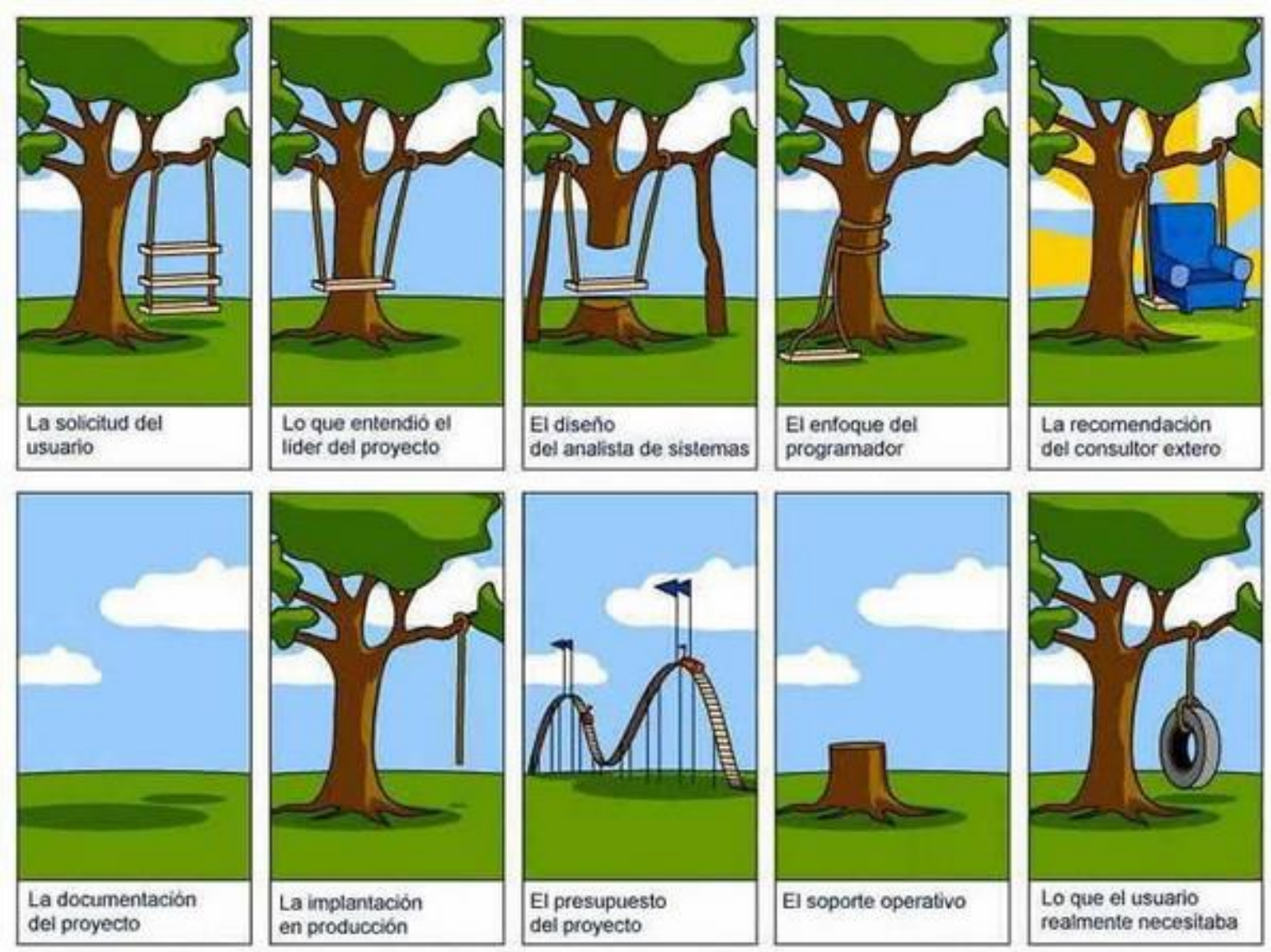
GMD

ISTP SANTIAGO ANTUNEZ DE MAYOLO

Ing. Moisés Álvarez Huamán

Esta situación resulta conocida……???

programación extrema



programación extrema

Programación Extrema

Antecedentes e Historia de Programación extrema

Antecedentes e Historia de Programación extrema

En 1989, Cunningham formó un equipo que usaba los principios y muchas de las prácticas que después adoptaría XP, mientras trabajaba para la compañía "Wyatt Software" [Fowler 2000].

Sin embargo, se reconoce a Kent Beck como el que articuló esta propuesta y le dio nombre propio.

Kent Beck

¿Qué es XP?

¿Que es XP?

