución.

**Analisis Estático** (Static Analysis) Análisis de los artefactos software, por ejemplo requisitos o código, llevado a cabo sin la ejecución de estos artefactos software.

**Análisis Estático de Código** (Static Code Analysis) Análisis del código fuente llevado a cabo sin la ejecución de ese software.

**Analizabilidad** (Analyzability) Capacidad de un producto software de ser diagnosticado por deficiencias o causas de fallos en el software, o para las partes a ser modificadas o identificadas.

**Analizador Estático** (Static Analyzer) Herramienta que lleva a cabo el análisis estático.

**Anomalía** (Anomaly) Cualquier condición que se desvíe de las expectativas basadas en las especificaciones de requisitos, documentos de diseño, documentos de usuario, estándares, etc., o de la percepción o experiencia de alguien. Las anomalías pueden ser encontradas durante, aunque no se limitan sólo a, revisiones, proceso de pruebas, análisis, compilación, o uso de productos de software o documentación aplicable.

**Aprendibilidad** (Learnability)Capacidad del producto software de permitir al usuario aprender su forma de uso. Véase también usabilidad.

**Arnés de Pruebas** (Test Harness) Entorno de pruebas constituido por stubs y controladores necesarios para llevar a cabo una prueba.

**Aseguramiento de la Calidad** (Quality Assurance) Parte de la gestión de calidad orientada a proporcionar confianza en que los requisitos serán cumplidos.

**Asistente de Instalación** (Installation Wizard) Software suministrado en cualquier medio adecuado que guía al instalador a través del proceso de instalación. Normalmente el asistente de instalación ejecuta el proceso de instalación, aporta información con respecto a los resultados de la instalación y presenta indicadores para opciones.

**Auditoría**(Audit) Una evaluación independiente del producto o proceso software para constatar el cumplimiento de estándares, guías, especificaciones y/o procedimientos, basada en criterios objetivos, incluyendo los documentos que especifican: (1) El formato o contenido de los productos a ser producidos (2) El proceso por el que el producto será producido (3) Cómo será medido el cumplimiento con estándares o guías.

**Automatización de Pruebas** (Test Automation) Uso de software para realizar o apoyar las actividades de pruebas, por ejemplo gestión de pruebas, diseño de pruebas, ejecución de pruebas y comprobación de resultados.

**Base de Prueba** (Test Basis) Todos los documentos de donde los requisitos de un componente o sistema pueden ser inferidos. La documentación en la que se basan los casos de prueba. Si un documento puede ser modificado sólo por medio de un procedimiento de cambio formal, entonces la base de las pruebas se denomina base de prueba congelada.

**Base de Prueba Congelada** (Frozen Test Basis) Documento base de prueba que solamente puede ser modificado por un proceso formal de control del cambio.

**Calendario de Ejecución de Pruebas** (Test Execution Schedule) Planificación para la ejecución de procedimientos de prueba. Los procedimientos de prueba están incluidos en el calendario de ejecución de pruebas en su contexto y en el orden en el cual deben ser ejecutados.

**Calidad** (Quality) Grado en el cual un componente, sistema o proceso satisface requisitos especificados y/o necesidades y expectativas del usuario/cliente.

**Camino** (Path) Secuencia de eventos, por ejemplo sentencias ejecutables, de un componente o sistema desde un punto de entrada hasta un punto de salida.

**Camino Inviable** (Infeasible Path) Camino que no puede ser practicado por ningún posible conjunto de valores de entrada.

**Capability Maturity Model Integration CMMI**Marco de trabajo que describe los elementos clave para un proceso de desarrollo y mantenimiento efectivo de un producto. El modelo CMMI cubre las mejores prácticas para la planificación, la ingeniería y gestión del desarrollo y el mantenimiento del producto. CMMI es el sucesor designado de CMM.

**Característica de Calidad / Atributo de calidad**(Quality characteristic / Quality attribute) Prestación o característica que afecta a la calidad de un elemento.

**Caso de Prueba** (Test Case) Conjunto de valores de entrada, pre condiciones de ejecución, resultados esperados y post condiciones de ejecución, desarrollado con un objetivo en particular o condición de prueba, tales como probar un determinado camino de ejecución o para verificar el cumplimiento de un requisito determinado.

**Caso de Prueba Bloqueado** (Blocked Test Case) Caso de prueba que no puede ser ejecutado porque las pre condiciones para su ejecución no son satisfechas.

**Caso de Prueba de Alto Nivel / Caso de Prueba Lógico** (High Level Test Case / Logical Test Case) Caso de prueba sin valores concretos (nivel de implementación) para datos de entrada y resultados esperados. Se utilizan operadores lógicos, las instancias de los valores reales no se han definido aún y/o no están disponibles.

**Caso de Prueba de Bajo Nivel / Caso de Prueba Concreto** (Low Level Test Case / Concrete Test Case) Caso de prueba con valores concretos (nivel de implementación) para datos de entrada y resultados esperados. Los operadores lógicos provenientes de los casos de prueba de alto nivel son reemplazados por valores reales que corresponden con los objetivos de los operadores lógicos.

**Causa Raíz** (Root Cause) Factor subyacente que ha causado una no conformidad y, posiblemente, debería ser eliminado de forma permanente mediante una mejora del proceso.

**Certificación** (Certification) Proceso de confirmación de que un componente, sistema o persona cumple con sus requisitos especificados, ejemplo al aprobar un exámen.

**Ciclo de Pruebas** (Test Cycle) Ejecución del proceso de pruebas contra una entrega única e identificable del objeto de prueba.

**Cierre de Pruebas** (Test Closure) Durante la fase de cierre de pruebas de un proceso de pruebas se recolectan los datos de aquellas actividades finalizadas con el objetivo de consolidar la experiencia, productos de soporte de pruebas, hechos y números. La fase de cierre de pruebas consiste en finalizar y archivar los productos de soporte de pruebas y evaluar el proceso de pruebas, incluyendo la preparación de un informe de evaluación de pruebas.

**Cobertura / Cobertura de Pruebas** (Coverage / Test Coverage) Grado, expresado como un porcentaje, en el que un elemento de cobertura especificado ha sido practicado por un juego de pruebas.

**Cobertura de Camino** (Path Coverage) Porcentaje de caminos que han sido practicados por un conjunto de pruebas. El 100% de la cobertura de caminos implica el 100% de la cobertura SLYSC.

**Cobertura de Código** (Code Coverage) Método de análisis que determina qué partes del software han sido ejecutadas (cubiertas) por el juego de pruebas y qué partes no han sido ejecutadas, ejemplo cobertura de sentencia, cobertura de decisión o cobertura de condición.

**Cobertura de Condición** (Condition Coverage) Porcentaje de los posibles resultados de una condición que han sido practicados por un juego de pruebas. Una cobertura de condición del 100% requiere que cada condición simple de toda sentencia de decisión haya sido probada como Verdadera y Falsa.

**Cobertura de Condición Múltiple** (Multiple Condition Coverage) Porcentaje de combinaciones de todos los resultados de condiciones simples dentro de una sentencia, que han sido practicadas por un juego de pruebas. El 100% de cobertura de condición múltiple implica el 100% de cobertura determinación de condición.

**Cobertura de Conmutador de Multiplicidad N** (N-Switch Coverage) Porcentaje de secuencias de N+1 transiciones que han sido practicadas por un juego de pruebas.

**Cobertura de Decisión** (Decision Coverage) Porcentaje de resultados de decisión que han sido practicados por un juego de pruebas. El 100% de cobertura de decisión implica tanto un 100% de cobertura de rama como un 100% de cobertura de sentencia.

**Cobertura de Decisión de Condición Modificada / Cobertura de Determinación de Condición** (Modified Condition Decision Coverage / Condition Determination Coverage) Porcentaje de todos los resultados de las condiciones unitarias que afectan de forma independiente al resultado de una decisión que han sido practicados por un juego de casos de prueba. Una cobertura de condición de determinación del 100% implica un 100% de cobertura de condición en la decisión.

