

METODOLOGÍAS DE DESARROLLO DE SOFTWARE

INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA

Viña del Mar 2014

FORMAR

TRANSFORMAR



METODOLOGÍAS ÁGILES

La Programación eXtrema o XP, es la metodología más destacada dentro de las denominadas ágiles gracias a sus buenas prácticas, su difusión y sus excelentes resultados.



METODOLOGÍAS ÁGILES

La programación extrema es una metodología ágil de desarrollo de software que intenta **potenciar las relaciones** entre los integrantes del proyecto y se **concentra en la adaptabilidad como clave para el éxito** del desarrollo.

Según Kent Beck: “*XP es un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollo de software*”

METODOLOGÍAS ÁGILES



Kent Beck es ingeniero de software estadounidense, uno de los creadores de las metodologías de desarrollo de software de programación extrema (eXtreme Programming o XP) y el desarrollo guiado por pruebas (Test-Driven Development o TDD), también llamados metodología ágil. Beck fue uno de los 17 firmantes originales del Manifiesto Ágil en 2001.

METODOLOGÍAS ÁGILES

Su nombre, programación extrema, es debido a que lleva un conjunto de técnicas y principios de sentido común a niveles extremos. Las practicas que se llevan a cabo son:

- Programación en Pares.
- Pruebas Unitarias.
- Pruebas Continuas.
- Refactorización.



METODOLOGÍAS ÁGILES

XP esta orientada al desarrollador, al usuario, al resultado real y al incremento de funcionalidad.

Sus inicios son de 1996, donde Beck llegó a la empresa y observó que los problemas del desarrollo están basados en las estructuras y las metodologías. Sus ideas originales sobre un buen desarrollo distaban demasiado de los modelos rígidos utilizados que hacían gran hincapié en el uso de extensa y refinada documentación.



FORMAR

TRANSFORMAR

METODOLOGÍAS ÁGILES

Objetivos de la Programación Extrema, esta centrada en los resultados. Sin embargo, XP se hace foco en la **satisfacción al cliente**. Se intenta desarrollar el producto de acuerdo a las **expectativas del usuario**, no sólo respecto a la funcionalidad sino también en **tiempo y forma**. Además, el cliente puede sentirse libre de realizar cambios en sus especificaciones sin que éstas alteren el desarrollo del proyecto, produzcan crisis interpersonales ni provoquen un descenso de la calidad.

La programación extrema intenta cumplir con los puntos del manifiesto ágil minimizando los elementos y sin incrementar la curva de aprendizaje de la metodología.

METODOLOGÍAS ÁGILES

Promesas de la Programación Extrema. Esta metodología promueve la relación entre los miembros del equipo, dándoles un trabajo justo y prometiéndoles que obtendrán ciertas ventajas por su implantación.

Promesas a los clientes y gestores.

- Obtener el máximo de valor en cada semana de desarrollo.
- Progresos concretos.
- Modificación de requisitos sin asumir costos prohibitivos.



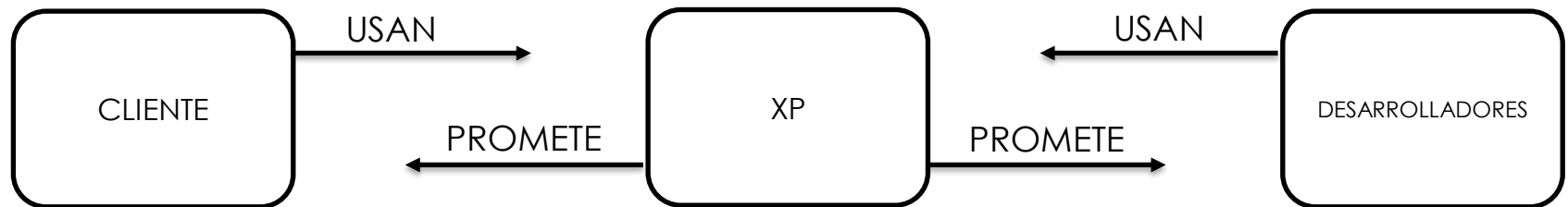
METODOLOGÍAS ÁGILES

Promesas a los desarrolladores.

- Trabajar en lo que realmente importa.
- Estarán acompañados por el equipo en la situación crítica.
- Mucho margen de acción en el desarrollo.
- Se enfocarán en el aspecto que les interesa (técnico) del proyecto.



METODOLOGÍAS ÁGILES



- Máximo Valor
- Progresos
- Cambio de requisitos

- Trabajo Estimulante
- Acción
- Libertad

METODOLOGÍAS ÁGILES

Cada vez que tenemos un proyecto entre manos, encontramos que los involucrados tienen **diferentes ideas** sobre qué es lo más importante de él y cuáles son las claves de su construcción. En las metodologías ágiles existe un **conjunto de valores críticos y necesarios para que éstas cumplan las expectativas.**

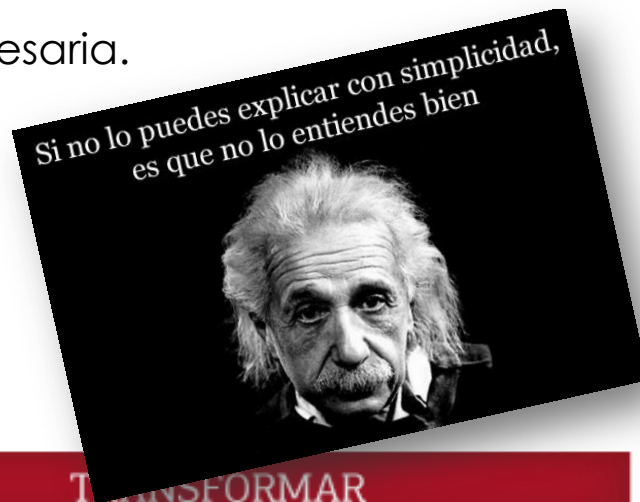
La programación extrema elige algunos valores y los promueve pensando, principalmente, en los **aspectos humanos sobre los técnicos.**

METODOLOGÍAS ÁGILES

La Simplicidad.