- La programación extrema es una metodología de desarrollo ligera basada en una serie de valores y una docena de prácticas que propician un aumento en la productividad a la hora de generar software.
- XP permite controlar los problemas de riesgo en los proyectos.
- XP permite la participación de pequeños grupos de programadores.
- XP requiere un variado equipo de desarrollo.
- XP permite la capacidad de hacer pruebas
- La meta real de XP es entregar el software requerido a tiempo.

Características de XP

- Las características generales de XP es deliberadamente una metodología "liviana" que pasa por alto la utilización de elaborados casos de uso, la exhaustiva definición de requerimientos y la producción de una extensa documentación.
- Todo lo anterior puede parecer caótico según el enfoque tradicional de la ingeniería de software, aunque no hay que olvidar que XP tiene asociado un ciclo de vida y es considerado a su vez un proceso.
- La tendencia de entregar software en lapsos cada vez menores de tiempo y con exigencias de costos reducidos y altos estándares de calidad, hace que XP sea una opción a considerar.

Características de la metodología XP

Enfoque Tradicional vs. XP

eXtreme Programming explained
EMBRACE CHANGE

PX programación extrema

Principios, roles y prácticas de Programación extrema

Principios de la Programación extrema

Se busca :

1. Realimentación rápida
2. Asumir la simplicidad
3. Cambio incremental
4. Aceptar el cambio
5. Hacer trabajo de calidad.

Principios de la Programación extrema

Las Prácticas son:

1. El juego de la planificación
2. Pequeñas entregas
3. Metáfora
4. Diseño simple
5. Pruebas
6. Refactorización

Principios de la Programación extrema

7. Programación por parejas
8. Propiedad colectiva
9. Integración continua
10. 40 horas semanales
11. Cliente en casa
12. Estándares de codificación.

Objetivos de la Programación extrema

Objetivos de XP

Son:

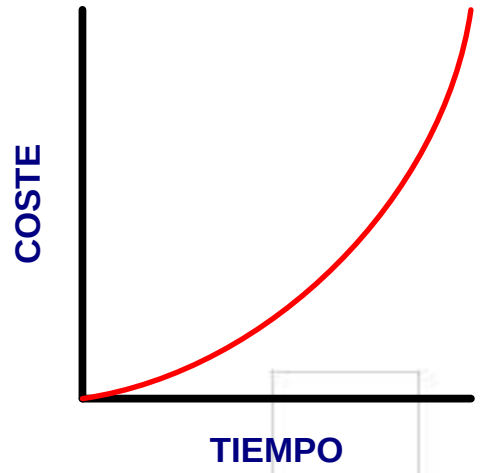
1. La satisfacción del cliente.
2. Potenciar el trabajo en grupo, todos están involucrados en el desarrollo del software.

Interacción entre Las cuatro variables de Gestión de proyecto

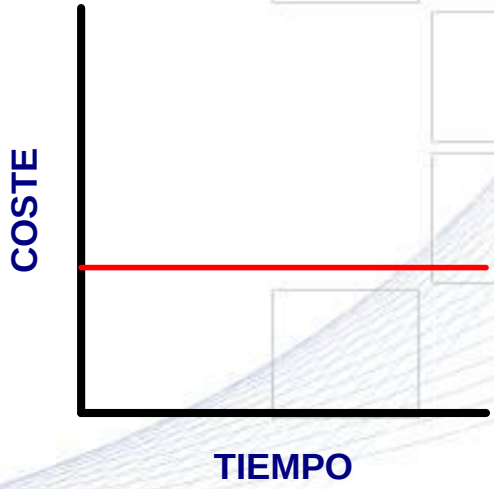
Variable	Si aumenta en exceso...	Si se reduce...
Alcance		Permite mejorar la calidad, siempre que resuelva el problema básico del cliente. Tambien permite reducir plazo y coste. La herramienta más potente de gestión (*)
Tiempo	Más puede mejorar calidad y alcance, pero en exceso puede dañar, pues la mejor realimentación viene del sistema en producción.	Si poco, sufrirá la calidad e inmediatamente detrás el alcance, el tiempo y el coste.
Coste	Más dinero puede engrasar el sistema, pero en exceso puede crear más problemas que los que resuelve.	Con poco dinero será imposible resolver los problemas del cliente.
Calidad	Insistir en mayor calidad permite conseguir plazos menores o hacer más en un tiempo dado. Efecto humano: se trabaja mejor si se siente que se hace un buen trabajo.	Variable terrible de control. Se puede sacrificar para obtener ganancias a corto, pero los costes posteriores son enormes (humanos, de negocio y técnicos).