**Cobertura de Partición de Equivalencia** (Equivalence Partition Coverage) Porcentaje de particiones de equivalencia que han sido practicadas por un juego de pruebas.

**Cobertura de Rama** (Branch Coverage) Porcentaje de ramas que han sido practicadas por un juego de pruebas. 100% de cobertura de rama implica 100% de cobertura de decisión y 100% de cobertura de sentencia.

**Cobertura de Sentencia** (Statement Coverage) Porcentaje de sentencias ejecutables que han sido practicadas por un juego de pruebas.

**Cobertura de Valores Límite** (Boundary Value Coverage) Porcentaje de valores límite que han sido practicados por un juego de pruebas.

**Cobertura Estructural** (Structural Coverage) Medidas de cobertura basadas en la estructura interna de un componente o sistema.

**Coexistencia** (Co-existence) Capacidad del producto software de coexistir con otro software independiente en un entorno común compartiendo recursos comunes.

**Comparación de Pruebas** (Test Comparison) Proceso de identificación de las diferencias entre los resultados reales producidos por el componente o sistema en pruebas y los resultados esperados para la prueba. La comparación de pruebas puede ser llevada a cabo durante la ejecución de la prueba (comparación dinámica) o con posterioridad a la ejecución de la prueba.

**Comparador de Pruebas** (Test Comparator) Herramienta de pruebas para realizar comparaciones automáticas de pruebas entre los resultados obtenidos y los resultados esperados.

**Compilador** (Compiler) Herramienta software que traduce programas expresados en un lenguaje de alto nivel a su lenguaje máquina equivalente.

**Complejidad** (Complexity) Grado en el cual un componente o sistema tiene un diseño y/o estructura interna que es difícil de comprender, mantener y verificar. Véase también complejidad ciclomática.

**Complejidad Ciclomática** (Cyclomatic Complexity) Número de caminos independientes en un programa. Se define la complejidad ciclomática como: L - N + 2P, donde - L = número de aristas/enlaces en un grafo - N = número de nodos en un grafo - P = número de partes desconectadas del grafo (por ejemplo un grafo invocado y una sub rutina)

**Comportamiento**(Behavior) Respuesta de un componente o sistema a un conjunto de valores de entrada y de pre condiciones.

**Comportamiento Temporal / Rendimiento** (Time behavior / Performance) Grado en el cual un sistema o componente logra la función señalada dentro de las restricciones dadas con respecto al tiempo de proceso y tasa de transferencia.

**Condición** (Condition) Expresión lógica que puede ser evaluada como Verdadera o Falsa. Por ejemplo A>B.

**Condición Compuesta** (Compound Condition) Dos o más condiciones simples unidas mediante un operador lógico (AND, OR o XOR). Por ejemplo ´A>B AND C>100´.

**Condición de Prueba / Situación de Prueba** (Test Condition / Test Situation) Elemento o evento de un componente o sistema que debería ser verificado por uno o más casos de prueba, por ejemplo una función, transacción, característica, atributo de calidad o elemento estructural.

**Conjunto de Pruebas Base** (Basis Test Set) Conjunto de casos de prueba derivados de la estructura interna de un componente o especificación para asegurar que será alcanzado el 100% de un criterio de cobertura especificado.

**Construcción Diaria** (Daily Build) Actividad diaria de desarrollo en la que un sistema completo se compila y enlaza (normalmente durante la noche), de modo que un sistema consistente esté disponible en cualquier momento incluyendo la totalidad de los últimos cambios.

**Contrato de Pruebas** (Test Charter) Declaración de los objetivos de pruebas y posibles ideas de cómo probar. El contrato de pruebas es usado por ejemplo en pruebas exploratorias.

**Control de Pruebas / Requisito de Prueba** (Test Control / Test Requirement) Tarea de la gestión de pruebas que se encarga de desarrollar y aplicar un conjunto de acciones correctivas para poner el proyecto de pruebas en la dirección correcta cuando el seguimiento (monitorización) muestra una desviación con respecto a lo que se había planificado.

**Controlador / Controlador de Prueba** (Driver / Test Driver) Componente software o herramienta de pruebas que sustituye a un componente que asume el control y/o la invocación a un componente o sistema.

**Criterios de Aceptación** (Acceptance Criteria) Los criterios de salida que un componente o sistema debe satisfacer para ser aceptado por un usuario, cliente u otra entidad autorizada.

**Criterios de Entrada** (Entry Criteria) Conjunto de condiciones genéricas y específicas para permitir que un proceso prosiga con una tarea definida, por ejemplo la fase de pruebas. El objetivo de los criterios de entrada es evitar que una tarea comience, lo cual conllevaría un mayor esfuerzo que el necesario para eliminar los criterios de entrada fallidos.

**Criterios de Paso / Fallo** (Pass / Fail Criteria) Reglas de decisión utilizadas para determinar si un elemento de prueba (función) o prestación ha pasado o fallado una prueba.

**Criterios de Salida / Criterios de Compleción de Pruebas / Meta de Prueba** (Exit Criteria / Test completion criteria / Test Target) Conjunto de condiciones genéricas y específicas, acordadas con los involucrados, para que un proceso sea considerado formalmente concluido. El propósito de los criterios de salida es evitar que una tarea se considere concluida habiendo partes destacadas de la misma sin completar. Los criterios de salida son utilizados como referencia para la elaboración de informes y planificación de la suspensión de las pruebas.

**Cumplimiento** (Compliance) Capacidad del producto software para adherirse a estándares, convenciones, normativa legal y prescripciones similares.

**Datos de Prueba** (Test Data) Datos que existen (por ejemplo en una base de datos) antes de que una prueba sea ejecutada y que afectan o son afectados por el componente o sistema en pruebas.

**Decisión** (Decision) Punto de un programa en el cual el flujo de control tiene dos o más rutas alternativas. Un nodo con dos o más enlaces para separar ramas.

**Defecto** (Defect) Desperfecto en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas, por ejemplo una sentencia o una definición de datos incorrectas. Si se localiza un defecto durante una ejecución puede causar un fallo en el componente o sistema.

**Definición de Datos** (Data Definition) Sentencia ejecutable donde se asigna un valor a una variable.

**Densidad de Defectos** (Defect Density) Número de defectos identificados en un componente o sistema dividido por el tamaño del mismo (expresado en términos de medidas estándar, por ejemplo líneas de código, número de clases o puntos de función).

**Depuración** (Debugging) Proceso de encontrar, analizar y eliminar las causas de los fallos en el software.

**Desarrollo Guiado por Pruebas** (Test-Driven Development (TDD) Modalidad de desarrollo software donde los casos de prueba son desarrollados, y a menudo automatizados, antes de que el software sea desarrollado para ejecutar esos casos de prueba.

**Disponibilidad** (availability) Grado hasta cuál un componente o sistema es operativo y accesible cuando se requiere su uso. A menudo es expresado como un porcentaje.

**Efecto Sonda** (Probe Effect) Efecto producido por el instrumento de medida sobre el sistema o componente que está siendo medido, por ejemplo mediante una herramienta de pruebas de rendimiento o un monitor. Por ejemplo el rendimiento puede ser ligeramente peor cuando las herramientas de prueba de rendimiento están siendo usadas.

**Eficiencia** (Efficiency) Capacidad del producto software para proporcionar un rendimiento apropiado, relativo a la cantidad de recursos usados bajo condiciones establecidas.

**Ejecución de Automatización de Pruebas** (Test Execution Automation) Uso de herramientas software, por ejemplo herramientas de captura/reproducción, para controlar la ejecución de pruebas, la comparación de resultados reales y esperados, el establecimiento de pre condiciones de pruebas, y otras funciones de control de pruebas y generación de informes.

**Ejecución de Prueba** (Test Execution) Proceso de practicar una prueba sobre el componente o sistema en pruebas, produciendo resultado(s) reales.