La base de la programación extrema es la **simpleza**. Se espera que afrontemos cada problema preguntándonos: **¿qué es lo más simple que nos pueden dar los resultados esperados?**.

La simpleza propuesta por XP se refiere a **evitar cualquier gasto innecesario**, y esto incluye la forma de desarrollar, las herramientas usadas y la cantidad de documentación necesaria.



METODOLOGÍAS ÁGILES

En el aspecto técnico se promueve el **diseño simple**, el cual garantiza **agilizar los tiempos de desarrollo y mantenimiento**. Los diseños complejos pueden ser reinventados para volverlos accesibles.

Las ventajas de la simpleza son más evidentes a medida que crece el proyecto. **La refactorización, el código compartido y la programación en pares** son algunas de las prácticas que se utilizan para poder mantener el valor de la simpleza como esencia de XP.



METODOLOGÍAS ÁGILES

La Comunicación.

La comunicación es una parte vital del proceso. En esta metodología, el valor de la comunicación se sustenta en prácticas muy específicas, como por ejemplo la documentación del código. Todo desarrollador debe comentar el código estable y, de esta forma, evitar los documentos externos.

La programación en pares con el cliente en el sitio es una práctica que se esfuerza por mantener la cooperación y la unidad en el proyecto.

METODOLOGÍAS ÁGILES

La Retroalimentación.

Todo proyecto cambia constantemente y, por supuesto, **cuanto antes** percibimos la **variación** y podemos responder a ella, tanto mejor será para el proceso total y para la calidad del producto.

Por lo tanto, se hace evidente que **la retroalimentación que se consigue al tener al cliente en el sitio para poder interactuar con él**, así como para evitar barreras entre integrantes del equipo, favorece el cumplimiento de los objetivos.



METODOLOGÍAS ÁGILES

Todos los clientes, desarrolladores y miembros del equipo tienen que poder responder en todo momento sobre qué es lo que se está haciendo, de qué manera y qué se quiere conseguir con ello.

De no ser así, sabremos que la comunicación y la retroalimentación no son las indicadas o sugeridas para el proceso.



FORMAR

TRANSFORMAR

METODOLOGÍAS ÁGILES

Prácticas de la Programación Extrema.

- **Retroalimentación a escala fina.** Los procesos cambiantes suponen una flexibilidad y una adaptación al cambio permanente. El aprendizaje continuo, en base a la realimentación y ejecución de los procesos, nos permite ajustar en poco tiempo aquellos elementos que nos pueden estar perjudicando, como también obtener beneficios.
 - El principio de pruebas.
 - Proceso de planificación.
 - Cliente en el sitio.
 - Programación en pares.

METODOLOGÍAS ÁGILES

Prácticas de la Programación Extrema.

- **Proceso continuo en lugar de por lotes.** En XP, en lugar de tener bloques de proceso que deben ser ejecutados con un estricto orden y con dependencia secuencial, tenemos un conjunto de prácticas que se realizan a lo largo del ciclo de vida del producto.
 - Integración continua.
 - Refactorización.
 - Entregas pequeñas.

METODOLOGÍAS ÁGILES

Practicas de la Programación Extrema.

- **Entendimiento compartido.** Un buen grupo con conocimiento, predisposición al trabajo y enfocado en un objetivo produce más y mejor.
 - Diseño simple.
 - Metáfora del Negocio (Visión del Sistema Completo).
 - Propiedad colectiva del código.
 - Estándar de codificación.

METODOLOGÍAS ÁGILES

Prácticas de la Programación Extrema.

- **Bienestar del programador.** El programador como parte esencial del desarrollo de software debe trabajar en condiciones normales y efectivas. Un profesional agobiado por las presiones diarias, además de la complejidad de las tareas, puede llegar a un nivel de saturación que le impida tener un rendimiento acorde y un grado de creatividad necesaria.

METODOLOGÍAS ÁGILES

Para lograr mejores resultados y bienestar del proyecto, XP nos presenta cuatro actividades básicas:

- **Codificar.** Escribir el código del producto, siguiendo los estándares y pautas establecidas.
- **Pruebas.** Realizar todas las actividades de pruebas contempladas por el equipo.
- **Escuchar.** Realizar reuniones para escuchar al cliente y al equipo de desarrollo, o cualquier método que promueva la discusión e intercambio de ideas.
- **Diseñar.** Obtener información del usuario y trasladarla a documentos validos para generar el código.

METODOLOGÍAS ÁGILES

Para todo el desarrollo en base a la metodología XP se utilizan sólo unos pocos artefactos.

- Historia de usuarios.
- Tareas de ingeniería.
- Pruebas de aceptación.
- Pruebas unitarias y de integración.
- Plan de entrega.
- Código.

METODOLOGÍAS ÁGILES

Historias de Usuario.

La fase de requisitos existe en todas las metodologías de desarrollo debido a que es el cliente el que manifiesta sus deseos o necesidades por un producto en particular.

En el caso de XP, las historias de usuario son la forma en que se especifican los requisitos del cliente. Consiste en un documento en donde se escriben las características esperadas. El nivel de detalle que se tiene en cada historia de usuario depende de la complejidad tratada.

METODOLOGÍAS ÁGILES

Estos documentos no son estáticos, sino que pueden ser modificados, reemplazados o destruidos.

