

Programación Extrema

Ing. Sebastian Priolo

Metodologías Ágiles

- Menos orientadas a los documentos.
- Orientadas al código.
- El cambio es bienvenido.
- Procesos que cambian
- NO son predictivos
- Son adaptables

Ejemplos

- Programación Extrema
- Scrum
- Crystal
- Evolutionary Project Management (Evo)
- Feature Driven Development (FDD)
- Adaptive Software Development (ASD)
- Lean Development (LD)
- Lean Software Development (LSD)

Manifiesto Ágil

Manifiesto Ágil

En marzo de 2001, 17 críticos de los modelos de mejora basados en procesos, convocados por Kent Beck, se reunieron en Salt Lake City para discutir sobre el desarrollo de software.

Se acuñó el término “Métodos Ágiles”.

Kent Beck

Mike Beedle

Arie van Bennekum

Alistair Cockburn

Ward Cunningham

Martin Fowler

James Grenning

Jim Highsmith

Andrew Hunt

Ron Jeffries

Jon Kern

Brian Marick

Robert C. Martin

Steve Mellor

Ken Schwaber

Jeff Sutherland

Dave Thomas

Manifiesto ágil

- Estamos descubriendo mejores maneras de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de esta experiencia hemos aprendido a valorar:

Manifiesto Ágil

- Individuos e interacciones sobre *procesos y herramientas*
- Software que funciona sobre *documentación exhaustiva*
- Colaboración con el cliente sobre *negociación de contratos*
- Responder ante el cambio sobre *seguimiento de un plan*

Manifiesto Ágil

- Esto es, aunque los elementos a la derecha tienen valor, nosotros valoramos por encima de ellos los que están a la izquierda.

Individuos e interacciones sobre *procesos y herramientas*

- Conocimiento
- Adaptar procesos a las personas
- Creatividad e innovación

Software que funciona sobre *documentación exhaustiva*

- Ver y actuar sobre prototipos
- Feedback
- Generar ideas
- Nuevas posibilidades
- Documentos < Comunicación
- Documentos = Barricadas

Colaboración con el cliente sobre *negociación de contratos*

- el cliente es un miembro más del equipo
 - se integra y colabora
 - grupo de trabajo.
-
- Los modelos de contrato por obra no encajan

Responder ante el cambio sobre *seguimiento de un plan*

- Entornos inestables
- factor inherente el cambio y la evolución
- más valiosa la capacidad de respuesta

Valores de gestión Ágil

- Anticipación y la adaptación
- gestión de proyectos ortodoxa: planificación y control para evitar desviaciones sobre el plan.