**Ejecución de una Prueba** (Test Run) Ejecución de una prueba sobre una versión específica de un objeto de prueba.

**Elemento  de Cobertura** (Coverage Item) Entidad o propiedad utilizada como base para la cobertura de las pruebas, por ejemplo particiones de equivalencia o sentencias de código.

**Emulador** (Emulator) Dispositivo, programa de ordenador o sistema que acepta las mismas entradas y produce las mismas salidas que un sistema dado.

**Enfoque de Pruebas** (Test Approach) Implementación de la estrategia de pruebas definida para un proyecto específico. En general ésta incluye las decisiones tomadas en función de los objetivos del proyecto (desde el punto de vista del proceso de pruebas) y la evaluación de riesgo llevada a cabo, puntos de entrada respecto del proceso de pruebas, las técnicas de diseño de pruebas a aplicar, criterios de salida y tipos de pruebas a ejecutar.

**Entorno de Operaciones** (Operational Environment) Productos hardware y software instalados en dependencias de los usuarios o clientes donde el componente o sistema en pruebas será utilizado. El software puede incluir sistemas operativos, sistemas de gestión de base de datos y otras aplicaciones.

**Entorno de Pruebas** (Test Environment) Entorno que contiene hardware, instrumentación, simuladores, herramientas software y otros elementos de soporte necesarios para realizar una prueba.

**Entorno de Pruebas / Cama de Pruebas** (Test Environment / Test Bed) Entorno que contiene hardware, instrumentación, simuladores, herramientas software y otros elementos de soporte necesarios para realizar una prueba.

**Entrada** (Input) Variable (almacenada tanto dentro como fuera de un componente) que es leída por un componente.

**Entrada de Prueba** (Test Input) Datos recibidos desde una fuente externa por el objeto de prueba durante la ejecución de pruebas. La fuente externa puede ser hardware, software o humana.

**Error** (Error) Acción humana que produce un resultado incorrecto.

**Escalabilidad** (Scalability) Capacidad de un producto software de ser actualizado para adaptarse a cargas crecientes.

**Escriba** (Scribe) Persona que registra en un acta cada defecto mencionado y cualquier sugerencia para la mejora de un proceso durante una reunión de revisión. El escriba tiene que asegurarse que el acta sea legible y comprensible.

**Especificación de Casos de Prueba** (Test Case Specification) Documento que especifica un conjunto de casos de prueba (objetivos, entradas, acciones de prueba, resultados esperados y pre condiciones de ejecución) para un elemento de prueba.

**Especificación de Diseño de Prueba** / **Diseño de Prueba** (Test Design Specification / Test Design) Documento que especifica las condiciones de prueba (elementos de cobertura) para el elemento de prueba, el enfoque de pruebas de forma detallada e identifica los casos de prueba de alto nivel asociados.

**Especificación de Procedimiento de Prueba / Escenario de Prueba** (Test Procedure Specification / Test Scenario) Documento que especifica la secuencia de acciones para la ejecución de una prueba. También conocido como script de prueba o script de prueba manual.

**Especificación de Prueba** (Test Specification) Documento que consta de una especificación de diseño de prueba, especificación de caso de prueba y/o especificación de procedimiento de prueba.

**Estrategia de Prueba** (Test Strategy) Descripción a alto nivel de los niveles de prueba a llevar a cabo y las pruebas asociadas a ellos para una organización o programa (uno o más proyectos).

**Evaluación Heurística** (Heuristic Evaluation) Técnica de prueba estática de usabilidad para determinar el cumplimiento de la interfaz de usuario con los principios de usabilidad reconocidos (la también llamada ""heurística"").

**Fallo / Fallo de Prueba** (Failure / Test Fail)  Desviación de un componente o sistema respecto de la prestación, servicio o resultado esperados.

**Fase de Ejecución de Pruebas** (Test Execution Phase) Periodo de tiempo en el ciclo de vida de desarrollo software, durante el cual los componentes de un producto software son ejecutados y el producto software es evaluado para determinar si los requisitos han sido satisfechos.

**Fase de Prueba** (Test Phase) Conjunto concreto de actividades recogidas en una fase controlable de un proyecto, por ejemplo las actividades de ejecución de un nivel de pruebas.

**Flujo de Control** (Control Flow) Secuencia de eventos (caminos) en la ejecución a través de un componente o sistema.

**Flujo de Datos** (Data Flow) Representación abstracta de la secuencia y posibles cambios de estado de los objetos de datos, donde el estado de un objeto puede ser cualquiera de los siguientes: creación, uso o destrucción.

**Frecuencia de Fallos** (Failure Rate) Cociente del número de fallos de una categoría dada por unidad de medida especifica, por ejemplo fallos por unidad de tiempo, fallos por unidad de transacciones, fallos por número de ejecuciones de programa.

**Fuga de Memoria** (Memory Leak) Defecto en la lógica de asignación de almacenamiento dinámica de un programa que provoca que falle al reclamar memoria tras haber finalizado su uso, causando eventualmente que el programa falle debido a una insuficiencia de memoria.

**Gestión de Defectos** (Defect Management) Proceso de reconocimiento, investigación, actuación y disposición de defectos. Incluye el registro de defectos, su clasificación e identificación del impacto.

**Gestión de Incidencias** (Incident Management) Proceso de reconocimiento, investigación, toma de medidas y eliminación de incidencias. Comprende registrar incidencias, clasificarlas e identificar el impacto.

**Gestión de la Calidad** (Quality Management) Actividades coordinadas para dirigir y controlar una organización con respecto a la calidad. La dirección y el control con respecto a la calidad generalmente incluyen el establecimiento de una política de calidad y objetivos de la calidad, planificación de la calidad, control de la calidad, aseguramiento de la calidad y mejora de la calidad.

**Gestión de la Configuración** (Configuration Management) Disciplina que aplica dirección y supervisión técnica y administrativa a: identificar y documentar las características funcionales y físicas de un elemento de la configuración, controlar cambios de esas características, registrar e informar sobre el estado de la implementación y proceso de cambio y verificar la conformidad con los requisitos especificados.

**Gestión de Pruebas** (Test Management) Planificación, estimación, monitorización y control de las actividades de pruebas, típicamente llevado a cabo por un jefe de pruebas.

**Gestión de Riesgos / Control de Riesgos** (Risk management / Risk Control) Aplicación sistemática de procedimientos y prácticas a las tareas de identificar, analizar, priorizar y controlar riesgos. Control, proceso a través del cual se alcanzan decisiones y se implementan medidas de protección con el objeto de reducir riesgos a, o mantener riesgos dentro de, unos niveles determinados.

**Gráfico Causa-Efecto** (Cause-Effect Graph) Representación gráfica de entradas y/o estímulos (causas) con sus salidas asociadas (efectos), que pueden ser utilizadas para diseñar casos de prueba.

**Gráfico de Flujo de Control** (Control Flow Graph) Representación abstracta de todas las posibles secuencias de eventos (caminos) en la ejecución a través de un componente o sistema.

**Guión de Prueba** (Test Script) Comúnmente usado para referirse a una especificación de procedimiento de prueba, especialmente una automatizada.

**Herramienta de Captura / Reproducción** (Capture/Playback Tool) Tipo de herramienta de ejecución de pruebas donde las entradas son grabadas durante las pruebas manuales con el objetivo de generar scripts de pruebas automatizados que puedan ser ejecutados con posterioridad (por ejemplo reproducir). Estas herramientas son utilizadas a menudo para apoyar pruebas de regresión automatizadas.

**Herramienta de Diseño de Pruebas** (Test Design Tool) Herramienta que da soporte a la actividad de diseño de pruebas generando entradas de prueba a partir de una especificación que puede estar almacenada en el repositorio de una herramienta CASE, por ejemplo herramienta de gestión de requisitos, a partir de condiciones de prueba especificadas almacenadas en la misma herramienta, o a partir del código.

