TAREA 4

Extreme Programing:

La programación extrema, o Extreme Programming (XP), es una metodología de desarrollo ágil, una de las más exitosas en tiempo reciente. Su autor principal es Kent Beck, quien eligió algunas características de otras metodologías y las relacionó de forma que cada una complementara a la otra.
Así, la XP se puede definir como un conjunto de pasos de diversas metodologías, acopladas de manera que sean pasos flexibles a seguir utilizadas con el uso común, para realizar un desarrollo más agradable y sencillo.
Esta metodología tiene como base la simplicidad y como objetivo principal la satisfacción del cliente; para lograrlo se deben tomar en cuenta cuatro valores fundamentales:

Comunicación
Es muy importante que haya una comunicación constante con el cliente y dentro de todo el equipo de trabajo, de esto dependerá que el desarrollo se lleve a cabo de una manera sencilla, entendible y que se entregue al cliente lo que necesita.

Simplicidad
En la XP se refiere que ante todo y sin importar qué funcionalidad requiera el usuario en su sistema, éste debe ser fácil. El diseño debe ser sencillo y amigable al usuario, el código debe ser simple y entendible, programando sólo lo necesario y lo que se utilizará.

Retroalimentación
Es la comunicación constante entre el desarrollador y el usuario.

Coraje
Se refiere a la valentía que se debe tener al modificar o eliminar el código que se realizó con tanto esfuerzo; el desarrollador debe saber cuándo el código que desarrolló no es útil en el sistema y, por lo mismo, debe ser eliminado. También se refiere a tener la persistencia para resolver los errores en la programación.

Dentro de la programación extrema se tiene 12 principios que llevan o guían el desarrollo con esta metodología:
1. El principio de pruebas
2. Proceso de planificación
3. El cliente en el lugar
4. Programación en parejas
5. Integración continua
6. Refactorización
7. Entregas pequeñas
8. Diseño simple
9. Metáfora
10. Propiedad colectiva del código
11. Estándar de codificación
12. La semana de 40 horas

Agile Unified Process:

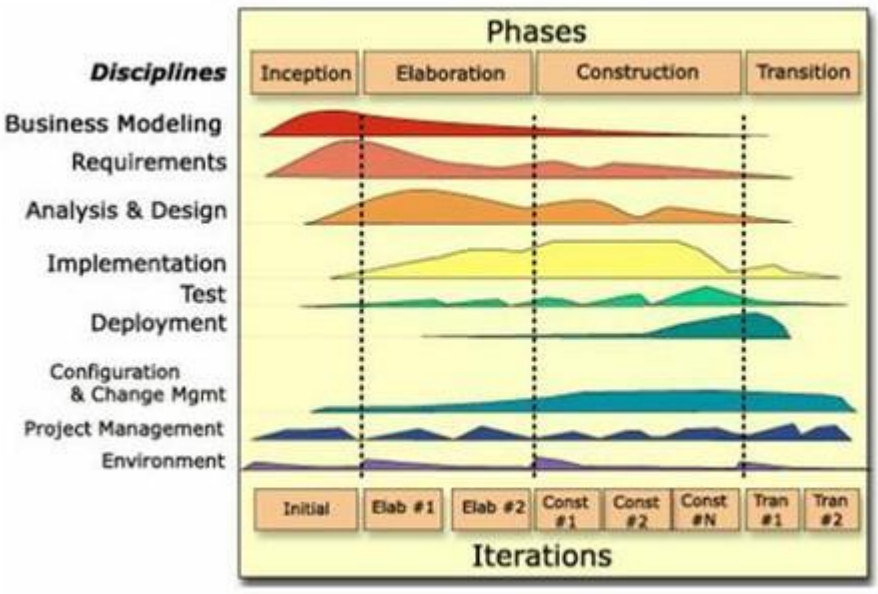
El Proceso Unificado Agil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Agil, Gestión de Cambios Agil, y Refactorización de Base de Datos para mejorar la productividad.

El proceso unificado (*Unified Process* o UP) es un marco de desarrollo software iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto software. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (*Rational Unified Process*) de IBM.

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iníciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.

CICLO DE VIDA DEL PROCESO UNIFICADO AGIL (AUP).-



Agile Modeling:

Modelado Ágil (AM) es una metodología basada en la práctica para modelado efectivo de sistemas de software. La metodología AM es una colección de prácticas, guiadas por principios y valores que pueden ser aplicados por profesionales de software en el día a día. AM no es un proceso prescriptivo, ni define procedimientos detallados de cómo crear un tipo de modelo dado. En lugar de eso, sugiere prácticas para ser un modelador efectivo. Es "suave al tacto “no es duro y es rápido piense en AM como un arte, no una ciencia. AM tiene tres objetivos:

1. Definir y mostrar cómo poner en práctica una colección de valores, principios y prácticas que conlleven a un modelado ligero efectivo.

2. Explorar la aplicación de técnicas de modelado en proyectos de software a través de un enfoque ágil, tal como XP, DSDM o SCRUM.

3. Explorar el cómo mejorar el modelado bajo procesos prescriptivos, tales como el Proceso Rational Unificado (RUP)

Principios

Principios centrales de AM

* Asumir simplicidad.

*Bienvenido el cambio.

* Permitir el siguiente esfuerzo es el objetivo

secundario.

*Cambio incremental.

* Maximizar la inversión de las partes

interesadas en el proyecto.

* Modelar con un propósito.

* Múltiples modelos.

* Trabajo de calidad.

* Rápida retroalimentación.

* El software es el objetivo primario.

* Viaje con poco equipaje