El coste de Cambio

- El coste de los cambios crece con el tiempo.



- XP propone que los costes de los cambios no tienen por que aumentar con el tiempo.



Las cuatro valores

programación extrema

Valores para desarrollar software:

1. Comunicación
2. Sencillez
3. Retroalimentación
4. Valentía.

Roles de XP

Cliente

- Escribe "Historias de Usuario" y especifica Pruebas Funcionales.
- Establece prioridades, explica las Historias
- Puede ser o no un usuario final
- Tiene autoridad para decidir cuestiones relativas a las Historias.

Programador

- Hace estimaciones sobre las Historias
- Define Tareas a partir de las Historias y hace estimaciones
- Implementa las Historias y las Pruebas Unitarias

Roles de XP

Tutor

- Observa todo, identifica señales de peligro, se asegura que el proyecto se mantiene en curso
- Ayuda en todo
- Da avisos cuando se necesita.

Perseguidor (calidad)

- Monitoriza el progreso de los programadores, toma acción si las cosas tienden a salirse de su senda.
- Las acciones incluyen reuniones con el Cliente, solicitar ayuda al Tutor u otro Programador.

Roles de XP

Verificador

- Implementa y corre las Pruebas Funcionales (no Pruebas Unitarias)
- Presenta gráficas de los resultados y se asegura de que la gente conoce cuándo los resultados empiezan a decaer.

"Agorero"

- Se asegura que todos conocen los riesgos que existen
- Se asegura que las malas noticias no se ocultan, se disculpan o se reducen de proporción.

Roles de XP

Gestor

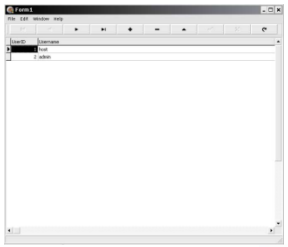
- Planifica las reuniones (por ej., plan de iteraciones, plan de lanzamientos releases), se asegura que el proceso de las reuniones se sigue, anota los resultados de la reunión para futuros informes y los pasa al Perseguidor.
- Posiblemente responsable ante el "Propietario de Oro"
- Asiste a las reuniones, aporta información útil anterior.

"Propietario de Oro"

- La persona que paga el proyecto, que puede ser o no la misma que el Cliente.

Las cuatro actividades básicas

1. Codificar
2. Hacer pruebas
3. Escuchar
4. Diseñar.



Proceso de Desarrollo

Artefactos esenciales en XP

programación extrema

+ Historias del Usuario
+ Tareas de Ingeniería
+ Pruebas de Aceptación

+ Pruebas Unitarias y de Integración
+ Plan de la Entrega
+ Código

Historia de Usuario

Historia de Usuario	
Número: **1**	Nombre: **Enviar artículo**
Usuario: **Autor**	
Modificación de Historia Número:	Iteración Asignada: 2
Prioridad en Negocio: **Alta** (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción: **Se introducen los datos del artículo (título, fichero adjunto, resumen, tópicos) y de los autores (nombre, e-mail, afiliación). Uno de los autores debe indicarse como autor de contacto. El sistema confirma la correcta recepción del artículo enviando un e-mail al autor de contacto con un userid y password para que el autor pueda posteriormente acceder al artículo.**	
Observaciones:	