Historia de Usuario		
Número:	Usuario:	
Nombre Historia:		
Prioridad en Negocio:	Riesgo en Desarrollo:	
Puntos Estimados:	Iteración Asignada:	
Programador Responsable:		
Descripción:		
Observaciones:		

METODOLOGÍAS ÁGILES

Tareas de Ingeniería.

Estas son elementos que se vinculan más al desarrollador, ya que permiten tener un acercamiento con el código. Como una historia de usuario puede ser dividida en varias tareas de ingeniería, es conveniente que éstas incorporen una numeración que permita localizar la historia de usuario original. Los datos necesarios son:

- Identificador.
- Relación con la historia.
- Tipo de tarea.
- Responsable.

METODOLOGÍAS ÁGILES

Tarea	
Número de Tarea:	Número de Historia:
Nombre de Tarea:	
Tipo de Tarea: Desarrollo/Corrección/Mejora/Otra	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

METODOLOGÍAS ÁGILES

Pruebas de Aceptación.

Las pruebas de aceptación surgen a partir de los elementos que el usuario destaca. Por lo tanto, es el cliente quien decide cuál es el escenario correcto para superar una prueba.

Las pruebas de aceptación son pruebas de caja negra, ya que el enfoque está orientado a conocer el producto final y el resultado que genera. Deben ejecutarse continuamente porque guía nuestro desarrollo y la visión que el usuario tiene de él.

Si no existen pruebas de aceptación nuevas, se dice normalmente que nada nuevo se ha hecho

METODOLOGÍAS ÁGILES

La prueba de aceptación está íntimamente ligada a las historias de usuarios y de su resultado positivo depende que el desarrollo vaya por el sendero correcto y el resultado sea satisfactorio.

Caso de Prueba	
Número de Caso de Prueba:	Número de Historia de usuario:
Nombre de Caso de Prueba:	
Descripción:	
Condición de Ejecución:	
Entrada:	
Resultado Esperado:	
Evaluación:	

METODOLOGÍAS ÁGILES

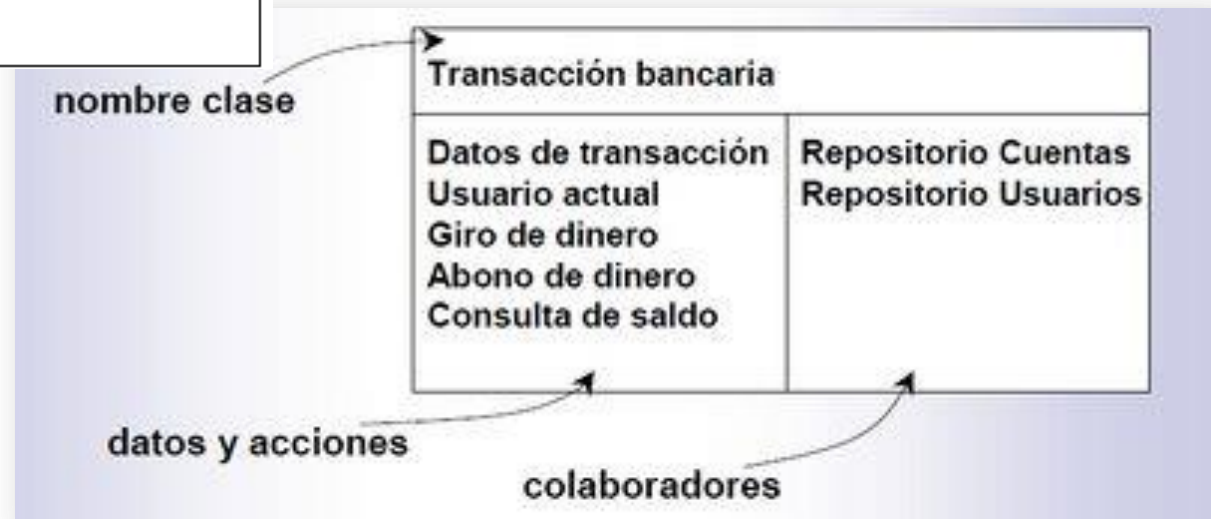
Tarjetas CRC.

Las tarjetas clase-responsabilidad-colaborador son elementos que intentan describir y facilitar el trabajo. Una clase puede ser cualquier concepto, elemento o persona. Los eventos que conoce y realiza son sus responsabilidades, atributos y métodos. Se denomina colaborador a todo aquel que interactúa con la clase. La confección de tarjetas supone llevar a cabo las siguientes actividades:

- Encontrar clases.
- Hallar responsabilidades.
- Definir colaboradores.
- Confeccionar tarjetas.

METODOLOGÍAS ÁGILES

Nombre de la clase.	
Responsabilidades	Colaboradores



METODOLOGÍAS ÁGILES

El Proceso XP.