**Herramienta de Ejecución de Pruebas** (Test Execution Tool) Tipo de herramienta de pruebas que permite la ejecución de otro software empleando un script de pruebas automatizado, por ejemplo captura/reproducción.

**Herramienta de Gestión de Incidencias** (Incident Management Tool) Herramienta que facilita el registro y seguimiento del estado de incidencias. A menudo incorporan facilidades orientadas al flujo de trabajo para el seguimiento y control de la asignación, corrección y repetición de las pruebas de las incidencias y proporcionan facilidades para la elaboración de informes. Véase también herramienta de administración de defectos.

**Herramienta de Gestión de Pruebas** (Test Management Tool) Herramienta que proporciona soporte a la gestión de pruebas y control de parte del proceso de pruebas. A menudo tiene varias capacidades, tales como gestionar los productos de soporte de pruebas, planificación de pruebas, registro de resultados, seguimiento del proceso, gestión de incidencias y generación de informes de las pruebas.

**Herramienta de Monitorización** (Monitoring Tool) Herramienta software o dispositivo hardware que se ejecuta de forma concurrente con el componente o sistema en pruebas y supervisa, registra y/o analiza el comportamiento del componente o sistema.

**Herramienta de Preparación de Datos de Prueba** (Test Data Preparation Tool) Tipo de herramienta de pruebas que permite seleccionar datos de bases de datos existentes o datos creados, generados, manipulados y editados para su uso en pruebas.

**Herramienta de Pruebas**(Test Tool) Producto software que da soporte a una o más actividades de prueba, tales como planificación y control, especificación, creación de ficheros iniciales y datos, ejecución de pruebas y análisis de las mismas.Véase también CAST.

**Herramienta de Pruebas de Rendimiento** (Performance Testing Tool) Herramienta que da soporte a pruebas de rendimiento y que normalmente tiene dos prestaciones principales: generar carga y probar la medición de transacción. La generación de carga puede o bien simular múltiples usuarios o grandes volúmenes de datos de entrada. Durante la ejecución, se miden y registran tiempos de respuesta de transacciones seleccionadas. Las herramientas de pruebas de rendimiento normalmente proporcionan informes basados en los registros de las pruebas y gráficos de carga contra tiempos de respuesta.

**Herramienta de Seguridad** (Security Tool) Herramienta que soporta la seguridad operativa.

**Hito** (Milestone) Punto en el tiempo de un proyecto en el cual entregables definidos (intermedios) y resultados deberían estar disponibles.

**Identificación de Riesgos** (Risk Identification) Proceso de identificación de riesgos mediante el uso de técnicas tales como lluvia de ideas, listas de comprobación e historial de fallos.

**Incidencia / Incidencia de prueba software** (Incident / Software Test incident) Cualquier ocurrencia de un suceso que requiere investigación.

**Independencia de Pruebas** (Independence of Testing) Separación de responsabilidades, que fomenta la realización de pruebas objetivas.

**Indicador de Rendimiento** (Performance Indicator) Métrica de alto nivel de la efectividad y/o eficiencia usada para dirigir y controlar un desarrollo progresivo, por ejemplo el deslizamiento del tiempo de acción para un desarrollo software.

**Indicador de Rendimiento de Prueba** (Test Performance Indicator) Métrica de alto nivel de la efectividad y/o eficiencia usada utilizada para dirigir y controlar el desarrollo progresivo de las pruebas, por ejemplo Porcentaje de Defectos Detectados (PDD).

**Informe de Defectos** (Defect Report) Documento que informa acerca de cualquier imperfección en un componente o sistema que puede causar que el componente o sistema no realice la función requerida.

**Informe de Incidencias / Informe de Incidencias de** **Pruebas**(Incident Report / Test incident report) Documento que informa de la ocurrencia de cualquier suceso, por ejemplo durante las pruebas, que requiere investigación.

**Informe de la Evaluación de las Pruebas** (Test Evaluation Report) Documento desarrollado al final del proceso de pruebas en el que se resumen todas las actividades de pruebas y resultados. También contiene una evaluación del proceso de pruebas y lecciones aprendidas.

**Informe de Pruebas / Informe Resumen de Pruebas** (Test Report / Test summary report) Documento que resume las actividades y resultados de las pruebas. También contiene una evaluación de los correspondientes elementos de prueba respecto de los criterios de salida.

**Infraestructura de Pruebas** (Test infrastructure) Artefactos propios de la organización necesarios para llevar a cabo pruebas, consistentes en entornos de pruebas, herramientas de pruebas, entorno de administración y procedimientos.

**Inspección** (Inspection) Tipo de revisión entre pares que se basa en el examen visual de documentos para detectar defectos, por ejemplo violaciones de estándares de desarrollo y no conformidades a la documentación de nivel superior. Es la técnica de revisión más formal y, por lo tanto, siempre basada en un procedimiento documentado.

**Instalabilidad** (Installability) Capacidad del producto software para ser instalado en un entorno especificado. Véase también portabilidad.

**Integración** (Integration) Proceso de combinar componentes o sistemas en estructuras más amplias.

**Integración Funcional** (Functional Integration) Enfoque de integración que combina los componentes o los sistemas con el fin de conseguir que una funcionalidad básica esté operativa de forma temprana. Véase también pruebas de integración.

**Interoperabilidad** (Interoperability) Capacidad del producto software para interactuar con uno o más componentes o sistemas especificados. Véase también funcionalidad.

**Inventario de Medición de Usabilidad Software IMUS** (Software Usability Measurement Inventory SUMI) Cuestionario basado en técnicas de pruebas de usabilidad para evaluar la usabilidad, por ejemplo, satisfacción del usuario de un componente o sistema.

**ISTQB International Software Testing Qualifications Board**El Comité Internacional de Certificaciones de Pruebas de Software (International Software Testing Qualifications Board, ISTQB) es una organización internacional sin fines de lucro, creada en noviembre de 2002, en Edimburgo (Escocia), y registrada legalmente en Bélgica, basada en el trabajo voluntario de cientos de expertos en pruebas de software a nivel mundial. Es responsable del esquema internacional de certificación profesional denominado "ISTQB Certified Tester" (Probador de Software Certificado del ISTQB).

**Jefe de pruebas / Líder de Pruebas** (Test Manager / Test leader) Persona responsable de la gestión de proyecto de las actividades y recursos de pruebas, y de la evaluación de un objeto de prueba. Individuo que dirige, controla, administra, planifica y regula la evaluación de un objeto de prueba.

**Juego de Prueba / Conjunto de Prueba** (Test Suite / Test Set) Conjunto de casos de prueba para un componente o sistema en pruebas, donde la post-condición de una prueba es a menudo usada como pre condición de la siguiente.

**Línea Base** (Baseline) Especificación o producto software que ha sido formalmente revisada o para la cual se ha llegado a un acuerdo, que en adelante sirve como base para un desarrollo posterior y que puede ser modificado sólo a través de un proceso formal de control de cambios.

**Madurez**(Maturity) (1) Capacidad de una organización con respecto a la efectividad y la eficiencia de sus procesos y prácticas de trabajo. Véase también Capability Maturity Model, Test Maturity Model. (2) Capacidad del producto software de evitar un fallo como resultado de defectos en el software. Véase también fiabilidad.

**Mantenibilidad**(Maintainability) Facilidad con la que un producto software puede ser modificado para corregir defectos, cumplir con nuevos requisitos, hacer más sencillo el mantenimiento futuro o ser adaptado a un entorno modificado.

**Mantenimiento**(Maintenance) Modificación de un producto software tras su entrega para corregir defectos, mejorar el rendimiento u otros atributos, o adaptar el producto a un entorno modificado.

**Máquina de Estado Finito** (Finite State Machine) Modelo informático consistente en un número finito de estados y transiciones entre estos estados, posiblemente con acciones anexas.

**Marco de Trabajo de Pruebas Unitarias** (Unit Test Framework) Herramienta que aporta un entorno para pruebas unitarias o de componente en el cual un componente puede ser probado de forma aislada o con los stubs o controladores adecuados. La herramienta también aporta soporte para el desarrollador, tales como capacidades de depuración.

