**A**

**Actividad:** Un componente del trabajo desarrollado durante el curso de un proyecto.

**Aplicación de software:** Software diseñado para satisfacer las necesidades de un usuario.

**C**

**Cantidad de requerimientos:** El número detallado de servicios y restricciones del sistema.

**Capacidad de soporte**: Combina la capacidad de ampliar el programa (extensibilidad), adaptabilidad y servicios (los tres representan mantenimiento), así como capacidad de hacer pruebas, compatibilidad, capacidad de configuración, la facilidad de instalación de un sistema y la facilidad con que se pueden localizar los problemas.

**Capacitación del personal:** Es la adquisición de conocimientos, principalmente de carácter técnico, científico y administrativo. Consiste en una actividad planeada y basada en las necesidades reales de una empresa y orientada hacia un cambio en los conocimientos del colaborador.

**Cascada:** Toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, como especificación de requerimientos, diseño de software, implementación, pruebas, etcétera.

**Casos de uso:** Estos son una técnica que se basa en escenarios para la obtención de requerimientos. Actualmente se han convertido en una característica fundamental de la notación UML.

**Clases:** Una clase define una implementación e interfase de un objeto de software. Esta define la representación interna del objeto y define las operaciones que se pueden solicitar al objeto.

**Cliente:** La persona u organización que paga por el producto y usualmente (pero no necesariamente) decide los requerimientos. La persona u organización que desea usar el servicio, producto o resultado del proyecto.

**Código Fuente:** El código fuente es un conjunto de líneas de código que conforman un bloque de texto que normalmente genera otro código mediante un compilador o intérprete para ser ejecutado por una computadora. Normalmente se refiere a la programación de software. Un único programador o un equipo de ellos escriben el código fuente en el lenguaje de programación elegido.

**Complejidad:** El grado en el cual el diseño o implementación de un sistema es difícil de entender y verificar. Compuesta de dos dimensiones: Complejidad del Negocio y Complejidad Técnica. Cada dimensión puede ser caracterizada por un conjunto de atributos. Los atributos de negocio típicos incluyen tamaño, ubicación y riesgo financiero. Los atributos técnicos típicos incluyen el nivel de integración de la tecnología, necesidades de seguridad, estabilidad del hardware/software y la experiencia del personal. [PMITBC 2005]

**Confiabilidad:** El grado de confianza esperado del usuario en el sistema que operará y en que el sistema no “caerá” al utilizarlo normalmente. Se evalúa midiendo la frecuencia y gravedad de los fallos, la exactitud de las salidas (resultados), el tiempo medio entre fallos, la capacidad de recuperación de un fallo y la capacidad de predicción del programa.

**Construcción:** El equipo de desarrollo realiza la programación y pruebas de unidades que produce nuevo software o cambia el software existente. El término construcción de software se refiere a la creación detallada de software, a través de una combinación de codificación, verificación, pruebas de unidades, pruebas de integración y depuración.

**Costo:** El valor monetario o precio de una actividad o componente del proyecto que incluye el valor monetario de los recursos requeridos para desarrollar y completar la actividad o componente, o producir el componente. Es la cantidad de dinero que la empresa desarrolladora de software invierte en la realización de un proyecto informático. Es diferente del precio de compra, que es el valor que el cliente cancela por adquirir el producto que la empresa de software desarrolla. [RDC 2006]

**D**

**Defecto:** Resultado no deseado de las tareas realizadas en el proyecto que implican 1) volver a realizar las mismas, 2) retrasar la fecha de entrega del proyecto, y/o 3) el malfuncionamiento del software. Un problema en la especificación, diseño o implementación de un producto.

**Diseño:** Actividad durante la cual el equipo de desarrollo crea un diseño general de alto nivel, después posiblemente un diseño detallado. Esta actividad con frecuencia se traslapa con la especificación y la construcción. El proceso de definir la arquitectura, componentes, interfaces y otras características de un sistema o componente. [IEEE 610]

**Disponibilidad:** Es el grado en el cual un sistema, componente o servicio está en un estado listo para realizar una función requerida bajo determinadas condiciones en un instante de tiempo dado o bajo un intervalo de tiempo dado, asumiendo que los recursos externos necesarios son provistos.