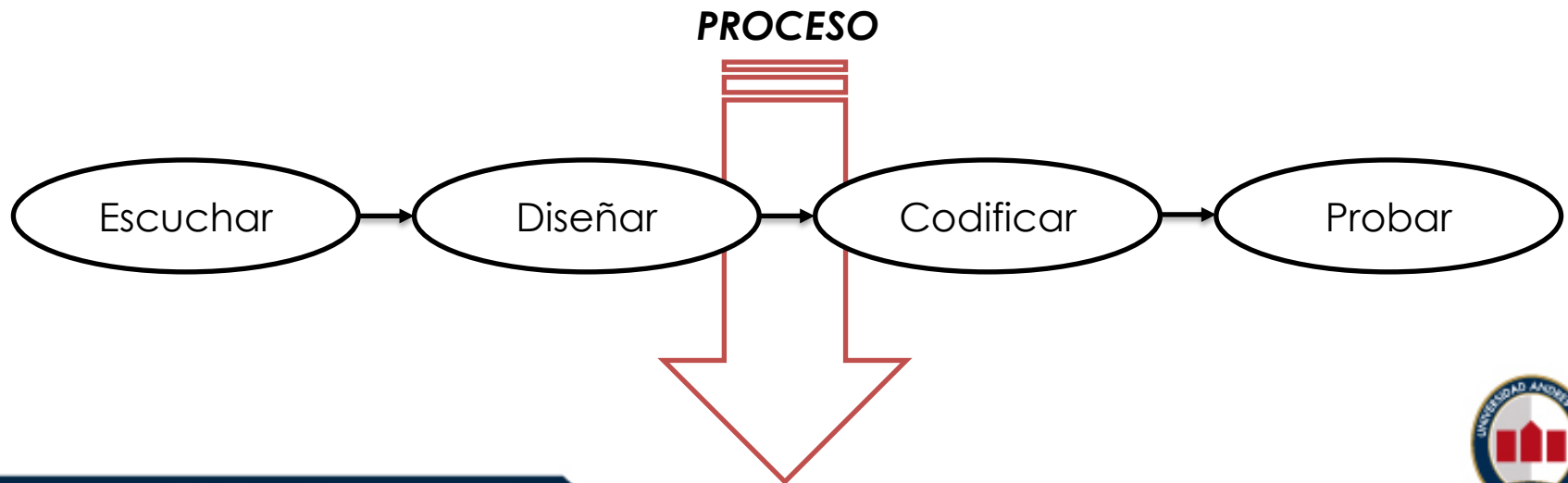
El ciclo de vida de los proyectos bajo XP pueden verse como una sucesión de definiciones por parte del cliente y una aceptación y continuo desarrollo de partes de los programadores.

La diferencia con otras metodologías es que estas sucesiones ocurren en muy corto tiempo. El equipo incorpora funcionalidad durante el día y esto le da la posibilidad al cliente de corregir y ajustar el rumbo.

METODOLOGÍAS ÁGILES

Los pasos que se llevan a cabo son los siguientes:

- El cliente decide el valor de negocio que quiere implementar.
- Se estima el esfuerzo necesario.
- El cliente selecciona qué construir.
- Se construye la selección.

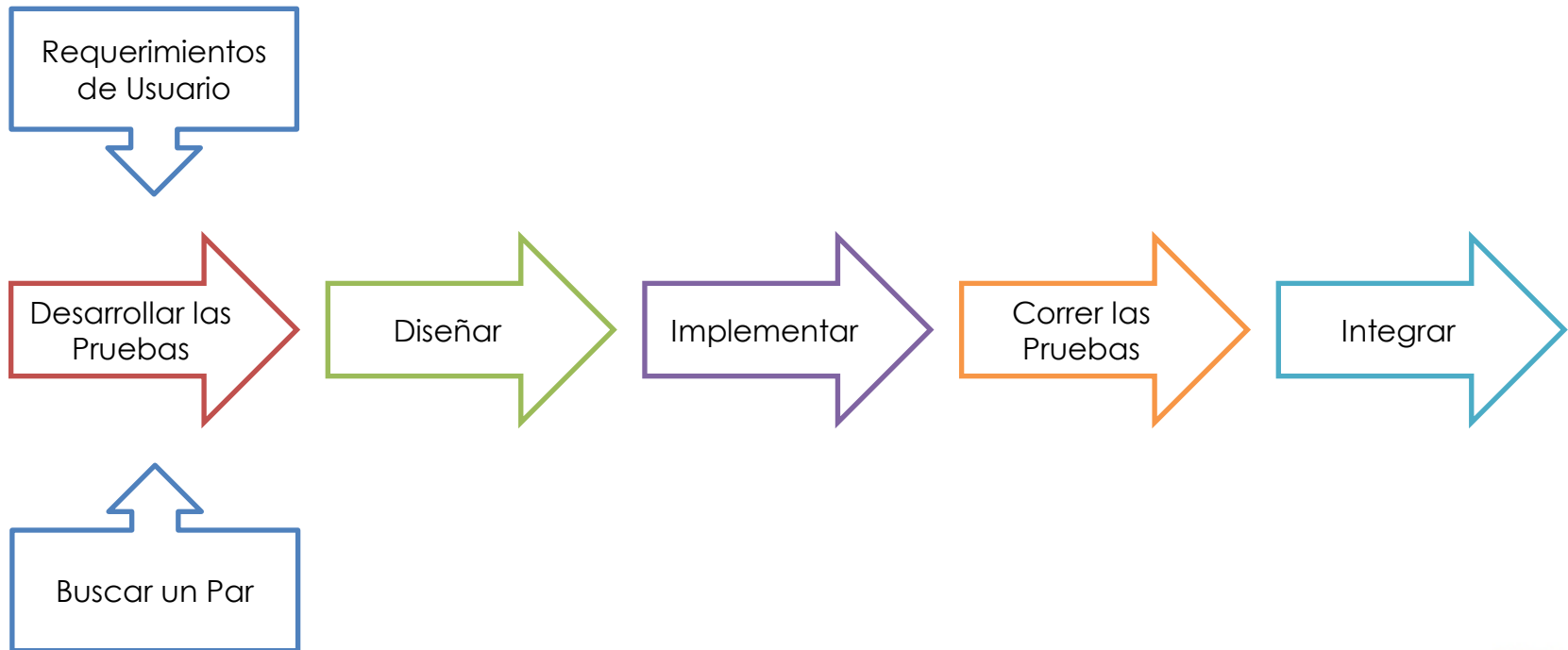


FORMAR

TRANSFORMAR

METODOLOGÍAS ÁGILES

En el ciclo de vida de XP nos encontramos con seis fases.