Tarea de Ingeniería

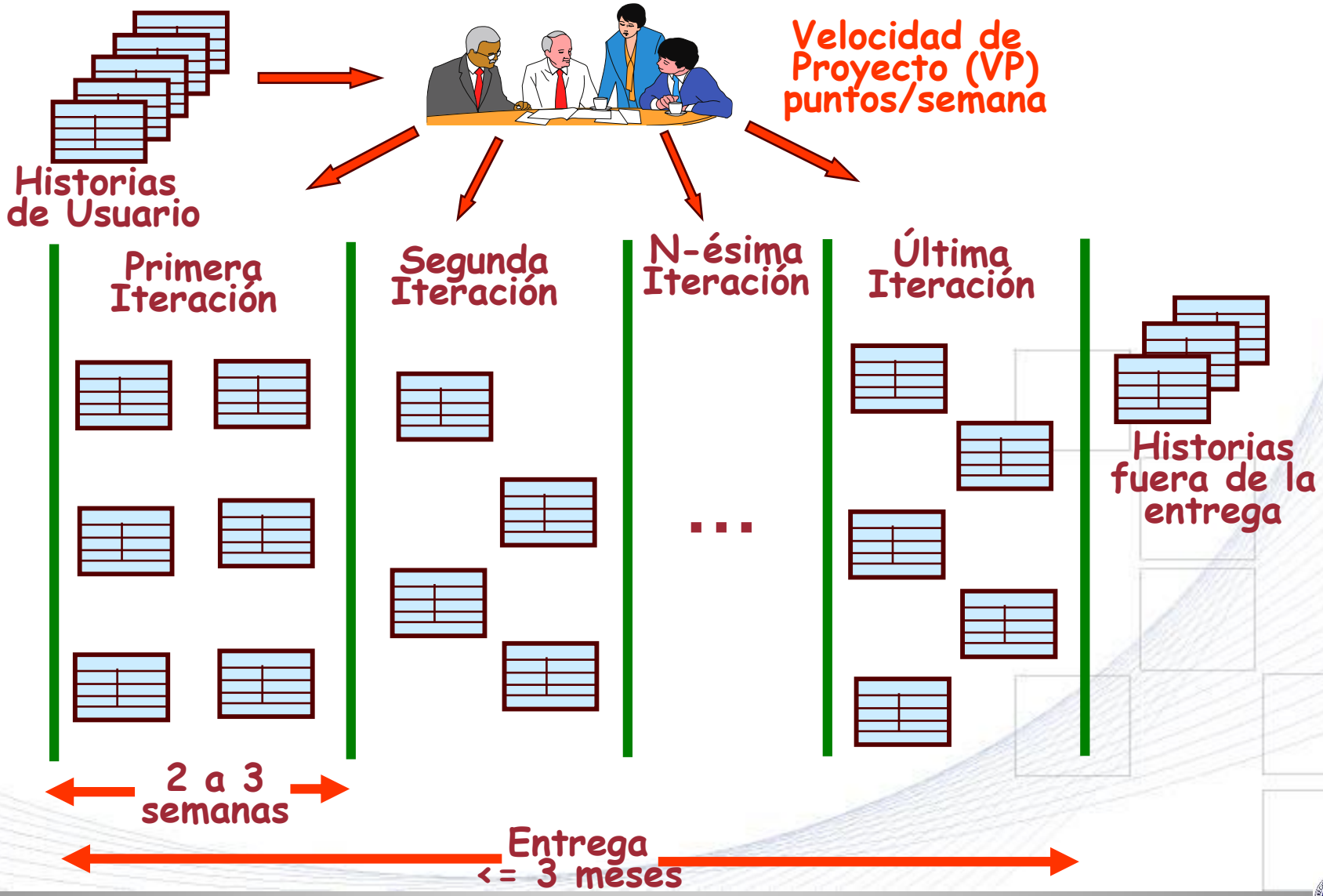
Tarea de Ingeniería	
Número tarea:	Número historia:
Nombre tarea:	
Tipo de tarea : Desarrollo / Corrección / Mejora / Otra	Puntos estimados:
Fecha inicio:	Fecha fin:
Programador responsable:	
Descripción:	

Prueba de Aceptación

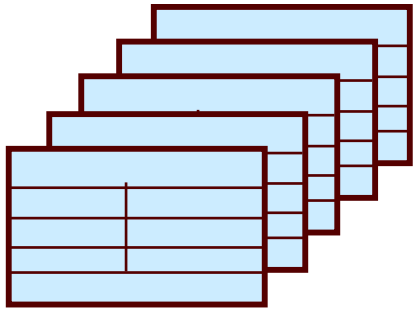
Caso de Prueba	
Número Caso de Prueba:	Número Historia de Usuario:
Nombre Caso de Prueba:	
Descripción:	
Condiciones de ejecución:	
Entradas:	
Resultado esperado:	
Evaluación:	