Principios ágiles

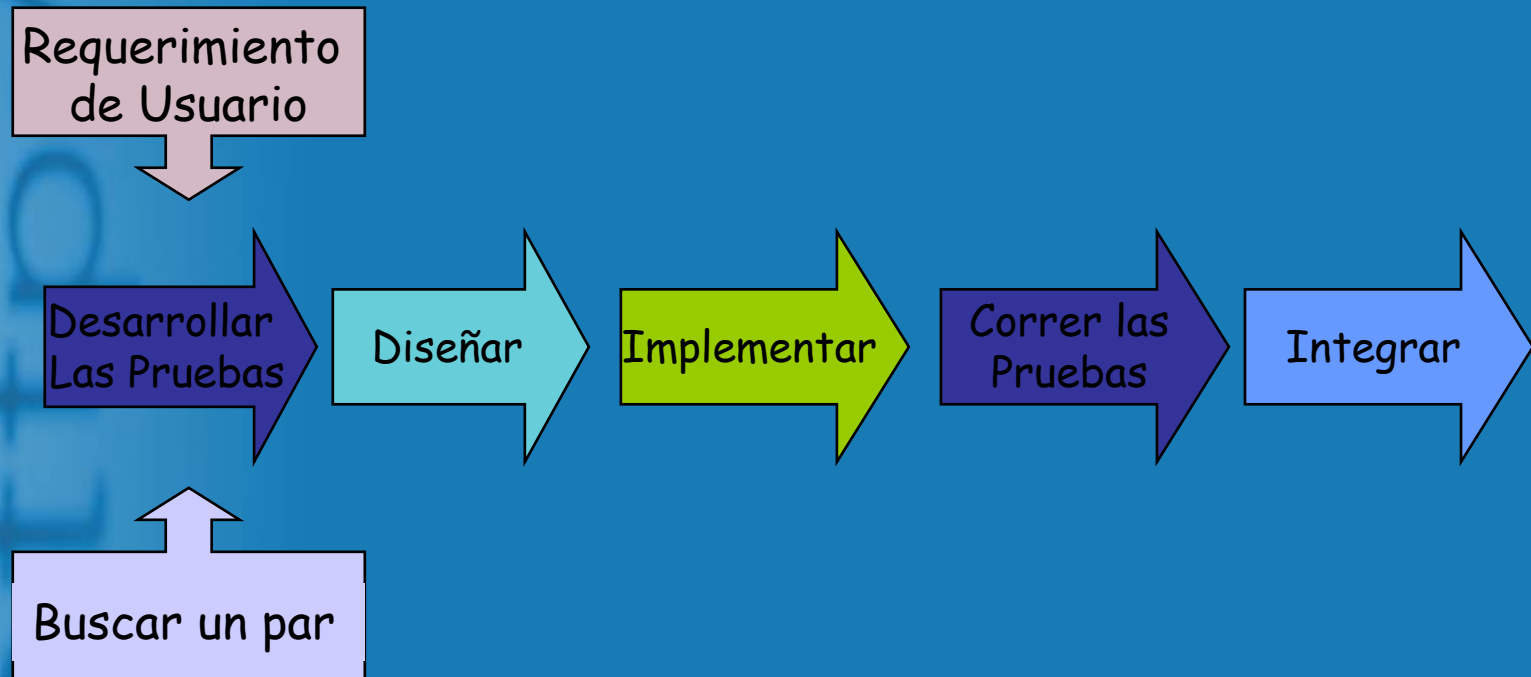
Conclusiones

- En software la construcción es tan barata que es casi gratis.
- En software todo el esfuerzo está en el diseño, de modo que requiere de personas creativas y talentosas.
- Los procesos creativos no se planean fácilmente, de modo que la previsibilidad bien puede ser una meta imposible.
- Debemos ser muy cautos al usar la metáfora de la ingeniería tradicional para construir software. Es un tipo diferente de actividad y por ende requiere un proceso diferente.

Programación Extrema

- Kent Beck, 1999.
- Conjunto de valores, practicas y actividades.
- Presenta distintos escenarios

Un día en un desarrollo XP



Valores XP

- **Comunicación:** Crear software requiere de sistemas comunicados.
- **Simplicidad:** Empezar con lo necesario y requerido y trabajar desde ahí.
- **Retroalimentacion:** Del sistema, del cliente, y del equipo.
- **Valentia:** Programa para hoy y no para mañana.
- **Respeto:** El equipo debe trabajar como uno, sin hacer desiciones repentinas

Practicas XP

- Conjunto de practicas:
 - Retroalimentación a escala fina
 - Proceso continuo en lugar de por lotes
 - Entendimiento compartido
 - Bienestar del programador

Retroalimentación a escala fina

- Desarrollo Guiado por Pruebas
- Juego de planificación
- Cliente presente
- Programación en pares

Proceso contínuo en lugar de por lotes

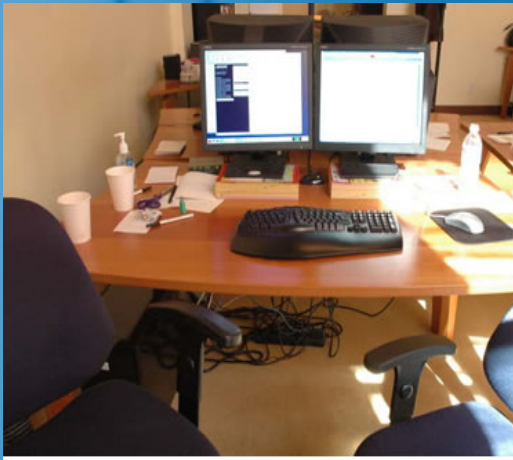
- Integración continua
- Refactorar sin piedad
- Liberación pequeña

Entendimiento compartido

- Diseño simple
- Metáfora del sistema
- Propiedad colectiva del código
- Convenciones del código

Bienestar del programador

- Paso sostenible



Actividades Básicas

- Codificar
- Hacer pruebas
- Escuchar
- Diseñar

Artefactos XP

- 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:	