FORMAR

TRANSFORMAR

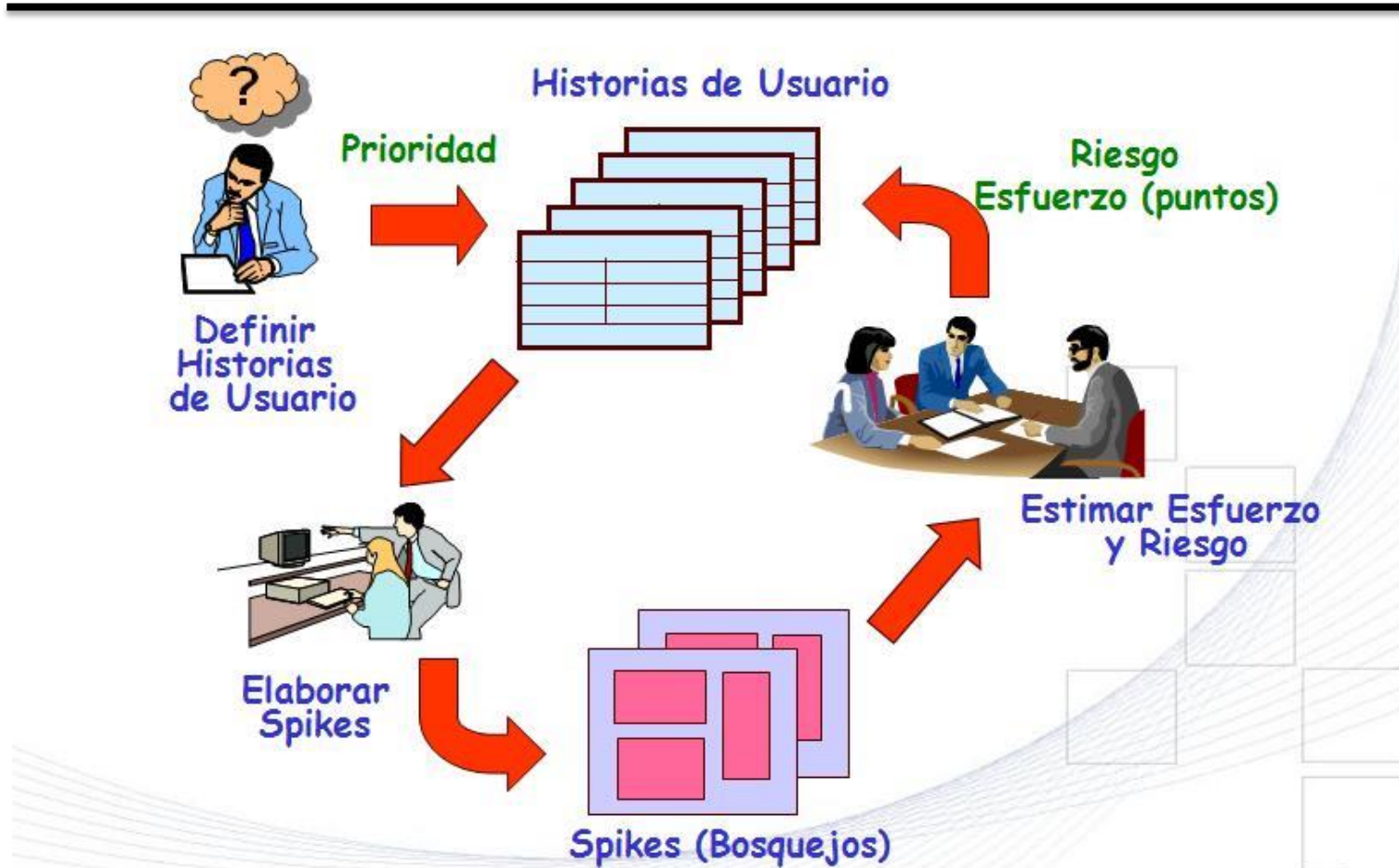
METODOLOGÍAS ÁGILES

Exploración.

Es la primera fase, cuando el cliente plantea cuáles son sus inconvenientes y qué soluciones considera apropiadas. Se generan las historias de usuarios con alto nivel de abstracción y sin ocuparse de detalles menores.

Todo el equipo comienza a aprender acerca de las herramientas que va a utilizar, sus puntos destacados, y se sortean las dificultades iniciales. La fase de exploración puede durar varias semanas, dependiendo de los conocimientos del equipo como de la dificultad del proyecto.