Escenarios en XP : Exploración

Escenarios en XP: Planificación de la Entrega

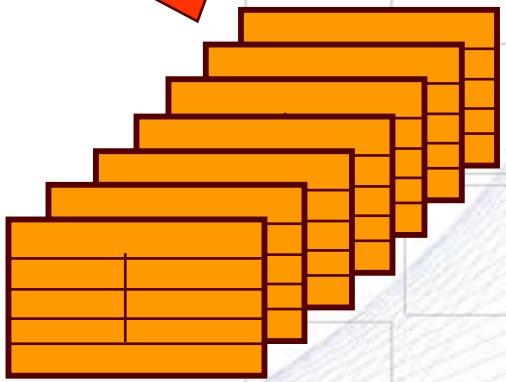


Escenarios en XP : Comenzar Iteración



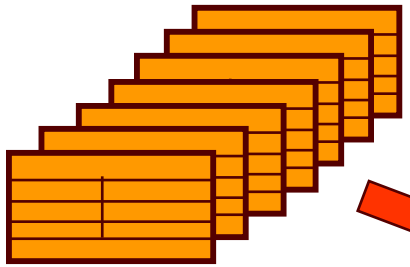
Historias de la Iteración

Definir y ordenar Tareas de Ingeniería

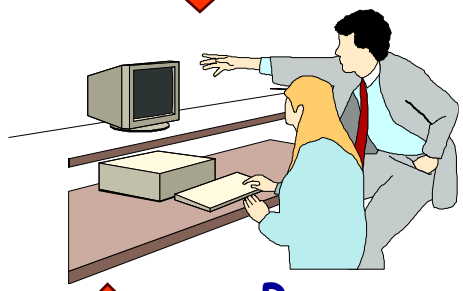


Tareas de la iteración

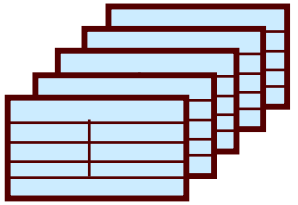
Escenarios en XP : Programación



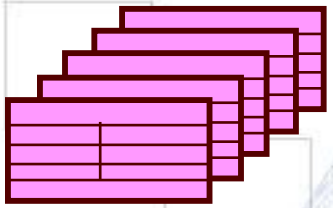
Tareas de Historias de la iteración



Programación en Parejas

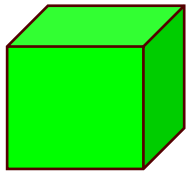


Historias de la Iteración



Pruebas de Aceptación de Historias de la iteración

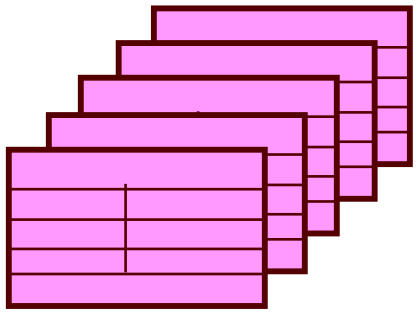
Diseño
Refactoring
Programación
Pruebas Unitarias
Integración
Pruebas de Integración
Pruebas de Aceptación



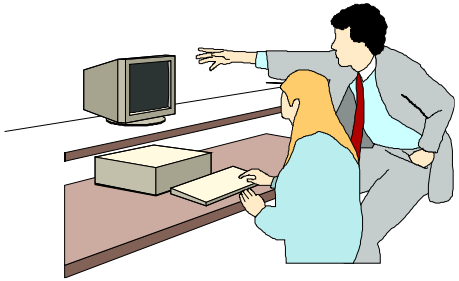
Versión del Producto

Escenarios en XP : Pruebas de Aceptación

Definir Pruebas de Aceptación



Pruebas de Aceptación



Aplicar Pruebas de Aceptación

Corregir errores
Definir nuevas Historias

Entorno y clima de trabajo
Espacio de trabajo XP

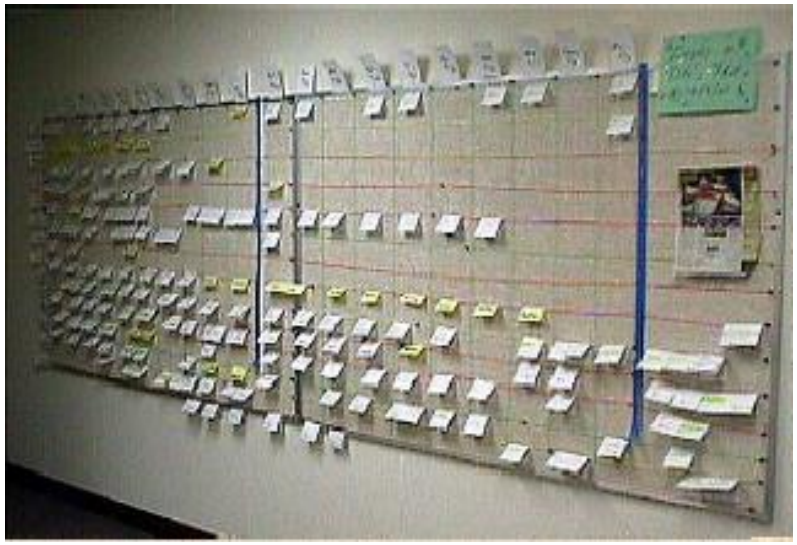
- Espacio abierto
- Mesas centrales
- Cubículos en el espacio exterior

Espacio de trabajo del proyecto C3 de DaimlerChrysler

… Entorno y clima de trabajo
Reunión diaria XP

- Reunión diaria: "Stand-up Meeting"
  - Todo el equipo
    - Problemas
    - Soluciones
  - De pie en un círculo
    - Evitar discusiones largas
    - Sin conversaciones separadas