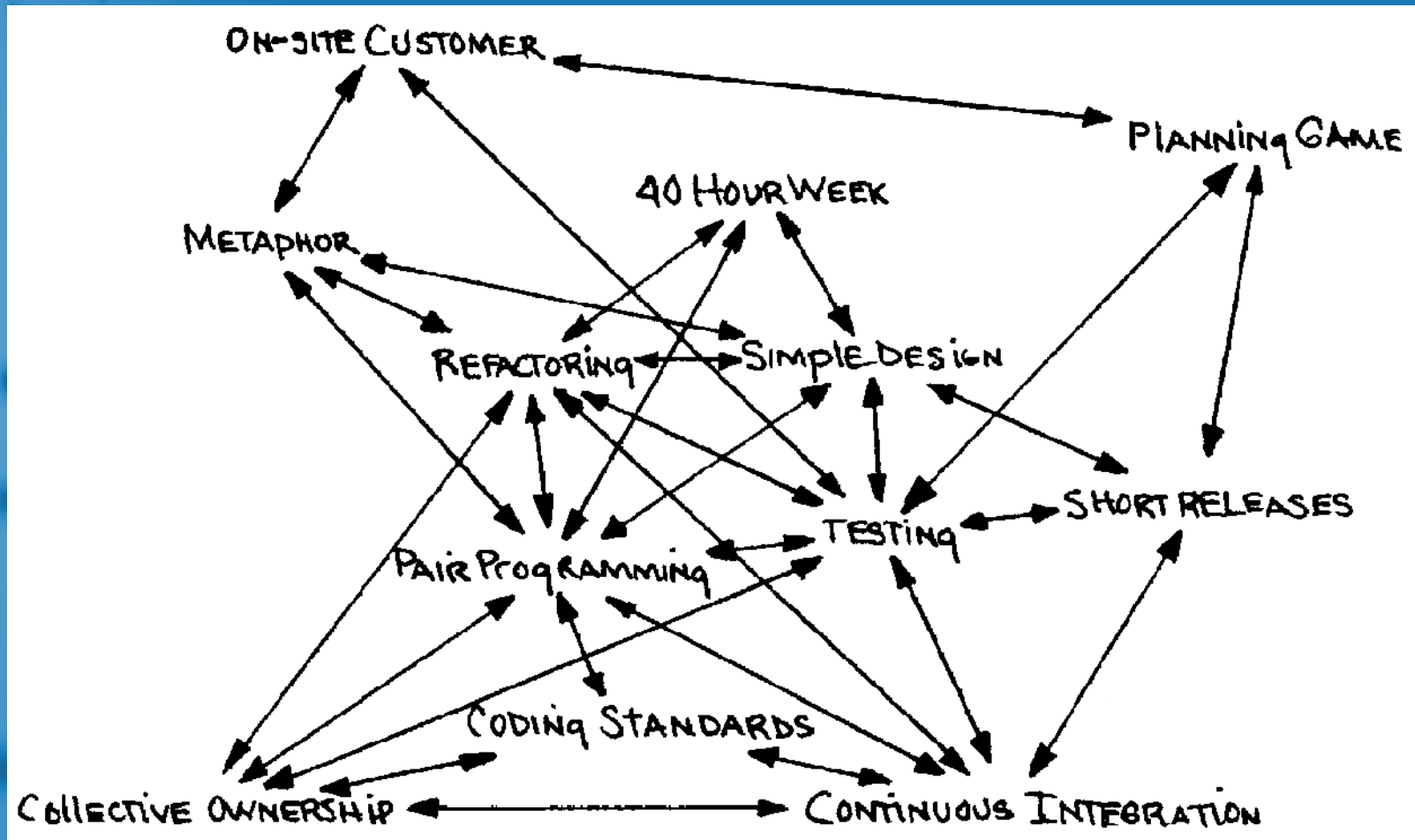
Tarea de Ingeniería

Tarea	
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

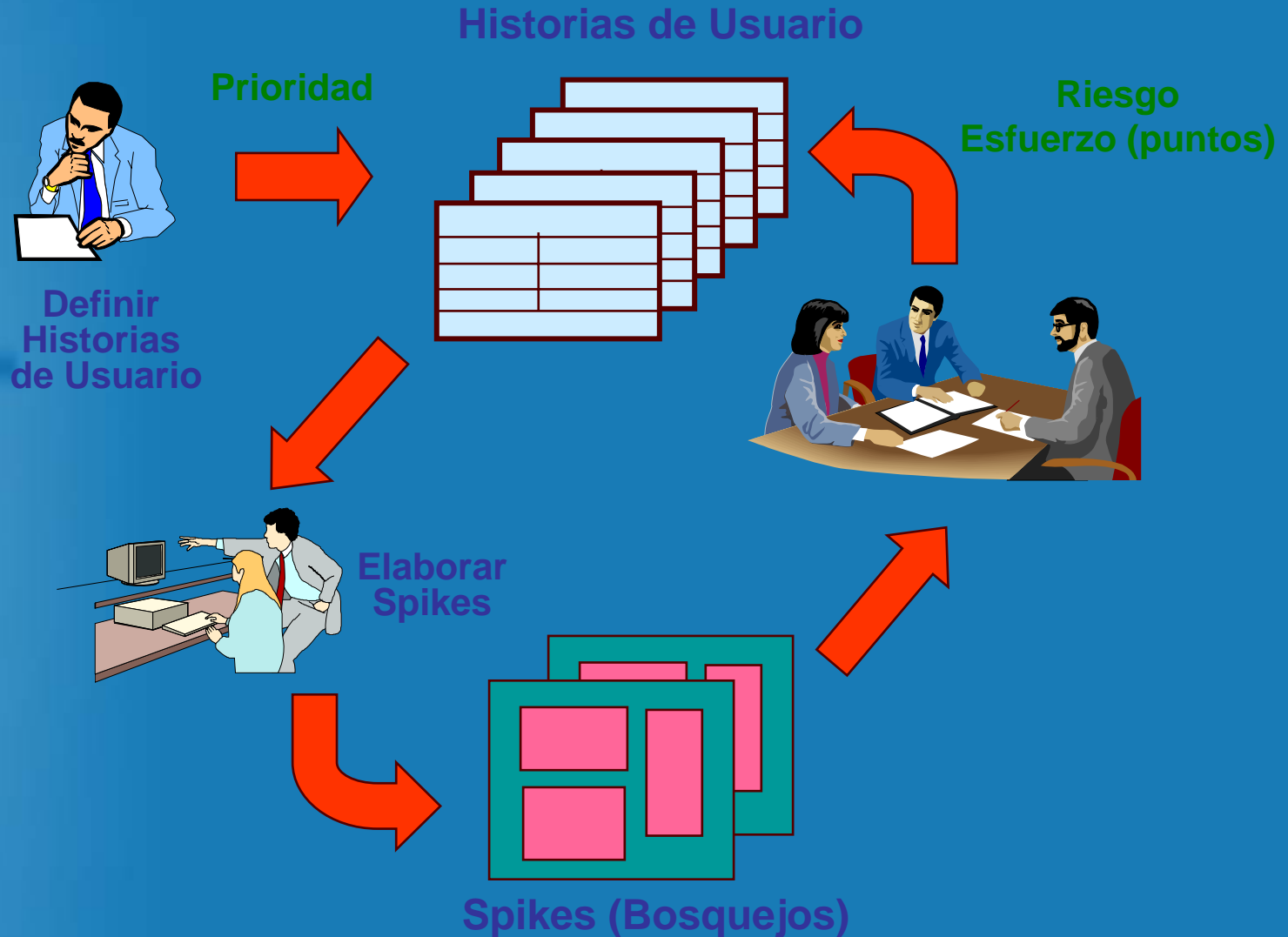
Caso de Prueba	
Número Caso de Prueba:	Número Historia de Usuario:
Nombre Caso de Prueba:	
Descripción:	
Condiciones de ejecución:	
Entradas:	
Resultado esperado:	
Evaluación:	