**Medición** (Measurement) Proceso de asignación de un número o categoría a una entidad para describir un atributo de esa entidad.

**Medida** (Measure) Número o categoría asignada a un atributo de una entidad al efectuar una medición.

**Métrica** (Metric) Escala de medida y el método utilizado para la medición.

**Modelo de Desarrollo Incremental** (Incremental Development Model) Ciclo de vida de desarrollo software en el cual un proyecto es descompuesto en una serie de incrementos, cada uno de los cuales suministra una porción de la funcionalidad respecto de la totalidad de los requisitos del proyecto. Los requisitos tienen asignada una prioridad y son entregados según el orden de prioridad en el incremento correspondiente. En algunas (pero no en todas) versiones de este modelo de ciclo de vida, cada sub proyecto sigue un ""mini-modelo V"" con sus propias fases de diseño, codificación y pruebas.

**Modelo de Desarrollo Iterativo** (Iterative Development Model) Ciclo de vida del desarrollo donde un proyecto es dividido en un número de iteraciones, generalmente grande. Una iteración es un ciclo completo de desarrollo que tiene como resultado una entrega (interna o externa) de un producto ejecutable, un subconjunto del producto final en desarrollo, que crece de iteración en iteración para convertirse en el producto final.

**Modelo-V** (V-model) Marco de trabajo para describir las actividades del ciclo de vida de desarrollo software desde la especificación de requisitos hasta el mantenimiento. El modelo-V ilustra cómo las actividades de del proceso de pruebas pueden ser integradas dentro de cada fase del ciclo de vida de desarrollo software.

**Moderador** (Moderator) Líder y principal persona responsable de una inspección u otro proceso de revisión.

**Modo de Fallo** (Failure Mode) Manifestación física o funcional de un fallo. Por ejemplo, un sistema en modo de fallo puede estar caracterizado por operar con lentitud, presentar salidas incorrectas o una finalización completa de la ejecución.

**Monitorización de Pruebas** (Test Monitoring) Tarea de gestión de pruebas que se ocupa de las actividades relacionadas con la comprobación periódica del estado de un proyecto de pruebas. Se preparan informes que comparan el estado real con lo que fue planificado.

**Nivel de Prueba / Etapa de Prueba** (Test Level / Test stage) Grupo de actividades que están organizadas y gestionadas de forma conjunta. Un nivel de prueba esta vinculado con las responsabilidades en un proyecto. Ejemplos de niveles de pruebas son las pruebas de componente, pruebas de integración, pruebas de sistema y pruebas de aceptación.

**Nota de la Entrega / Informe de Transmisión de Elemento de Prueba** (Release Note / Test Item Transmittal Report) Documento en el que se identifican elementos de prueba, su configuración, estado actual y otra información adicional de la entrega proporcionada por desarrollo para pruebas y, posiblemente otros afectados, al principio de la fase de ejecución de pruebas.

**Objetivo de Prueba** (Test Objective) Razón o propósito para el diseño y la ejecución de una prueba.

**Objeto de Prueba** / **Elemento de Prueba** (Test Object / Test Item) Componente o sistema a ser probado.Normalmente hay un objeto de prueba y muchos elementos de prueba.

**Operabilidad**(Operability) Capacidad de un producto software para permitir al usuario manejarlo y controlarlo.

**Oráculo de Prueba** (Test Oracle) Fuente para determinar resultados esperados para compararlos con los resultados reales del software en pruebas. Un oráculo puede ser un sistema existente (para una evaluación comparativa), un manual de usuario o el conocimiento especializado de un individuo, pero no debería ser el código.

**Par Definición-Uso** (Definition-Use Pair) Asociación entre la definición de una variable y el uso de esa variable. Los usos de las variables incluyen operaciones (por ejemplo multiplicación) o dirigir la practica (ejecución/recorrido) de un camino (uso de ""predicado"").

**Partición de Equivalencia** (Equivalence Partition) Porción del dominio de una entrada o una salida para la cual se asume que el comportamiento de un componente o sistema, basado en la especificación, es el mismo.

**Pasar / Pasado de Prueba** (Pass / Test Pass) Se considera que una prueba pasa si su resultado real coincide con su resultado esperado.

**Plan de Prueba** (Test Plan) Documento que describe el alcance, enfoque, los recursos y planificación de las actividades de pruebas previstas. Identifica, entre otros, los elementos de prueba, las prestaciones a ser probadas, las tareas de pruebas, quien realiza cada tarea, el grado de independencia del probador, el entorno de pruebas, las técnicas de diseño de pruebas y los criterios de entrada y salida a utilizar, y los motivos para cada elección, y cualquier riesgo que requiera un plan de contingencia. Es un registro del proceso de planificación de pruebas.

**Plan de Pruebas de Fase** (Phase Test Plan) Plan de pruebas que normalmente se refiere a una fase de pruebas.

**Plan de Pruebas de Nivel** (Level Test Plan) Plan de pruebas que normalmente hace referencia a un único nivel de pruebas. Véase también plan de pruebas.

**Plan Maestro de Pruebas / Plan de Pruebas de Proyecto** (Master Test Plan / Project Test Plan) Plan de pruebas que normalmente aborda múltiples niveles de prueba.

**Planificación de Prueba** (Test Planning) La actividad de establecer o actualizar un plan de prueba.

**Política de Pruebas** (Test Policy) Documento de alto nivel que describe los principios, el enfoque y los principales objetivos de la organización en lo referente a las pruebas.

**Porcentaje de Detección de Defectos PDD** (Defect Detection Percentage DDP) Número de defectos encontrados por una fase de pruebas dividido por el número de defectos encontrados por esa fase de pruebas y, posteriormente, por cualesquiera otros medios.

**Portabilidad** (Portability) Facilidad con la que un producto software puede ser transferido de un entorno hardware o software a otro.

**Postcondición** (Postcondition) Condiciones de entorno y estado que deben ser satisfechas tras la ejecución de una prueba o un procedimiento de pruebas.

**Precondición** (Precondition) Condiciones de entorno y estado que deben ser satisfechas antes de que un componente o sistema puede ser ejecutado con una prueba particular o un procedimiento de pruebas.

**Predicción de Error** (Error guessing) Técnica de diseño de pruebas donde la experiencia de quien prueba es utilizada para anticipar qué defectos podrían estar presentes en el componente o sistema en prueba como resultado de los errores cometidos, y diseñar pruebas específicas para ponerlos al descubierto.

**Prioridad** (Priority) Nivel de importancia (negocio) asignado a un elemento, por ejemplo a un defecto.

**Probador** (Tester) Profesional experto que esta involucrado en las pruebas de un componente o sistema.

**Probar** (Testing) Proceso que consiste en todas las actividades del ciclo de vida software, tanto estáticas como dinámicas, concernientes con la planificación, preparación y evaluación de productos software y los productos de trabajo relacionados para determinar que éstos satisfacen los requisitos especificados, para demostrar que se ajustan al propósito y para detectar defectos.

**Procedimiento de Prueba** (Test Procedure) Documento que especifica la secuencia de acciones para la ejecución de una prueba. También conocido como script de prueba o script de prueba manual.

**Proceso de prueba** (Test process) El proceso de prueba básico comprende la planificación y el control de pruebas, análisis y diseño de pruebas, implementación y ejecución de pruebas, evaluación de criterios de salida y elaboración de informes, y actividades de cierre de prueba.

**Producto de Soporte de Prueba** (Testware) Artefactos producidos durante el proceso de pruebas necesarios para la planificación, el diseño y la ejecución pruebas, tales como la documentación, scripts, entradas, resultados esperados, procedimientos de configuración y despejado, archivos, bases de datos, entorno y cualquier software o utilidades adicionales utilizadas en pruebas.