… Entorno y clima de trabajo
Gantt de Pared



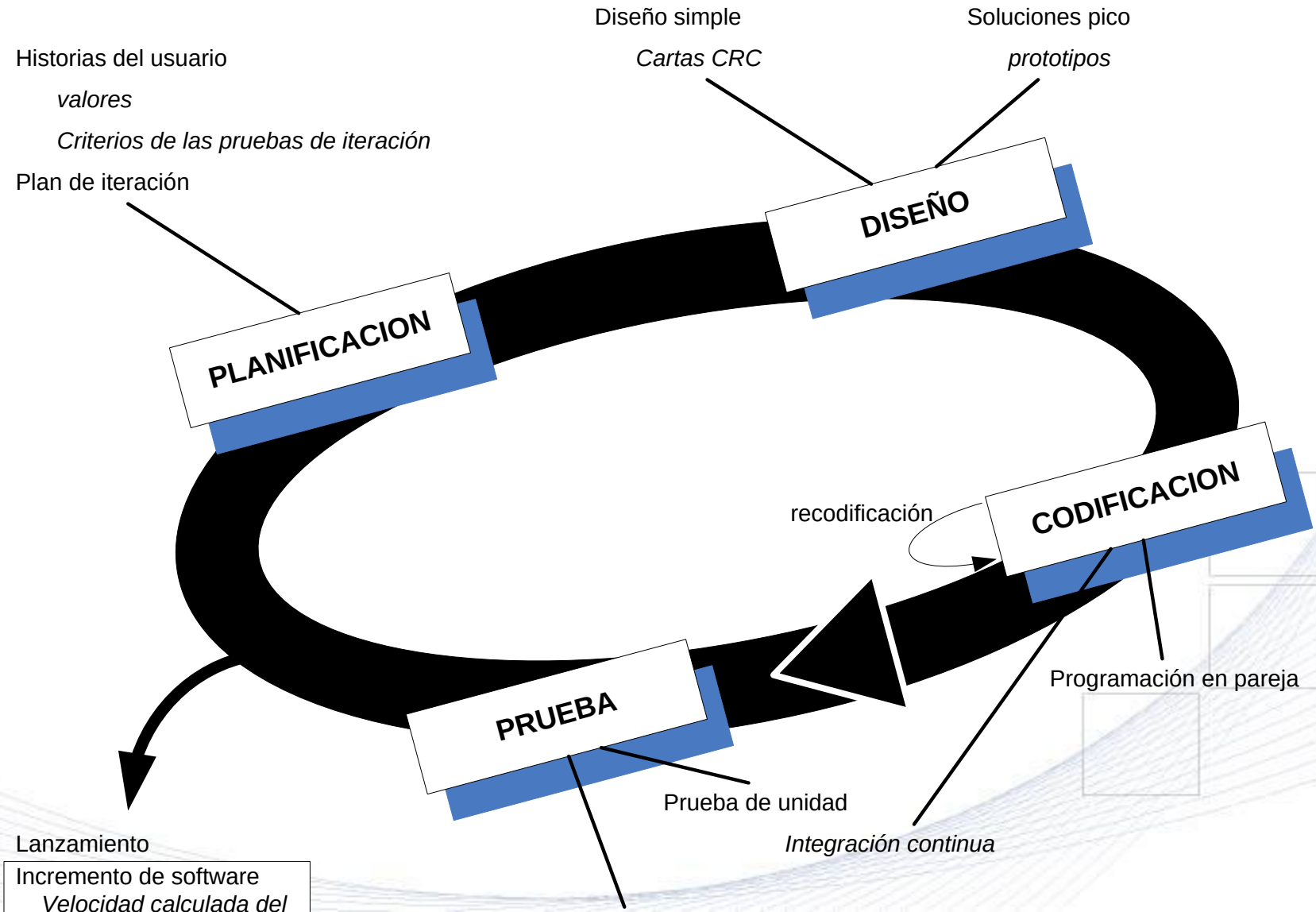
"Centro del universo del proyecto"

"Punto de reunión para la "Stand-up Meeting"

Obtenida de www.agiletek.com

Fases de la metodología XP

Como hacemos funcionar la Metodología XP



Planificación

XP plantea la planificación como un permanente dialogo entre las partes la empresarial (deseable) y la técnica (posible).

deseable

posible

… Planificación

.1 El Juego de la Planificación

- Negocio
  - Ámbito *¿Qué debe resolver el software?*
  - Prioridad *¿Qué debe ser echo en primer lugar?*
  - Composición de versiones *¿Cuánto es necesario hacer para aportar valor?*
  - Fechas de versiones *¿Fechas para presencia del software?*

… Planificación

.1 El Juego de la Planificación

- Técnico.
  - Estimaciones *¿Cuánto lleva implementar una característica?*
  - Consecuencias, *informar sobre consecuencias de las decisiones que adopta el negocio.*
  - Procesos *¿Cómo se organiza el trabajo en el equipo?*
  - Programación detallada: En una versión *¿Qué se resolverá primero?*

… Planificación

programación extrema

.2 Pequeñas versiones.