**Documento:** Un medio y la información registrada en el, que generalmente tiene permanencia y puede ser leído por una persona o una máquina. Ejemplos incluyen los planes de administración de proyectos, especificaciones, procedimientos, estudios y manuales.

**Documentación:** La descripción narrativa y gráfica de un sistema. La documentación para un sistema de información incluye la documentación del sistema, documentación técnica, y sus procedimientos operativos.

**E**

**Esfuerzo:** El número de horas requeridas para completar una tarea, un entregable, una fase o un proyecto. El enfoque aquí es sobre la cantidad de tiempo expresado en horas.

**Especificación:** Actividad estructurada durante la cual el equipo de desarrollo trabaja con el cliente para identificar y documentar que funcionalidad, que interfaces, que calidad es requerida.

**Espiral:** Este modelo de proceso del software se representa como una espiral en la que cada ciclo se divide en 4 partes (1. Definición de objetivos, 2. Evaluación y reducción de riesgos, 3. Desarrollo y validación, 4. Planeación) La diferencia importante entre este modelo y los otros modelos de proceso del software es la consideración explícita del riesgo. La construcción de prototipos se utiliza para resolver las dudas en los requerimientos y así reducir el riesgo.

**Estándar:** Un documento establecido por consenso y aprobado por un organismo reconocido que provee, para un uso común y repetido, reglas, guías o características para actividades o sus resultados, dirigido a alcanzar el grado optimo de orden en un contexto dado.

**Estándar de codificación:** Especificaciones para un estilo de codificación específica.

**Estimación:** Un cálculo o evaluación tentativa del esfuerzo, duración, y/o costo aproximado para realizar un trabajo.

**Experiencia del personal:** Generalmente se refiere al conocimiento de como llevar a cabo una actividad o una tarea específica, en lugar del conocimiento factual.

**F**

**Facilidad de uso:** Es evaluada al considerar factores humanos, la estética general, la consistencia y la documentación.

**Fallas:** La interrupción inesperada de la capacidad de un sistema para desarrollar sus funciones requeridas.

**Fecha de inicio del proyecto:** La fecha (día, mes y año) cuando el cliente ha autorizado a la empresa que proceda a la realización del proyecto de software.

**Fecha de terminación del proyecto:** La fecha (día, mes y año) cuando la empresa de software ha realizado la entrega formal del producto de software que ha desarrollado.

**Funcionalidad:** Se determina al evaluar el conjunto de características y capacidades del programa, la generalidad de las funciones que son entregadas y la seguridad de todo el sistema.

**H**

**Herramienta:** Algo tangible, tales como una plantilla o un programa de software, utilizado para desarrollar una actividad para producir un producto o resultado.

**I**

**Inspecciones:** Una actividad semejante a medir, examinar, probar o estimar una o más características de una entidad y comparar los resultados con los requerimientos especificados para establecer si la conformidad está siendo alcanzada en cada característica.

**Instalación:** La actividad de prepararse para la instalación del software que será utilizado por los usuarios finales, a continuación trabajar con ellos para realizar la instalación, la documentación para el usuario y su capacitación en el uso adecuado del software.

**Interrupciones:** Periodos de tiempo durante el cual el personal asignado al proyecto interrumpió las labores del mismo para realizar otras actividades

**M**

**Metodología:** Un cuerpo de practicas, procedimientos, roles, responsabilidades y reglas (de una disciplina). Un conjunto de métodos de trabajo. Un sistema de prácticas, técnicas o procedimientos, y reglas usadas por aquellos que trabajan en una disciplina.