**Programación por Pares** (Pair Programming) Enfoque de desarrollo software por el que las líneas del código (producción y/o pruebas) de un componente son escritas por dos programadores situados en un mismo ordenador. Esto significa, de forma implícita, que se realizan constantemente revisiones de código en tiempo real.

**Prueba Comparativa** (Benchmark Test) (1) Un estándar contra el cual se pueden realizar medidas o comparaciones. (2) Una prueba utilizada para comparar componentes o sistemas entre si o con un estándar como en (1).

**Prueba de Admisión / Prueba Previa** (Intake Test / Pretest) Caso especial de pruebas de humo cuyo objetivo es decidir si el componente o sistema esta en condiciones de ser probado en detalle y proseguir con el proceso de pruebas. Una prueba de admisión se realiza normalmente al inicio de la fase de ejecución de pruebas.

**Prueba de Ciclo de Proceso** (Process Cycle Test) Técnica de diseño de pruebas de caja negra en la cual los casos de prueba están diseñados para ejecutar procedimientos y procesos de negocio.

**Pruebas Ágiles** (Agile Testing) Práctica de pruebas para un proyecto que utiliza metodologías ágiles, tales como programación extrema (XP), tratando al desarrollo como el cliente del proceso de pruebas y enfatizando el paradigma de diseño de primero-probar.

**Pruebas Aleatorias** (Random Testing) Técnica de diseño de pruebas de caja negra donde se seleccionan casos de prueba, posiblemente usando un algoritmo de generación pseudo-aleatorio, para asemejar a un perfil operativo. Esta técnica se puede usar para probar atributos no funcionales como la fiabilidad y el rendimiento.

**Pruebas Alfa** (Alpha Testing) Pruebas simuladas u operacionales realizadas por usuarios/clientes potenciales o por un equipo de pruebas independiente en las dependencias de desarrollo, pero fuera de la organización de desarrollo. Las pruebas alfa son utilizadas con frecuencia para software de distribución masiva como una forma de pruebas de aceptación internas.

**Pruebas AlgorÍtmicas** **[TMap]** / Pruebas de Rama (Algorithm test [TMap]) Es una técnica de diseño de pruebas de caja blanca en la cual los casos de prueba son diseñados para ejecutar ramas.

**Pruebas Basadas en Requisitos** (Requirements-Based Testing) Enfoque de pruebas en el cual se diseñan los casos de prueba en base a los objetivos de las pruebas y condiciones de prueba derivados de los requisitos, por ejemplo pruebas que practican funciones específicas o prueban atributos no funcionales tales como la fiabilidad o la usabilidad.

**Pruebas basadas en Riesgos** (Risk-based testing) Enfoque de pruebas para reducir el nivel de riesgos de producto e informar a los afectados de su estado, comenzando desde las fases iniciales de un proyecto. Implica la identificación de riesgos de producto y su uso para dirigir el proceso de pruebas.

**Pruebas Beta** (Beta Testing) Pruebas operacionales realizadas por usuarios/clientes potenciales y/o existentes, en un sitio externo no relacionado de ninguna manera con los desarrolladores, para determinar si un componente o sistema satisface o no las necesidades del usuario/cliente y se ajusta a los procesos de negocio. Con frecuencia las pruebas beta se emplean como una forma de prueba de aceptación externa para software de distribución masiva con el objetivo de obtener la respuesta del mercado.

**Pruebas Caja Blanca** (White-Box Testing) Pruebas basadas en un análisis de la estructura interna del componente o sistema.

**Pruebas de Accesibilidad** (Accessibility Testing) Pruebas para determinar la facilidad con la que un usuario con discapacidades puede usar un componte o sistema.

**Pruebas de Aceptación** (Acceptance Testing) Pruebas formales con respecto a las necesidades de usuario, requisitos y procesos de negocio dirigidas a determinar si el sistema satisface o no los criterios de aceptación y a habilitar al usuario, cliente u otra entidad autorizada a determinar si acepta o no el sistema.

**Pruebas de Camino** (Path Testing) Técnica de diseño de pruebas de caja blanca en la cual los casos de prueba están diseñados para ejecutar caminos.

**Pruebas de Campo / Pruebas Beta** (Field Testing / Beta Testing) Pruebas operacionales realizadas por usuarios/clientes potenciales y/o existentes, en un sitio externo no relacionado de ninguna manera con los desarrolladores, para determinar si un componente o sistema satisface o no las necesidades del usuario/cliente y se ajusta a los procesos de negocio. Con frecuencia las pruebas beta se emplean como una forma de prueba de aceptación externa para software de distribución masiva con el objetivo de obtener la respuesta del mercado.

**Pruebas de Carga** (Load testing) Tipo de prueba relacionado con medida del comportamiento de un componente o sistema con una carga creciente, por ejemplo el número de usuarios concurrentes y/o número de transacciones para determinar qué carga puede ser soportada por el componente o sistema. Véase también pruebas de estrés.

**Pruebas de Caso de Uso** (Use Case Testing) Técnica de diseño de prueba de caja negra en la que los casos de prueba están diseñados para ejecutar escenarios de usuario.

**Pruebas de Componente** (Component Testing) Pruebas de componentes software individuales.

**Pruebas de Concurrencia** (Concurrency Testing) Pruebas para determinar cómo trata un componente o sistema la ocurrencia de dos o más actividades dentro del mismo intervalo de tiempo, llevadas a cabo por intercalado de las actividades o por ejecución simultánea.

**Pruebas de Condición Múltiple** (Multiple Condition Testing) Técnica de diseño de pruebas de caja blanca en la cual los casos de prueba se diseñan para ejecutar combinaciones de resultados de condiciones simples (dentro de una sentencia).

**Pruebas de Condición-Decisión** (Decision Condition Testing) Técnica de diseño de pruebas de caja blanca en la cual los casos de prueba son diseñados para practicar resultados de condición y resultados de decisión.

**Pruebas de Confirmación / Repetición de pruebas** (Confirmation Testing / Re-Testing) Pruebas que ejecutan aquellos casos de prueba que hubieran fallado la última vez que fueran ejecutados con el objetivo de verificar el éxito de acciones correctivas.

**Pruebas de Cumplimiento** (Compliance testing) Proceso de pruebas para determinar el cumplimiento de un componente o sistema.

**Pruebas de Decisión** (Decision Testing) Técnica de diseño de casos de prueba de caja blanca en la cual los casos de prueba se diseñan para practicar resultados de decisiones."

**Pruebas de Decisión de Condición Modificada / Pruebas de Determinación de Condición** (Modified Condition Decision Testing / Condition Determination Testing) Técnica de diseño de pruebas de caja blanca en la cual los casos de prueba son diseñados para ejecutar salidas de condición unitarias que afectan de forma independiente al resultado de la decisión.

**Pruebas de Escalabilidad** (Scalability Testing) Pruebas para determinar la escalabilidad de un determinado producto software.

**Pruebas de Estrés** (Stress testing) Pruebas orientadas a evaluar un componente o sistema en o más allá de los límites especificados en los requisitos.

**Pruebas de Flujo de Datos** (Data Flow Testing) Técnica de diseño de pruebas de caja blanca, en la que los casos de prueba se diseñan para ejecutar pares definición-uso de variables.

**Pruebas de Humo / Prueba de Confianza/ Pruebas de Circunspección** (Smoke Test / Confidence Test / Sanity Test) Subconjunto de todos los casos de prueba definidos/planificados que cubren la funcionalidad principal de un componente o sistema, con el objeto de asegurar que las funciones cruciales de un programa funcionan, pero sin preocuparse por los detalles finos. Una contrucción diaria y pruebas de humo pertenecen a las mejores prácticas de la industria.

**Pruebas de Integración** (Integration Testing) Pruebas realizadas con el objeto de poner en evidencia defectos en las interfaces e interacciones entre componentes o sistemas integrados.

**Pruebas de Integración de Componente** (Component Integration Testing) Pruebas ejecutadas con el objeto exponer defectos en las interfaces e interacciones entre elementos integrados.