Relación entre practicas

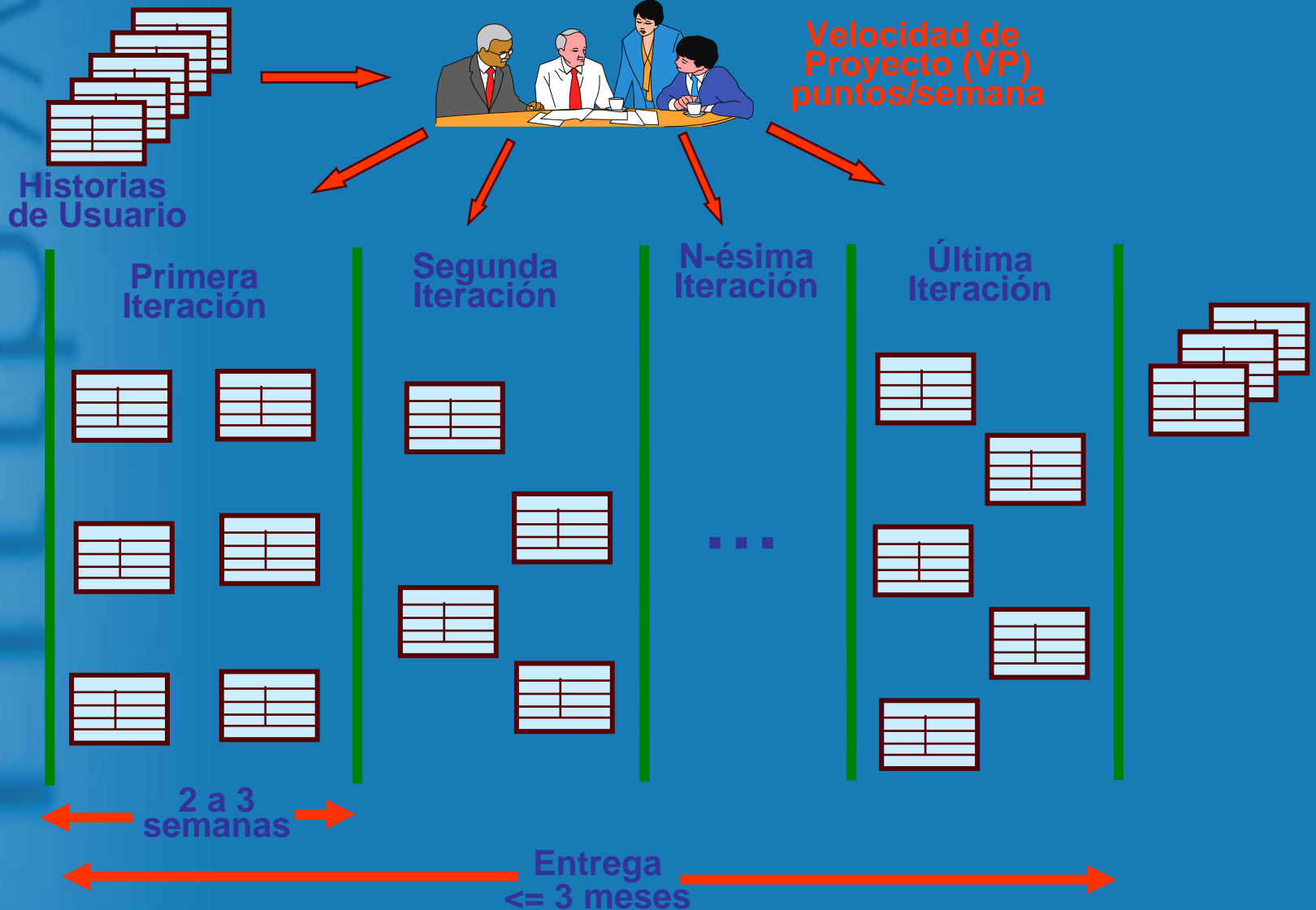


Escenarios en XP

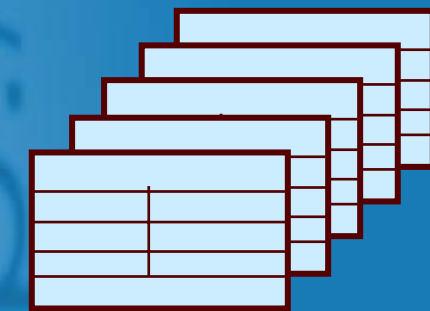
Escenarios en XP : Exploración



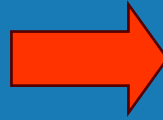
Planificación de la Entrega



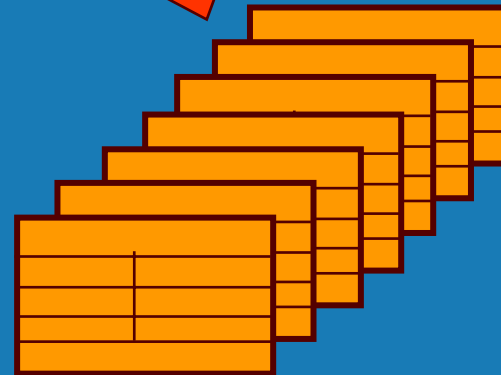
Comenzar Iteración