METODOLOGÍAS ÁGILES



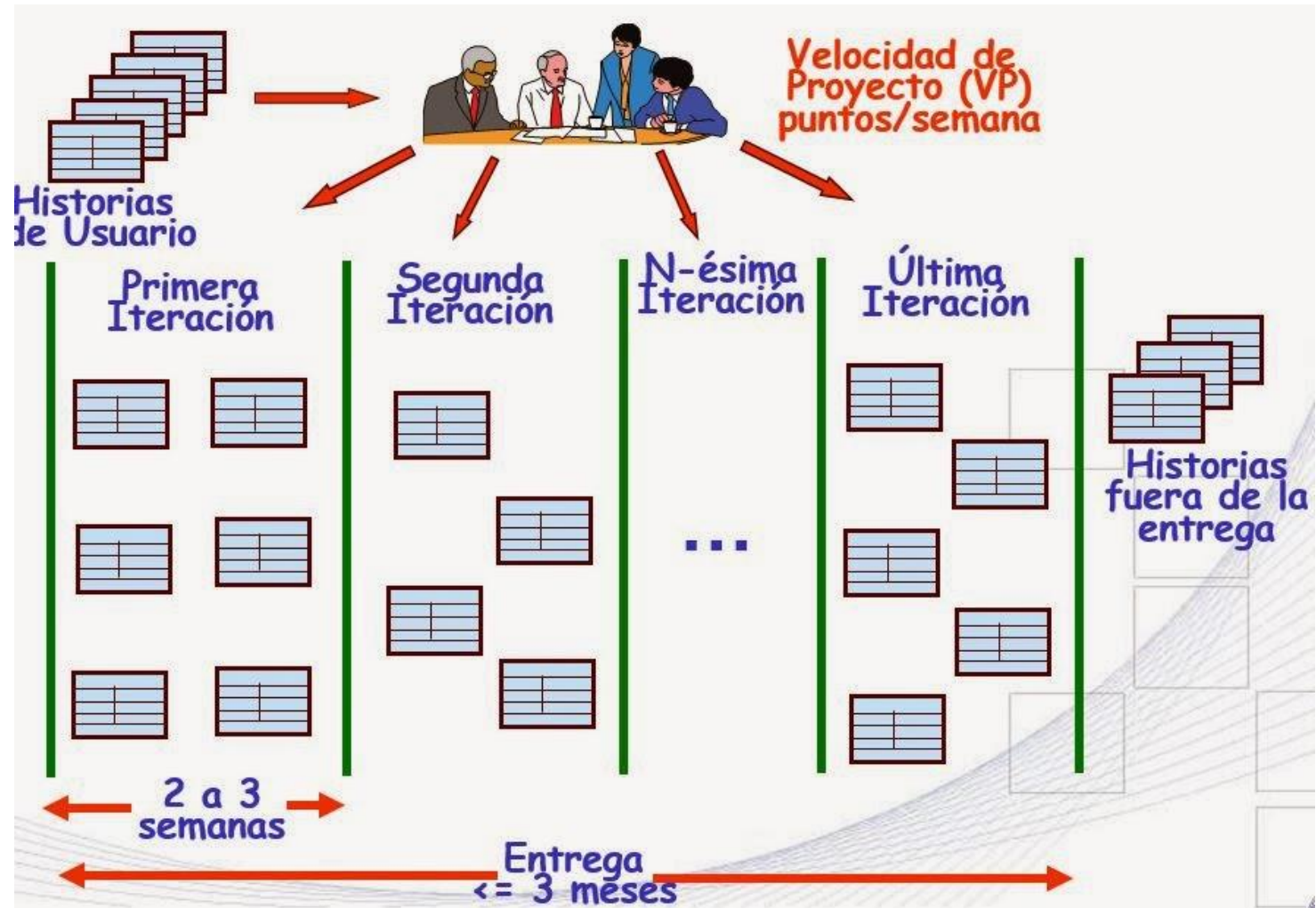
Spikes: también llamados bosquejos, son aproximaciones que van desde el nivel funcional al técnico y que luego puede acabar siendo material útil de desarrollo.

METODOLOGÍAS ÁGILES

Planificación de la Entrega.

Una vez finalizada las historias de usuarios, el cliente determina la prioridad y los programadores estiman el esfuerzo necesario para cada una. De acuerdo a las estimaciones y las expectativas del cliente se preparan y confirman las condiciones sobre la primera entrega. Ésta no debería demorar más de tres meses.

METODOLOGÍAS ÁGILES



FORMAR

TRANSFORMAR



UNIVERSIDAD
ANDRÉS BELLO

METODOLOGÍAS ÁGILES

Iteraciones.

Esta fase es un conjunto de interacciones de entre 2 y 3 semanas correspondientes a las planificaciones y a la definiciones de la arquitectura. En la iteración se tienen en cuenta las historias de usuarios seleccionadas, las faltantes y las tareas que se realizarán.

Todas las historias de usuario se volcarán en tareas de programación que luego serán abordados por las parejas de programadores.

Una vez finalizada la etapa de iteraciones, el proyecto está listo para entrar a la fase de producción.

METODOLOGÍAS ÁGILES

Producción.

En la fase de producción se realizan pruebas y se evalúa si es necesario incorporar funcionalidad extra. Los cambios realizados impactan en el sistema y se debe decidir sobre el estado de la entrega, es decir, si tiene las características mínimas acordadas para la presentación al cliente.

METODOLOGÍAS ÁGILES

Mantenimiento.

La fase de mantenimiento, es el intento de sostener funcionando la versión actual. Por eso, al finalizar la producción de la primera versión, inmediatamente se le brinda soporte al cliente.

El equipo posiblemente verá incrementar su número de personal debido a estas nuevas tareas, no realizadas hasta el momento.

