XP PROGRAMACION EXTREMA

INTEGRANTES:

- TONCCOCHI SANI RUTH KARINA
- CCOPA YAPURA DAVID
- PFUÑO CANALES LUIS ANGEL

- La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, Programación eXtrema explicada: Aceptando el cambio (Extreme Programming Explained: Embrace Change) (1999).

¿QUE ES XP?

La programación extrema usa un efoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y practicas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas.

XP inicia cerca de 1980 escrito por kent beck y es la variante mas utilizada en el mundo agil.

VALORES XP

COMUNICACIÓN:

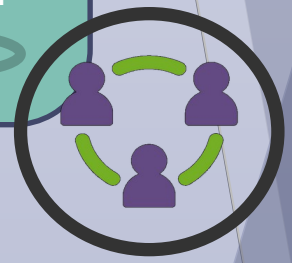
Eficaz entre los ingenieros de software y otros participantes. XP pone el énfasis en la colaboración estrecha pero informal (verbal entre los clientes).

SIMPLICIDAD

XP restringe a los desarrolladores para que diseñen solo para las necesidades inmediatas, en lugar de considerar los del futuro.

RETROALIMENTACION

Se obtiene de tres fuentes el software implementado, el cliente y otros miembros del equipo de software



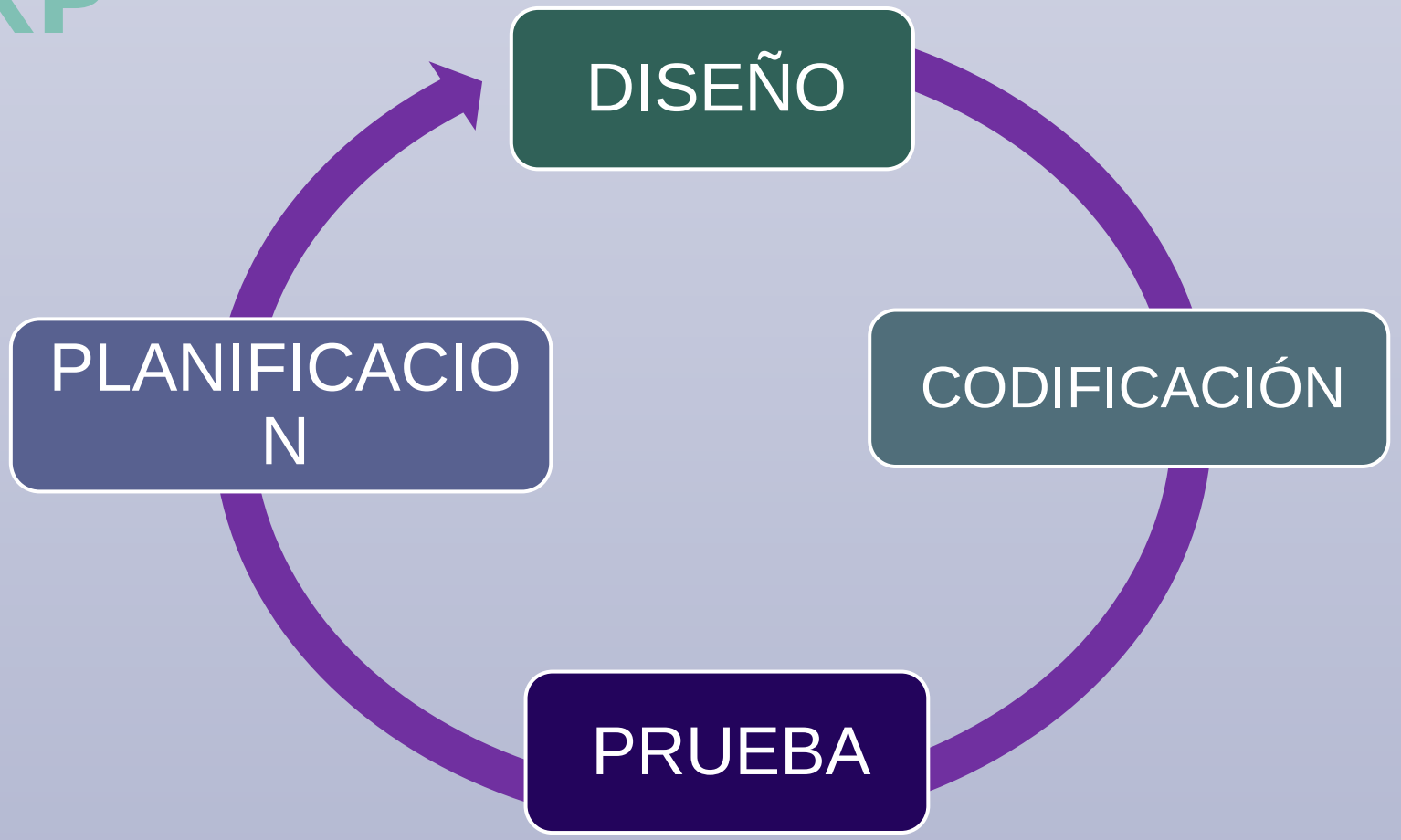
VALENTIA

Un termino mas apropiado seria disciplina. Por ejemplo, es frecuente que haya mucha presión para diseñar requerimientos futuros

RESPETO

Entre sus miembros entre otros participantes y los integrantes del equipo, e indirectamente para el software en si mismo.