Cada versión debe de ser tan pequeña como fuera posible, conteniendo los requisitos de negocios más importantes, las versiones tiene que tener sentido como un todo.

.3 Metáfora.

Es una historia que todo el mundo puede contar a cerca de cómo funciona el sistema.

Diseño

.4 Diseño simple.

El diseño adecuado par el software es aquel que:

1. Funciona con todas las pruebas.
2. No tiene lógica duplicada.
3. Manifiesta cada intención importante para los programadores
4. Tiene el menor número de clases y métodos.

Codificación

.5 Recodificación.

Este proceso se le denomina recodificar o refactorizar (refactoring).y consiste en hacer el programa mas simple sin perder funcionalidad.

No debemos de recodificar ante especulaciones si no solo cuándo el sistema te lo pida.

… Codificación

.6 Programación por parejas.

Todo el código de producción se escribe con dos personas mirando a una máquina, con un solo teclado y un solo ratón.

Cada miembro de la pareja juega su papel: uno codifica en el ordenador y piensa la mejor manera de hacerlo, el otro piensa mas estratégicamente, ¿Va a funcionar?, ¿Puede haber pruebas donde no funcione?, ¿Hay forma de simplificar el sistema global para que el problema desaparezca?.

… Codificación

.7 Propiedad Colectiva.

Cualquiera que crea que puede aportar valor al código en cualquier parcela puede hacerlo, ningún miembro del equipo es propietario del código.

.8 Integración continúa.

El código se debe integrar como mínimo una vez al día, y realizar las pruebas sobre la totalidad del sistema.

… Codificación

.9 Cuarenta horas.

Si queremos estar frescos y motivados cada mañana y cansado y satisfecho cada noche. del sistema.

.10 Cliente In Situ.

Un cliente real debe sentarse con el equipo de programadores, estar disponible para responder a sus preguntas, resolver discusiones y fijar las prioridades.

… Codificación

.11 Estándares de Codificación.

Se debe establecer un estándar de codificación aceptado e implantado por todo el equipo.

Pruebas

.12 Hacer pruebas.

Toda característica en el programa debe ser probada, los programadores escriben pruebas para chequear el correcto funcionamiento del programa, los clientes realizan pruebas funcionales. El resultado un programa mas seguro que soporte cambios en el tiempo.

