**Glosario**

**CMMI** es el acrónimo de Capability Maturity Model Integration y se refiere a los modelos que contienen las mejores prácticas que ayudan a las organizaciones a mejorar sus procesos. El enfoque del modelo permite evolucionar desde un proceso en crisis a un proceso controlado, estandarizado, medido y optimizado que sienta las bases de la mejora continua y permite a la organización adoptar nuevas prácticas sobre un proceso estable y controlado que está institucionalizado.

**MOPROSOFT** es el Modelo de Procesos para la Industria del Software. Un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por la Asociación Mexicana para la Calidad en Ingeniería de Software a través de la Facultad de Ciencias de la Universidad Nacional Autónoma de México (UNAM) y a solicitud de la Secretaría de Economía para obtener una norma mexicana que resulte apropiada a las características de tamaño de la gran mayoría de empresas mexicanas de desarrollo y mantenimiento de software.

La guía del **PMBOK** es un instrumento desarrollado por el Project Management Institute (o PMI), que establece un criterio de buenas prácticas relacionadas con la gestión, la administración y la dirección de proyectos mediante la implementación de técnicas y herramientas que permiten identificar un conjunto de 47 procesos, distribuidos a su turno en 5 macro procesos generales. PMBOK son las siglas de Project Management Body of Knowledge, y la realización de su guía es, como decíamos, responsabilidad del Project Management Institute (PMI).

Existen diferentes modelos y metodologías que han sido en los últimos años herramientas de apoyo para el desarrollo del software. Someerville (2005), menciona que:

•**Modelo de desarrollo de software**: es una representación simplificada del proceso para el desarrollo de software, presentada desde una perspectiva específica. Es una representación abstracta de un proceso. Cada modelo representa un proceso desde una perspectiva particular y así proporcione información parcial sobre el proceso.

•**Metodología de desarrollo de software**: es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos.

Metodologías Tradicionales:

Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

**Rational Rose Enterprise Edition**. Es una herramienta CASE (Computer – Arded Software Engineering), traducido al español como Ingeniería Asistida por Computadora, desarrollada por Rational Corporation basada en el Lenguaje Unificado de Modelación (UML), que permite crear los diagramas que se van generando durante el proceso de Ingeniería en el Desarrollo del Software.

Metodologías Agiles:

Proporcionan una serie de pautas y principios junto a técnicas pragmáticas que puede que no curen todos los males pero harán la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega.

**Scrum** es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación. Con la metodología Scrum el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema.

Un **sistema de control de versiones** es una herramienta que registra todos los cambios hechos en uno o más proyectos, guardando así versiones del producto en todas sus fases del desarrollo. Las versiones son como fotografías que registran su estado en ese momento del tiempo y se van guardando a medida que se hacen modificaciones al código fuente. Para un desarrollador esta herramienta es muy valiosa porque permite viajar atrás en el tiempo (hacer roll back) si los cambios aplicados no resultaron de la manera que se esperaba, pudiendo restaurar en cualquier momento una versión previa. Es como un respaldo permanente.

El **control del proceso** consistirá en la recepción de unas entradas, variables del proceso, su procesamiento y comparación con unos valores predeterminados por el usuario, y posterior corrección en caso de que se haya producido alguna desviación respecto al valor preestablecido de algún parámetro de proceso

**Jira** es una plataforma empresarial que permite implementar flujos orientados a una amplia gama de procesos tales como: desarrollo de software, mesa de ayuda, control de pedidos de cambio, resolución de incidentes y soporte técnico, etc. y también otros tipos de procesos de negocio, no necesariamente relacionados con el área TI de la empresa. Está basado en Java y para su instalación se puede usar cualquier Base de Datos ya sea opensource o comercial. Una característica que diferencia a JIRA de otros productos es la gran cantidad de plugins disponibles (tanto comerciales como opensource) que aumentan la funcionalidad de Jira.