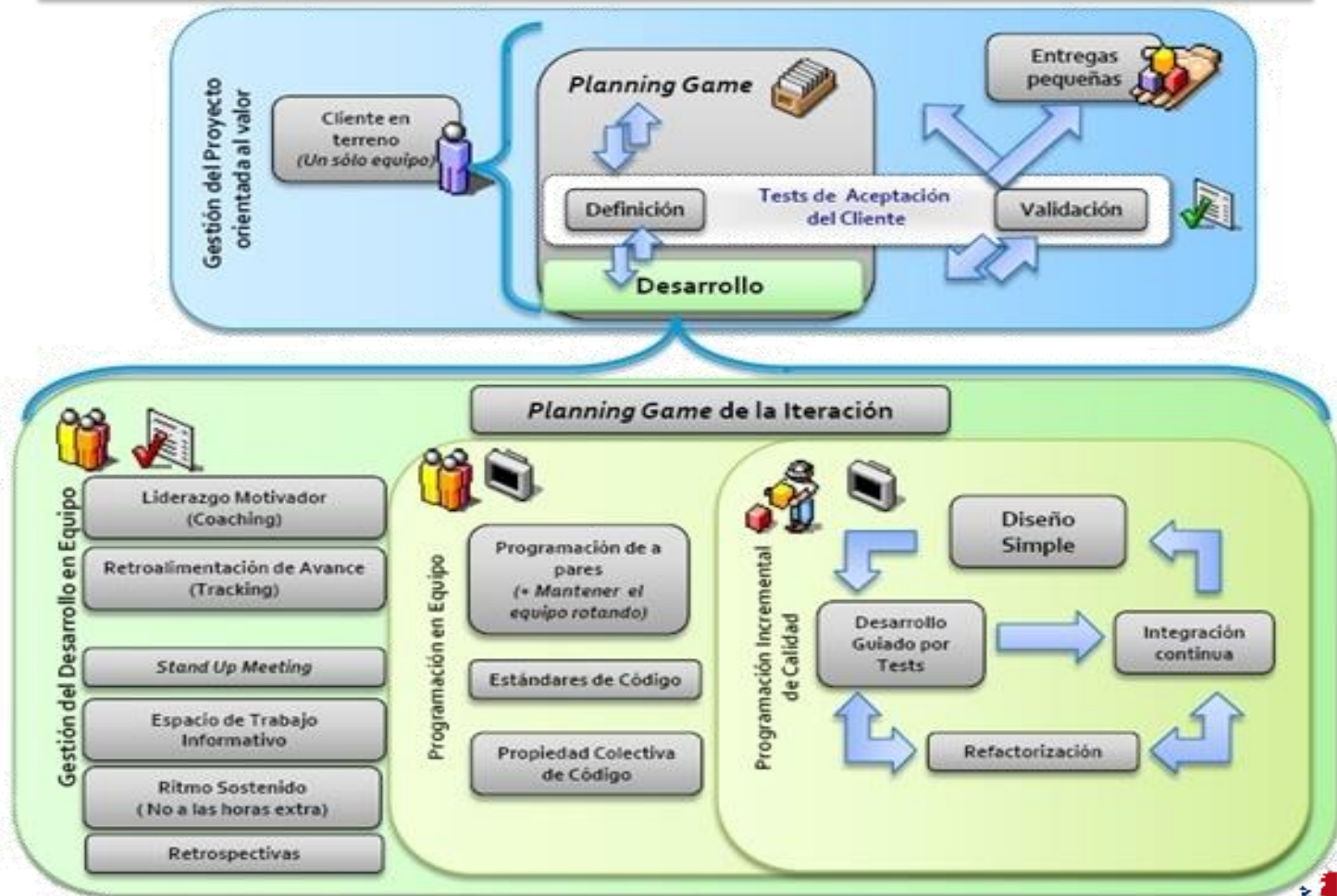
METODOLOGÍAS ÁGILES

Muerte del Proyecto.

La muerte o finalización del proyecto es cuando hemos agotado todas las **historias de usuario**. También puede ocurrir antes de lo pautado por falta de fondos o cambios en las condiciones del negocio del cliente que hacen no deseable el producto.

Una vez finalizada la incorporación de funcionalidad, se refinarán todas las variables asociadas al sistema, como la fiabilidad, el rendimiento o la **facilidad de uso**. En esta etapa también se genera la documentación final.

METODOLOGÍAS ÁGILES



FORMAR

TRANSFORMAR

METODOLOGÍAS ÁGILES

Roles del Equipo XP.

Xp se basa en un equipo pequeño de personas con habilidades complementarias que tienen un mismo objetivos. Por lo tanto, su conformación es una tarea clave en la metodología.



METODOLOGÍAS ÁGILES

El Cliente. Es parte del equipo de trabajo y sus principales tareas están relacionadas con:

- Decidir que se implementará.
- Saber el estado.
- Conocer el progreso.
- Añadir, cambiar o quitar requerimientos.
- Obtener un sistema funcionando cada cierto periodo de tiempo.



METODOLOGÍAS ÁGILES

El **Programador**. Es la estructura fundamental del proyecto y sus tareas son:

- Decidir cómo se implementan las soluciones.
- Crear el sistema con la mayor calidad posible.
- Pedir aclaraciones al cliente.
- Estimar el esfuerzo de implementación.
- Cambiar las estimaciones.



METODOLOGÍAS ÁGILES

Para cumplir con sus tareas, cada uno posee roles dentro del equipo XP, cada rol tiene una función clara durante las diferentes fases del proyecto.

Roles:

- Cliente.
- Programador.
- Encargado de pruebas.
- Encargado de seguimiento.
- Entrenador.
- Gestor.

METODOLOGÍAS ÁGILES

También existen otros roles que son complementarios en todo proyecto:

- Consultor.
- Analista.
- Operador.
- Manager del proyecto.

METODOLOGÍAS ÁGILES

El Programador debe:

- Escribir las pruebas unitarias.
- Producir el código del programa.
- Estimar las historias de usuarios.
- Participar en las propuestas por la metodología.

METODOLOGÍAS ÁGILES

El Cliente debe:

- Escribir las historias de usuario.
- Diseñar las pruebas de aceptación.
- Priorizar las historias de usuario.
- Aportar la dimensión del negocio al equipo de desarrollo (valor del negocio).
- Representar al colectivo de usuarios finales.
- Estar siempre disponible para consultas.

METODOLOGÍAS ÁGILES

El Encargado de Pruebas (tester) debe:

El **tester**, es la persona que está designado para los test de funcionamiento y asegurar que el sistema trabaje de una manera que resulte adecuada.

Los desarrolladores continuamente están trabajando sobre el código y la resolución de defectos, por lo tanto los errores encontrados por los encargados de prueba tienden a ser menores.