Prácticas XP

PLANIFICACION

1. El juego de la planificación
2. Entregas pequeñas
3. Metáfora

DISEÑO

4. Diseño simple

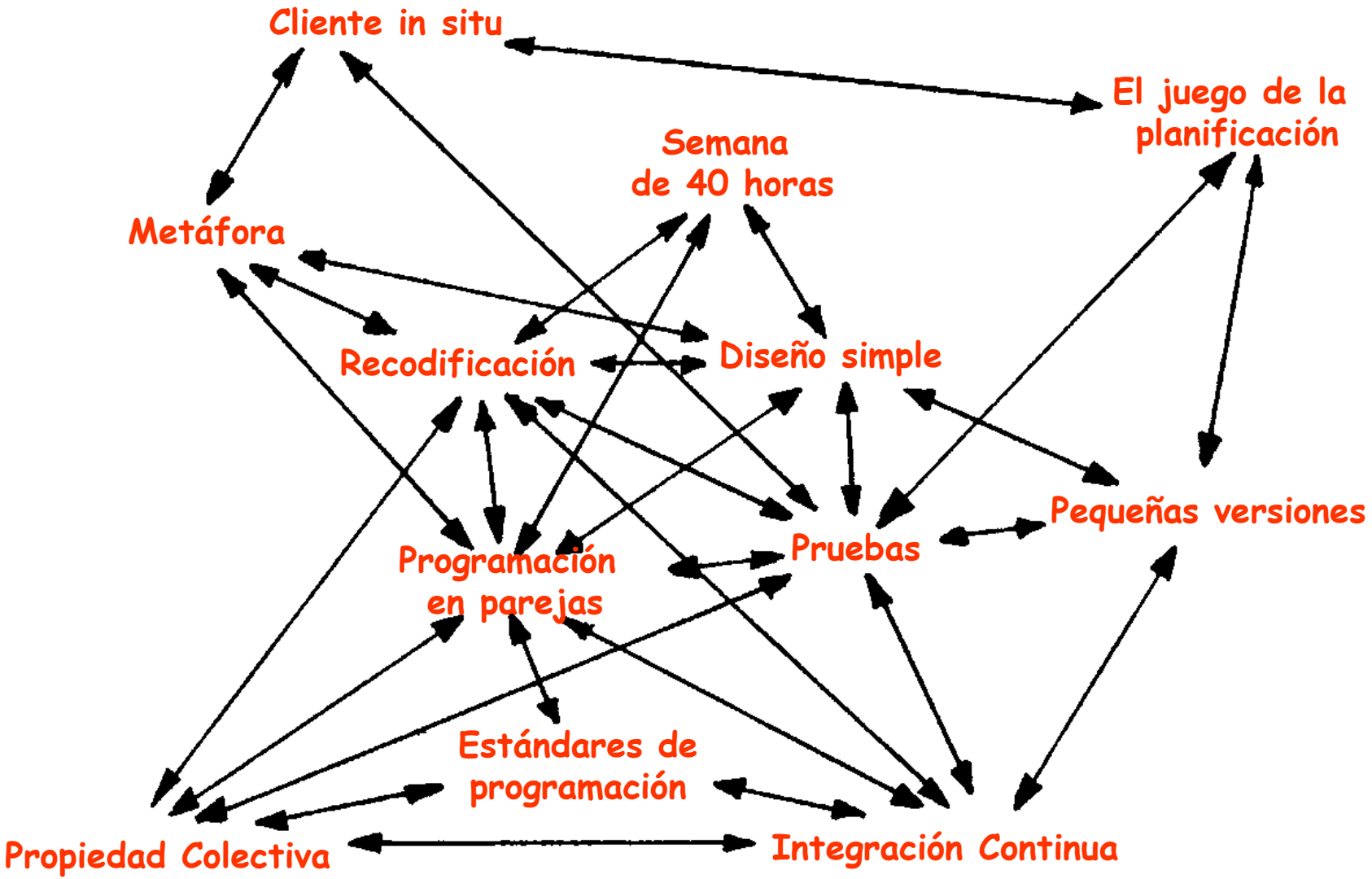
CODIFICACION

5. Recodificación
6. Programación en parejas
7. Propiedad colectiva
8. Integración continua
9. Semana de 40 horas
10. Cliente in situ
11. Estándares de programación

PRUEBAS

12. Pruebas

… Prácticas XP
Interacción entre Prácticas



XP: Kent Beck

Aspectos sobre Programación Extrema

Aspectos Positivos De Xp

- Pruebas unitarias en el código factor clave para obtener un software de alta calidad.
- La integración continua es aceptada y recomendada para evitar catástrofes ocasionadas por defectos no detectados a tiempo.
- la simplicidad y la refabricación es encontrado como un factor saludable en la práctica de programación.
- El enfoque "extremadamente humano", siendo este un aspecto que el resto del campo del software debería tratar de emular.
- Cliente también se percibe el enfoque humano, ya que tenemos su presencia constante en las instalaciones del desarrollador.

Aspectos Controversiales de Xp

- La XP se ha afirmado que no es la metodología que va a resolver todos los problemas en IS y se han resaltado sus limitaciones.
- No es aconsejable XP si no es posible disminuir la curva costo/tiempo.
- Tampoco si la tecnología o el entorno no permiten realizar integraciones frecuentes o realizar pruebas continuamente.
- No se recomienda intentar XP si la distribución física impide la programación en pares o si no todos los programadores se encuentran en el mismo sitio.

Aspectos Controversiales de Xp

- Desalienta el diseño, que es débil en la documentación, que el modelo no aplica para proyectos donde la seguridad es crítica.
- Exceso de pruebas retrasa el desarrollo, el diseño simple solo aplica a proyectos simples, que la programación en pares consume mayor tiempo y recursos.
- XP asume implícitamente que siempre se utiliza el enfoque de programación orientada a objetos.

BENEFICIOS

+ Satisfacción del cliente.
+ Cumplimiento de plazos.
+ El cliente tiene el control sobre las prioridades.
+ Se hacen pruebas continuas durante el proyecto.
+ Calidad en el trabajo.
+ La XP es mejor utilizada en la implementación de nuevas tecnologías donde los requerimientos cambian rápidamente.

programación extrema

GRACIAS