Historias de la Iteración

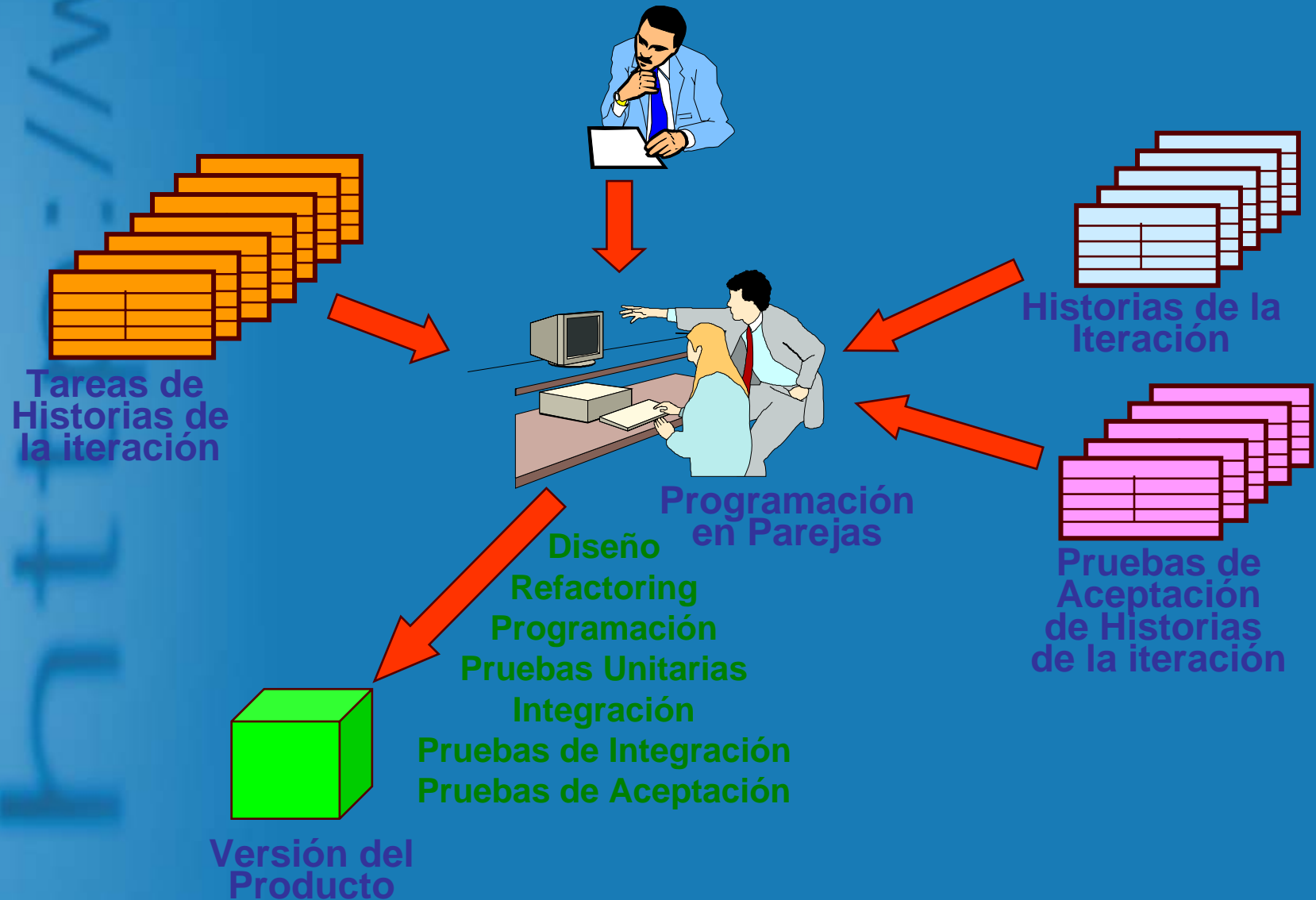


Definir y
ordenar
Tareas de
Ingeniería

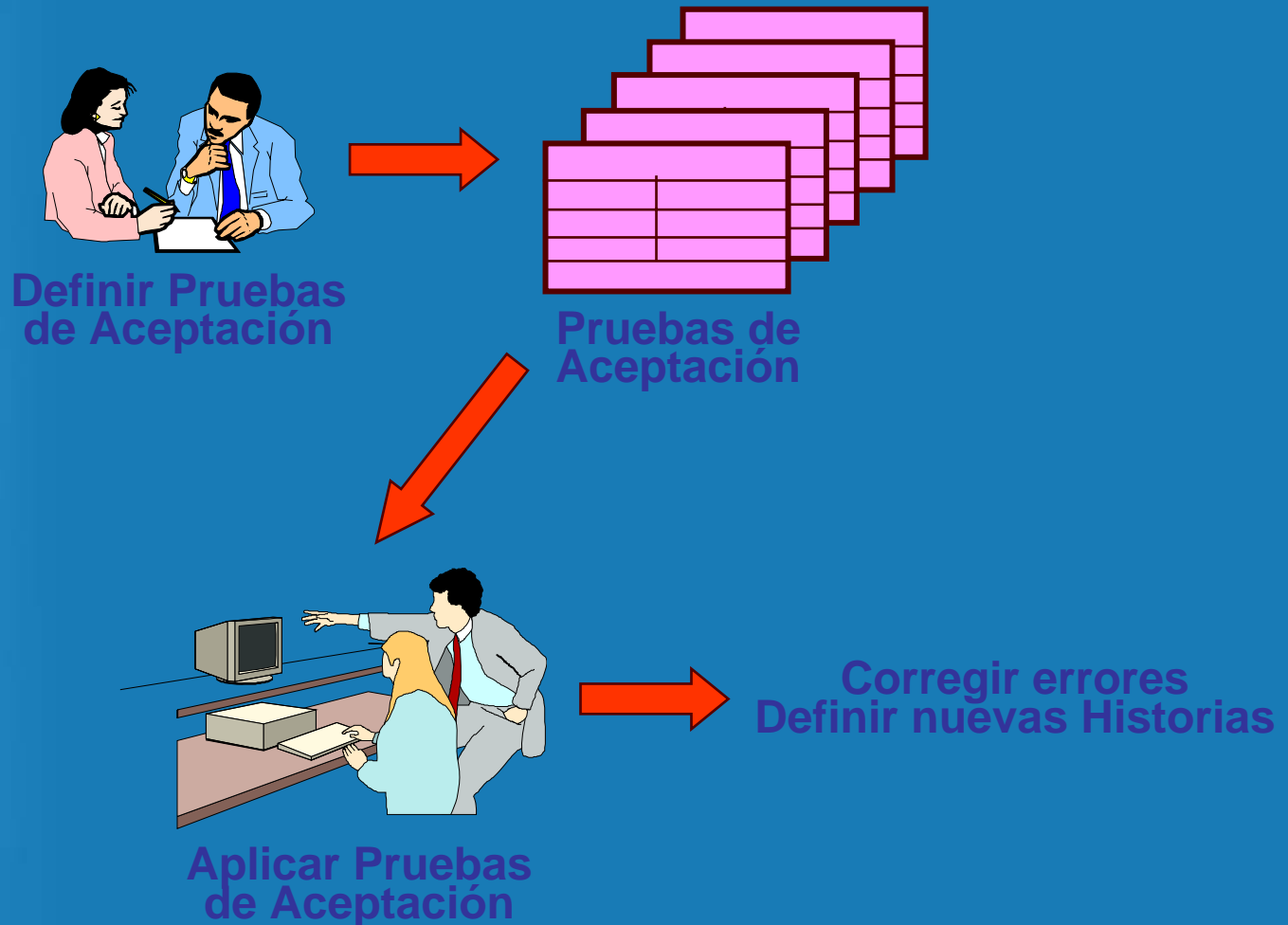


Tareas de
la iteración

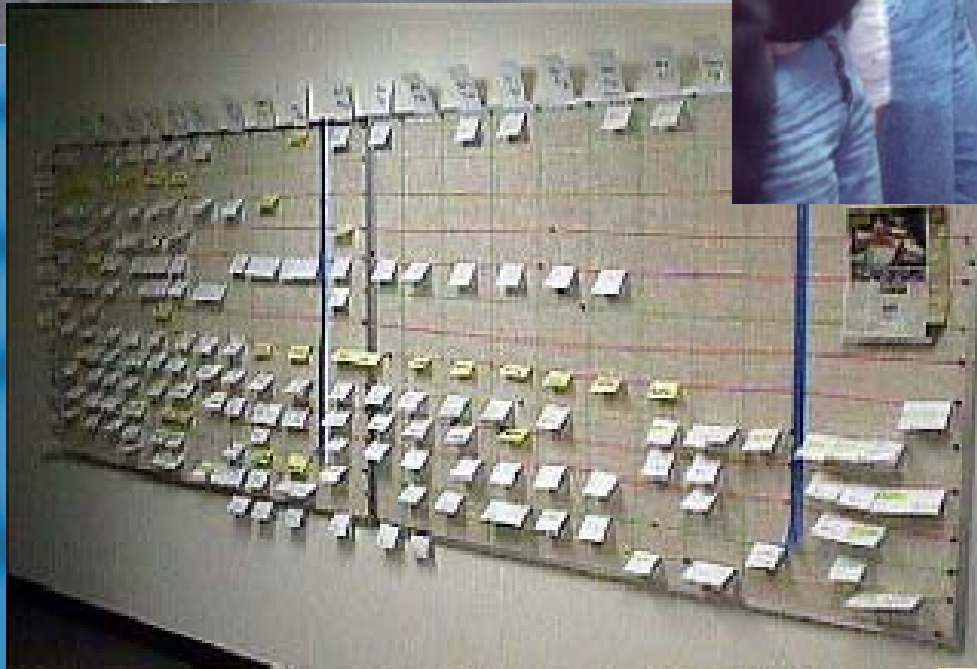
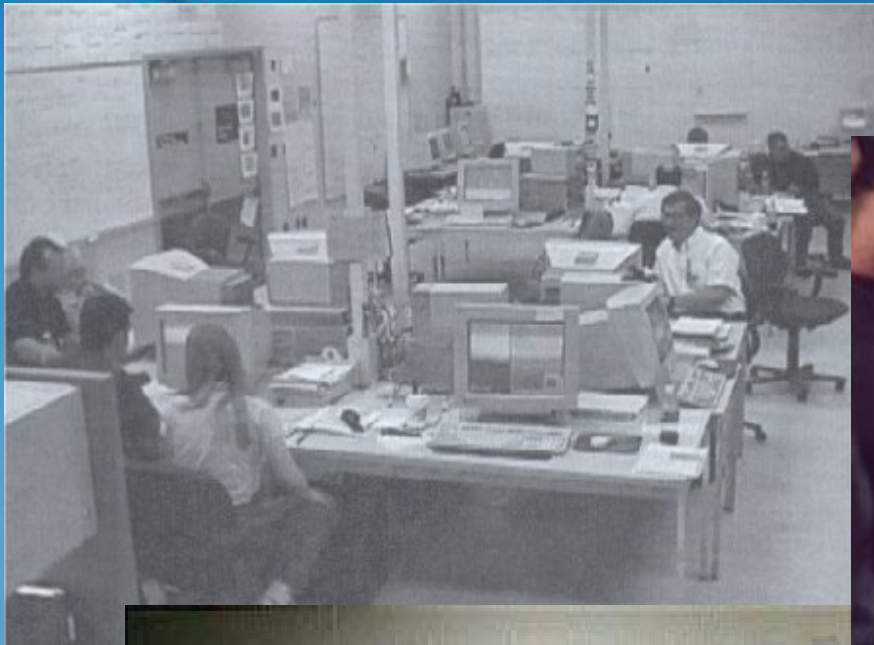
Programación



Pruebas de Aceptación



Entorno y clima de trabajo



Herramientas

Herramientas

- Xplanner
- Fitnessse
- MediaWiki
- Cruise Control
- Mantis

Resumen y conclusiones

Comparando

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

¿Cuándo utilizar una Metodología Ágil?

- **¿Tienes ya un proceso?** No
 - o existe pero no reacciona bien a los cambios
 - o existe pero el equipo no está contento con él
- ⇒ **Una Metodología Ágil puede ser una buena forma de empezar**
- No involucra gran inversión
 - A los programadores les (suele) gustar
 - A los clientes les ofrece mayor visibilidad y menor riesgo en el proyecto

FIN