**Pruebas de Integración de Sistema** (System Integration Testing) Pruebas de la integración de sistemas y paquetes; pruebas de interfaces con organizaciones externas (por ejemplo Intercambio de Datos Electrónicos, Internet)

**Pruebas de Interfaz** (Interface Testing) Tipo de prueba de integración que se ocupa de probar las interfaces entre componentes o sistemas.

**Pruebas de Mantenimiento** (Maintenance Testing) Pruebas de los cambios en un sistema en operación o el impacto de un entorno modificado para un sistema en operación."

**Pruebas de Perfil Operativo** (Operational Profile Testing) Pruebas estadísticas que utilizan un modelo de operaciones del sistema (tareas de corta duración) y su probabilidad de uso típico.

**Pruebas de Regresión** (Regression Testing) Pruebas de un programa previamente probado que ha sufrido modificaciones, para asegurarse que no se han introducido o descubierto defectos en áreas del software que no han sido modificadas como resultado de los cambios realizados. Se realiza cuando el software o su entorno han sido modificados.

**Pruebas de Rendimiento** (Performance Testing) Proceso de pruebas para determinar el rendimiento de un producto software. Véase también pruebas de eficiencia.

**Pruebas de Seguridad** (Security Testing) Pruebas para determinar la seguridad de un producto software.

**Pruebas de Sentencia** (Statement Testing) Técnica de diseño de pruebas de caja blanca en la cual los casos de prueba se diseñan para ejecutar sentencias.

**Pruebas de Sistema** (System Testing) Proceso de probar un sistema integrado para verificar que cumple los requisitos especificados.

**Pruebas de Transición de Estado** (State Transition Testing) Técnica de diseño de pruebas de caja negra en la cual los casos de prueba son diseñados para ejecutar transiciones de estado válidas e inválidas.

**Pruebas de Usabilidad** (Usability testing) Pruebas para determinar en que medida el producto software es comprensible, fácil de aprender, fácil de operar y atractivo a los usuarios bajo condiciones especificadas.

**Pruebas de Usuario** (User Test) Prueba a través de la cual se involucra a usuarios reales en la evaluación de la usabilidad de un componente o sistema.

**Pruebas Dinámicas** (Dynamic Testing) Pruebas que implican la ejecución del software de un componente o sistema.

**Pruebas Estáticas** (Static Testing) Pruebas de un componente o sistema a nivel de especificación o implementación sin ejecutar el software, por ejemplo revisiones o análisis estático de código.

**Pruebas Exhaustivas** (Exhaustive Testing) Enfoque de pruebas donde el conjunto de pruebas abarca todas las combinaciones de valores de entrada y pre condiciones.

**Pruebas Exploratorias** (Exploratory Testing) Técnica informal de diseño de pruebas donde quien prueba controla activamente el diseño de las pruebas a medida que las pruebas son realizadas y utiliza la información obtenida durante las pruebas para diseñar unas nuevas y mejores.

**Pruebas Funcionales** (Functional Testing) Pruebas basadas en el análisis de las especificaciones funcionales de un componente o de un sistema.

**Pruebas Guiadas por Datos** (Data Driven Testing) Técnica basada la creación de scripts que almacena la entrada de la prueba y los resultados esperados en una tabla o una hoja de cálculo, de tal manera que un solo script pueda ejecutar todas las pruebas de la tabla. Las pruebas guiadas por datos a menudo se utilizan para dar soporte en el uso de herramientas de ejecución de pruebas, tales como herramientas de captura/reproducción.

**Pruebas Guiadas por Palabras Clave** (Keyword Driven Testing) Técnica de creación de scripts que utiliza archivos de datos para contener no solamente datos de prueba y resultados esperados, sino también palabras claves que están relacionadas con la aplicación que esta siendo probada. Las palabras claves son interpretadas por scripts especiales de soporte que son invocados por el script de control para la prueba. Vea también pruebas guiadas por datos.

**Pruebas Negativas** (Negative Testing) Pruebas destinadas a mostrar que un componente o sistema no funciona. Las pruebas negativas están relacionadas con la actitud del probador más que a un enfoque de pruebas o técnica de diseño de pruebas específicos, por ejemplo probar con valores de entrada inválidos o excepciones.

**Pruebas No Funcionales** (Non-Functional Testing) Pruebas de atributos de un componente o sistema que no se refieren a la funcionalidad, por ejemplo fiabilidad, eficiencia, usabilidad, mantenibilidad y portabilidad.

**Pruebas Operativas** (Operational Testing) Pruebas dirigidas a evaluar a un componente o sistema en su entorno de operaciones.

**Pruebas por Pares** (Pair Testing) Dos personas, por ejemplo dos probadores, un desarrollador y un probador o un usuario final y un probador, trabajando juntas para detectar defectos. Normalmente comparten un único ordenador e intercambian el control del mismo durante las pruebas.

**Pruebas Programadas**(Scripted Testing) Ejecución de pruebas llevadas a cabo siguiendo una secuencia de pruebas documentadas con anterioridad.

**Pruebas Secuenciadas** (Back-To-Back Testing) Pruebas en las que dos o más variantes de un componente o sistema son ejecutados con las mismas entradas, las salidas son comparadas y analizadas en caso de discrepancias.

**Pseudo-Aleatoria** (Pseudo-Random) Serie que parece ser aleatoria pero que de hecho es generada de acuerdo a una secuencia predeterminada.

**Rama**(Branch) Bloque básico que puede ser seleccionado para su ejecución en base a una estructura propia de un programa en la cual están disponibles uno de dos o más caminos alternativos, por ejemplo case, jump, go to, ifthen-else.

**Recuperabilidad / Fiabilidad** (Recoverability / Reliability) Capacidad de un producto software para re establecer un nivel específico de rendimiento y recuperar la información directamente afectada en caso de fallo. Fiabilidad, habilidad de un producto software para llevar a cabo aquellas funciones requeridas en condiciones establecidas para un período de tiempo específico, o para un número específico de operaciones.

**Reemplazabilidad** (Replaceability) Capacidad de un producto software para ser usado en lugar de otro software especificado, con el mismo propósito en el mismo entorno.

**Registrar Pruebas / Registro de Pruebas** (Test Logging /Test record) Proceso de registro de información respecto a las pruebas ejecutadas en un registro de pruebas.

**Registro de Pruebas / Registro de Ejecución de Pruebas** (Test Log / Test Run Log) Registro cronológico de los detalles relevantes respecto a la ejecución de pruebas.

**Representación Causa-Efecto** (Cause-Effect Graphing) Técnica de diseño de pruebas de caja negra en la cual los casos de prueba se diseñan a partir de gráficos causa-efecto.

**Reproducibilidad de una Prueba** (Test Reproduceability) Atributo de una prueba que indica si se generan los mismos resultados cada vez que se ejecuta la prueba.

**Requisito** (Requirement) Condición o capacidad necesaria para un usuario con el objeto de solucionar un problema o lograr un objetivo que debe ser alcanzado o poseído por un sistema o componente de un sistema, para satisfacer un contrato, estándar, especificación u otro documento impuesto formalmente.

**Resultado de Decisión** (Decision Outcome) Resultado de una decisión (la cual determina las ramas a seguir).

**Resultado de Prueba / Resultado** (Test Result / Result) Consecuencia/efecto de la ejecución de una prueba. Incluye salidas por pantalla, cambios en datos, informes y mensajes de comunicación emitidos. Véase también resultado real, resultado esperado.

**Resultado Esperado** (Expected Result) Comportamiento previsto por la especificación, u otra fuente, del componente o del sistema bajo condiciones especificadas.

**Resultado Real** (Actual Result) El comportamiento producido/observado cuando un componente o sistema es probado.

**Revisión Ad Hoc / Revisión informal** (Ad Hoc Review / Informal Review) Revisión que no está basada en un procedimiento formal (documentado).

**Revisión Entre Pares** (Peer Review) Revisión de un producto de trabajo software por parte de compañeros de trabajo del desarrollador del producto con el objeto de identificar defectos y mejoras. Ejemplos de este tipo de revisión son la inspección, revisión técnica y revisión guiada.