PROCESO DE DESARROLLO XP



PLANEACIÓN

- Tambien llamada juego de planeación comienza escuchando
- Creacion de historias de usuario
- Historia es escrita por el cliente y se ubica en una tarjeta indizada con un valor.
- El equipo evalua las historias y le asignan un costo (tiempo).
- Historias muy costosas deben ser descompuestas.
- Entre usuarios y el equipo agrupan las historias de la siguiente entrega.
- Luego de la primera entrega se puede calcular la velocidad del proyecto

DISEÑO

- Sigue rigurosamente el principio MS
- Un diseño sencillo por sobre una representación mas compleja
- Guia laimplementacion de una historia no es el diseño funcional
- Trajetas CRC
- Historias de usuario con Diseños difíciles los prototipos
- Rediseño proceso mediante el cual se cambia un sistema de software en formatal que no altere el comportamiento externo del código, pero si mejore la estructura interna.

CODIFICACIÓN

- No se inicia directo con las codificación sino a pruebas unitarias
- Programación orientada a pruebas
- La prueba garantiza retroalimentación inmediata

PRUEBAS

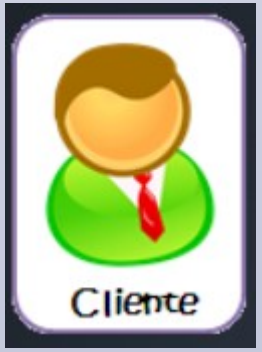
- Creación de Pruebas unitarias antes de que comience la codificación
- Pruebas de estructura que permita automatización7A medida que se organizan las pruebas unitarias individuales es un grupo de pruebas universal las pruebas de la integración y validación del sistema pueden ejecutarse a diario
- Las pruebas de aceptación XP, también llamadas pruebas del cliente y se centran en las características y funcionalidad general del sistemas que son visibles y revisables por parte del cliente

CARACTERISTICAS XP

- Desarrollo iterativo e incremental.
- Programación en parejas.
- Pruebas unitarias continuas.
- Corrección periódica de errores.
- Integración del equipo de programación con el cliente.
- Simplicidad, propiedad del código compartida y refactorización del código.

ROLES


Programador


Cliente


Encargado
de
Pruebas


Tracker


Entrenador


Consultor

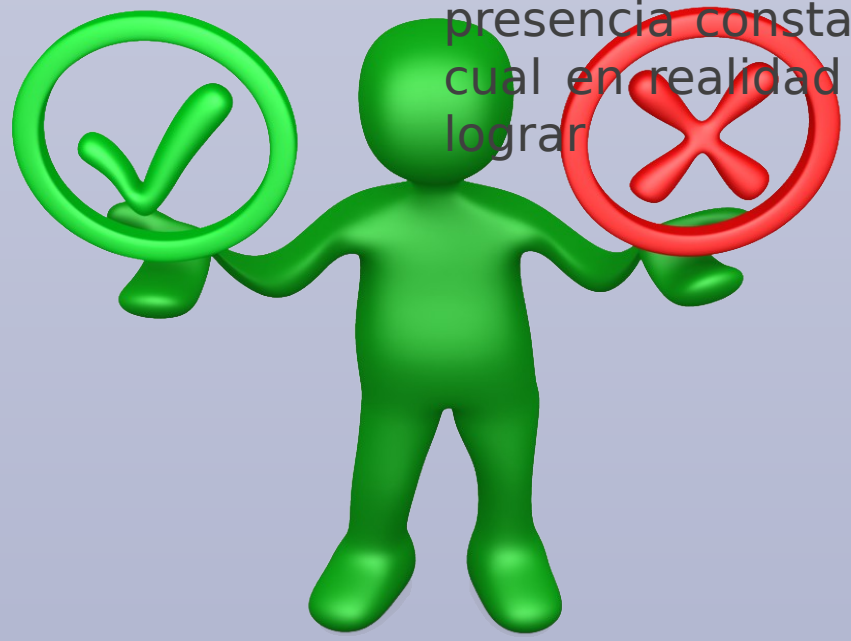

Jefe del
Proyecto

VENTAJAS

Se adapta al desarrollo de sistemas pequeños y grandes; optimiza el tiempo de desarrollo; permite realizar el desarrollo del sistema en parejas para complementar los conocimientos el código es sencillo y entendible

DESVENTAJAS

Las desventajas son que no se tiene la definición del costo y el tiempo de desarrollo; el sistema va creciendo después de cada entrega al cliente y nadie puede decir que el cliente no querrá una función mas; se necesita de la presencia constante de usuario, lo cual en realidad en muy difícil de lograr



CONCLUCIÓN

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo

GRACIAS POR SU ATENCIÓN!