**Microsoft Solution Framework (MSF):** MSF es una estrategia disciplinada y sistemática para proyectos de tecnología basados sobre un conjunto de principios, modelos, disciplinas, conceptos, guías y practicas probadas por Microsoft.

**Módulo:** Una parte lógicamente separable de un programa. Los módulos son usualmente compilados separadamente y proveen un mecanismos de abstracción u ocultamiento de información a fin de que una implementación del modulo pueda ser cambiada sin requerir ningún cambio a otros módulos.

**P**

**Patrones de diseño:** Es una descripción del problema y la esencia de su solución de tal forma que ésta se pueda reutilizar en diferentes casos.

**Personal:** Conjunto de los empleados de una empresa.

**Planificación:** Esta actividad abarca tanto la planificación de alto nivel del proyecto y el análisis de requerimientos preliminar. Este se enfoca en los objetivos, stakeholders, riesgos, presupuestos y calendarios.

**Proyecto:** Un conjunto administrado de recursos interrelacionados que entrega uno o más productos a un consumidor o usuario. Este conjunto de recursos tiene inicio y fin definidos y generalmente operan de acuerdo a un plan. Tal plan es frecuentemente documentado y especifica el producto a ser entregado o implementado, los recursos y fondos usados, el trabajo a ser realizado y el calendario para realizar dicho trabajo.

**Proceso definido:** Un proceso repetible que tiene entradas claramente establecidas, criterios de entrada, actividades, roles, medidas, pasos de verificación, resultados y criterios de salida. Un proceso definido puede ser caracterizado por estándares, procedimientos, capacitación, herramientas y métodos.

**Pruebas:** El planear y desarrollar los diferentes niveles de pruebas sobre el software, realizado por personal que puede ser independiente de los desarrolladores. Esta actividad frecuentemente se traslapa con la actividad de construcción.

Es una actividad en la cual un sistema, producto o componente es usado bajo condiciones definidas, los resultados son observados o almacenados y una evaluación es realizada para ver si cumplen adecuadamente algunos o todos los requerimientos.

**R**

**Re-utilización:** El grado con el cual un componente u otro producto de trabajo pueden ser usados en uno o más sistemas.

**Rendimiento:** El grado en que un sistema realiza sus funciones designadas con el mínimo consumo de recursos. Se mide por la velocidad de procesamiento, el tiempo de respuesta, consumo de recursos, rendimiento efectivo total y eficacia.

**Requerimientos funcionales:** Son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares y de cómo se comportará en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer.

Describen las funciones que el software ejecutará; por ejemplo, formatear algún texto o modular una señal. Algunas veces son conocidos como capacidades.

**Requerimientos no funcionales:** Son aquellos que actúan como restricciones de la solución. Los requerimientos no funcionales a veces son conocidos como restricciones o requerimientos de calidad.

**Rol:** Una definición del comportamiento y responsabilidades de un individuo o conjunto de individuos que trabajan juntos como un equipo, dentro del contexto de una organización de ingeniería de software.

**Rotación:** Es la cantidad de personas que ingresan y que se desvinculan de la empresa durante el desarrollo del proyecto.

**RUP:** El Proceso Unificado Racional o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

**S**

**Satisfacción del usuario:** La percepción del usuario de haber utilizado correctamente su tiempo, habiendo recibido respecto de sus propias expectativas y en un determinado contexto ambiental la mejor prestación posible del servicio o producto adquirido.

**T**

**Tamaño del personal:** El número de personal que forman parte del equipo asignado al desarrollo del proyecto de software.

**Tareas:** La unidad más pequeña de trabajo sujeta a la responsabilidad de la gerencia, la cual puede incluir una o más acciones. Una tarea es una asignación bien definida de trabajo para una o más personas.

**Tiempo en corregir defectos:** El tiempo requerido expresado en horas para resolver un problema existente en el software.

**U**

**Usuario final:** Persona o personas, que operan o interactúan directamente con el sistema.

**V**

**Volumen:** El número de páginas generadas por documento desarrollado.