**Revisión Formal** (Formal Review) Revisión caracterizada por procedimientos y requisitos documentados, por ejemplo la inspección.

**Revisión Guiada** (Walkthrough) Presentación paso a paso de un documento por parte del autor con el fin de recoger información y establecer un entendimiento común de su contenido.

**Revisión Informal** (Informal Review) Revisión que no está basada en un procedimiento formal (documentado).

**Revisión Técnica** (Technical Review) Actividad de discusión de grupo de pares que se centra en alcanzar consenso respecto del enfoque técnico a adoptar.

**Revisor** (Reviewer) Persona involucrada en la revisión, responsable de la identificación y descripción de anomalías en el producto o proyecto bajo revisión. Los revisores pueden ser seleccionados para representar diferentes puntos de vista y roles dentro del proceso de revisión.

**Riesgo de Producto / Riesgo** (Product Risk / Risk) Riesgo directamente relacionado con el objeto de prueba. Riesgo, Factor que puede resultar en futuras consecuencias negativas, expresada normalmente como impacto y probabilidad.

**Riesgo de Proyecto** (Project Risk) Riesgo relativo a la gestión y control de un proyecto (de pruebas), por ejemplo, falta de personal, plazos estrictos, requisitos cambiantes, etc.

**Robustez** (Robustness) Grado en el cual un componente o sistema puede funcionar correctamente en presencia de entradas no válidas o condiciones de entorno de estrés. Véase también tolerancia al error, tolerancia a faltas.

**Salida** (Output) Variable (tanto si se almacena dentro como fuera de un componente) que es escrita por un componente.

**Seguridad** (Security) Atributos de productos software que se refieren a su capacidad para prevenir accesos no autorizados, sean accidentales o deliberados, a programas y datos.

**Sentencia** (Statement) Entidad propia de un lenguaje de programación que es generalmente la unidad mínima e indivisible de ejecución.

**Sentencia Ejecutable** (Executable Statement) Sentencia que, una vez compilada, es traducida a código objeto y que se ejecutará cuando el programa esté funcionando, pudiendo realizar acciones sobre datos.

**Severidad** (Severity) Grado de impacto que un defecto tiene en el desarrollo u operación de un componente del sistema.

**Simulador** (Simulator) Dispositivo, programa o sistema usado durante las pruebas, que se comporta u opera como un sistema dado cuando se le proporciona un conjunto de entradas controladas.

**Stub** (Stub) Implantación estructural (esqueleto) o de propósito especial de un componente software, usado para desarrollar o probar un componente al que se llama o es dependiente de él de algún modo. Sustituye a un componente llamado.

**Tabla de Decisión** (Decision Table) Tabla que muestra las combinaciones de entradas y/o estímulos (causas) con sus salidas y/o acciones asociadas (efectos), que puede ser utilizada para diseñar casos de prueba.

**Tabla de Estado** (State Table) Tabla que muestra las transiciones resultantes para cada estado combinado con cada posible evento, mostrando tanto transiciones válidas como no válidas.

**Técnica de Diseño de Prueba / Técnica de Especificación de Prueba** (Test Design Technique / Test Specification Technique) Procedimiento utilizado para obtener y/o seleccionar casos de prueba.

**Técnica de Diseño de Prueba** (Test Design Technique) Procedimiento utilizado para obtener y/o seleccionar casos de prueba.

**Técnica de Diseño de Prueba de Caja Negra** ( Black-box test design technique) Procedimiento para obtener y/o seleccionar casos de prueba basados en el análisis de la especificación, tanto funcional como no funcional de un componente o sistema sin referencia a su estructura interna.

**Técnica de Diseño de Prueba Funcional** (Functional Test Design Technique) Procedimiento utilizado para obtener y/o seleccionar casos de prueba basado en un análisis de la especificación funcional de un componente o de un sistema sin hacer referencia a su estructura interna.

**Técnica de Diseño de Pruebas Basada en la Experiencia** (Experience-Based Test Design Technique) Procedimiento para derivar y/o seleccionar casos de prueba basados en la experiencia, conocimiento e intuición del probador.

**Técnica de Ejecución de Pruebas** (Test Execution Technique) Método utilizado para llevar a cabo la ejecución de la prueba real , ya sea de forma manual o automática.

**Técnica de Prueba / Técnica de Diseño de Prueba** (Test technique / Test Design Technique) Procedimiento utilizado para obtener y/o seleccionar casos de prueba.

**Técnicas de Diseño de Pruebas No Funcionales** ( Non-Functional Test Design Techniques) Procedimiento para obtener y/o seleccionar casos de prueba para pruebas no funcionales basadas en el análisis de las especificaciones de un componente o sistema sin hacer referencia a su estructura interna.

**Test Maturity Model TMM** Marco de trabajo de cinco niveles para la mejora del proceso de pruebas, relacionado con el "Capability Maturity Model" (CMM) que describe los elementos claves de un proceso de pruebas efectivo.

**Test Process Improvement (TPI)** Marco de trabajo contínuo para la mejora de los procesos de prueba, que describe los elementos clave de un proceso de prueba efectivo, especialmente destinado a pruebas de sistema y de aceptación.

**Informe Resumen de Pruebas** (Test Summary Report) Documento que resume las actividades y resultados de las pruebas. También contiene una evaluación de los correspondientes elementos de prueba respecto de los criterios de salida.

**Testabilidad** (Testability) Capacidad del producto software que hace posible que el software modificado sea probado.

**Tipo de Prueba** (Test Type) Grupo de actividades de pruebas dirigidas a probar un componente o sistema orientado a un objetivo de prueba específico, por ejemplo pruebas funcionales, pruebas de usabilidad, pruebas de regresión, etc. Un tipo de prueba puede tener lugar en uno o más niveles de prueba o fases de prueba.

**Tolerancia a Faltas** (Fault Tolerance) Capacidad del producto software para mantener un nivel especificado de rendimiento en caso de producirse faltas (defectos) en el software o infracción de su interfaz especificada.

**Tolerancia al Error** (Error tolerance) Capacidad de un sistema o componente para continuar su operación normal a pesar de la presencia de entradas erróneas."

**Trazabilidad**(Traceability) Capacidad de identificar elementos relacionados en la documentación y el software, tales como requisitos con las pruebas asociadas. Véase también trazabilidad horizontal y trazabilidad vertical.

**Trazabilidad Horizontal** (Horizontal Traceability) Trazas de requisitos para un nivel de pruebas a través de las capas de la documentación de pruebas (por ejemplo, plan de pruebas, especificación de diseño de prueba, especificación de caso de prueba y especificación de procedimiento de prueba o script de pruebas).

**Usabilidad** (Usability) Capacidad del software para ser comprendido, aprendido, utilizado y atractivo al usuario cuando es utilizado bajo condiciones especificadas.

**Utilización de Recursos** (Resource Utilization) Capacidad de un producto software para hacer uso de cantidades y tipos de recursos apropiados, por ejemplo cantidad de memoria principal y secundaria utilizada por el programa y tamaños requeridos de archivos temporales, cuando el software realiza su función en condiciones establecidas.

**Validación** (Validation) Confirmación por examen y a través de la aportación de evidencia objetiva que han sido satisfechos los requisitos para un uso o aplicación previstos.

**Valor de Entrada** (Input Value) Instancia de una entrada. Véase también entrada.

**Valor Límite** (Boundary Value) Valor de entrada o de salida que se encuentra en la frontera de una partición de equivalencia o a la mínima distancia incremental a cualquier lado de la frontera, por ejemplo el valor mínimo o máximo de un rango.

**Variable**(Variable) Elemento de almacenamiento en un ordenador que es accesible a un programa software al referirse a él mediante un nombre.

**Verificación**(Verification) Confirmación por exámen y a través de la aportación de evidencia objetiva que se han satisfecho los requisitos especificados.