METODOLOGÍAS ÁGILES

Se debe dejar en claro que el encargado de pruebas es un integrante más del equipo y no un consultor o revisor externo. Entonces, el tester se encarga de cumplir las siguientes tareas:

- Ayudar al cliente a escribir las pruebas de aceptación.
- Ejecutar las pruebas regularmente.
- Ejecuta las pruebas funcionales.
- Difundir los resultados en el equipo.
- Seleccionar las herramientas adecuadas para ejecutar las pruebas.

METODOLOGÍAS ÁGILES

El Encargado de seguimiento (tracker) debe:

Comprobar que el conjunto de tareas se esté realizando de manera adecuada, acercando las estimaciones con la realidad. *Se encarga de medir introduciendo métricas simples para el proyecto.*

El uso de herramientas sencillas colaborará con la eficacia del proyecto.

- *Mide las desviaciones a las estimaciones y comunica los resultados.*
- *Realiza el seguimiento de cada iteración.*
- *Revalúa la posibilidad de incorporar o eliminar historias de usuario.*

METODOLOGÍAS ÁGILES

Para resumir mejor las tareas que le corresponden, se puede decir que el encargado de seguimiento se ocupa de realizar lo siguiente:

- Discutir y colaborar con el equipo.
- Adoptar métricas (simples).
- Discutir y difundir las métricas.
- Verificar las desviaciones del proyecto.
- Refinar los métodos de estimación utilizados.
- Realizar el seguimiento de las iteraciones.
- Reportar los progresos del equipo.
- Conservar los valores históricos.

METODOLOGÍAS ÁGILES

Entrenador (coach) debe:

Es el responsable de conocer a fondo el proceso XP para poder explicarlo a los demás y así facilitar la comunicación del equipo.

Para cumplir de forma correcta con su rol, debe tener **conocimientos técnicos** acabados, además de las necesarias **habilidades interpersonales** para relacionarse.

- Se encarga del proceso global.
- Garantiza que se sigue la filosofía XP.
- Provee guía y ayuda a los miembros del equipo acerca de XP.

METODOLOGÍAS ÁGILES

El Gestor (big boss) debe:

Es el vinculo entre los clientes y los programadores. Su rol es de facilitador.

- Es el máximo responsable del proyecto.
- Hace de enlace con los clientes.

El Consultor debe:

- No forma parte del equipo.
- Tiene un conocimiento específico de un área en concreto.
- Ayuda a resolver un problema puntual (de spike tecnológico o de valor al negocio).

METODOLOGÍAS ÁGILES

Cuándo utilizar XP.

La programación extrema fue concebida pensando en desarrollos que presenten las siguientes características:

- Proyectos con requisitos variables, cambiantes.
- Proyectos de alto riesgo.
- Proyecto con pocos programadores.

METODOLOGÍAS ÁGILES

Proyectos con requisitos variables, cambiantes.

Casi todos los proyectos de software podrían entrar en esta categoría, debido a que es muy poco frecuente que el establecimiento de requisitos sea extenso, detallado, profundo y lo suficientemente fiable para que no se deban introducir cambios en medio del proceso.

METODOLOGÍAS ÁGILES

Proyectos de alto riesgo.

Todos los proyectos poseen riesgos (del negocio, de gestión y del producto). Aquellos proyectos que proponen soluciones con calendarios realmente ajustados suponen un riesgo para su cumplimiento.

Lo mismo sucede en proyectos altamente innovadores que **no permiten utilizar líneas base o conocimientos anteriores** directamente sobre el proceso, sino que implican **crear nuevos caminos a medida que avanzamos en el desarrollo.**

METODOLOGÍAS ÁGILES

Proyecto con pocos programadores.

Las organizaciones con poco personal generalmente no pueden estructurarse de forma tal de dotar a todas las áreas de los profesionales adecuados.

XP permite que equipos de entre 2 y 12 desarrolladores puedan seguir perfectamente un proceso controlado, con normas de calidad y mejores practicas sin necesidad de mayor cantidad de recursos humanos.

METODOLOGÍAS ÁGILES

Criticas a XP o Detractores de XP.

- XP sólo funciona con equipos con experiencias, disciplinadas y con **visión del negocio**. Si tenemos a programadores “normales”, al aplicar XP obtendremos el efecto contrario, desarrollo mucho más lento y lleno de errores.
- Las empresas no dejarán que un cliente se integre al proyecto de desarrollo, y si conseguimos que lo consientan, seguramente será aquel usuario final que tenga menor valor para el negocio y que no represente **los problemas de la empresa**. Así, las historias de usuario y su asignación no serán representativas y construiremos la aplicación sobre una falacia.

METODOLOGÍAS ÁGILES

- Cuando se va a hacer un proyecto en una empresa, lo primero que se pide es saber el tiempo y el alcance, antes de aceptar un proyecto. Y con XP ambas son variables a lo largo del proyecto. Sin haberlas prefijado, no se puede hacer proyectos para terceras empresas.
- XP se basa en las pruebas y los tests, pero en un entorno real, con la presión que suelen sufrir los proyectos de desarrollo, lo primero que se reduce a su mínima expresión son las pruebas.

METODOLOGÍAS ÁGILES

- La programación en parejas es una utopía. Al final deriva en que uno trabaja y el otro descansa. Además, es de difícil justificación de cara al cliente que paga el software, ya que la medida hora/hombre es la que impera en el mercado. ¿cómo facturamos si se ve que cada dos personas, una sola apunta cosas a la otra?
- Los equipos XP han de ser pequeños, no se podría realizar con equipos de proyectos grandes.
- Deja de lado la documentación, algo fundamental para controlar la alta rotación en las empresas de servicios TI.



Viña del Mar 2014

FORMAR

TRANSFORMAR