

Unidad V

Tema: Metodologías para el desarrollo ágil de SW

UNTDF – 2020

Problemas comunes de proyectos de SW

- Falta de entendimiento del negocio.
- Cambios en el negocio y en los requerimientos.
- Altos costos de mantenimiento.
- Gran cantidad de defectos.
- Inestabilidad de las personas.
- Retrasos en las entregas.
- Proyectos cancelados

Problemas de Metodologías tradicionales

- Parten de:
 - imponer un modelo predecible sobre una entidad compleja.
 - minimizar o restringir el cambio en todas las fases.
 - congelar los requerimientos en forma temprana.
 - congelar la arquitectura y diseño en las primeras fases del ciclo de vida.
- Son orientadas a la documentación.
- Priorizan a los procesos sobre las personas.
- Plantean la construcción en términos de todo o nada.
- No incorporan feedback sobre el proceso

Necesidad de metodologías de DSW

- Creciente interés en metodologías de desarrollo durante la década pasada. De nada sirven buenas notaciones y herramientas si no se proveen directivas para su aplicación.
- Procesos “pesados o burocráticos”: El proceso de desarrollo llevaba asociada un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada.
- Esquema "tradicional" para abordar el desarrollo de software: efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos).

Necesidad de metodologías de DSW

- Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales:
 - donde el *entorno del sistema es muy cambiante*, y
 - en donde se *exige reducir drásticamente los tiempos de desarrollo* pero manteniendo una alta calidad.
- Ante las dificultades para utilizar metodologías tradicionales, muchos equipos de desarrollo no utilizan procesos de desarrollo, asumiendo el riesgo que ello conlleva.

Metodologías ágiles. Cómo nacen

- Surgieron a partir de la *necesidad de dar respuestas rápidas a los requerimientos cambiantes de las organizaciones*, reduciendo los tiempos de desarrollo
- Premisa de las MA:
 - No evitar el cambio, sino incorporarlo
 - Controlar el costo del cambio

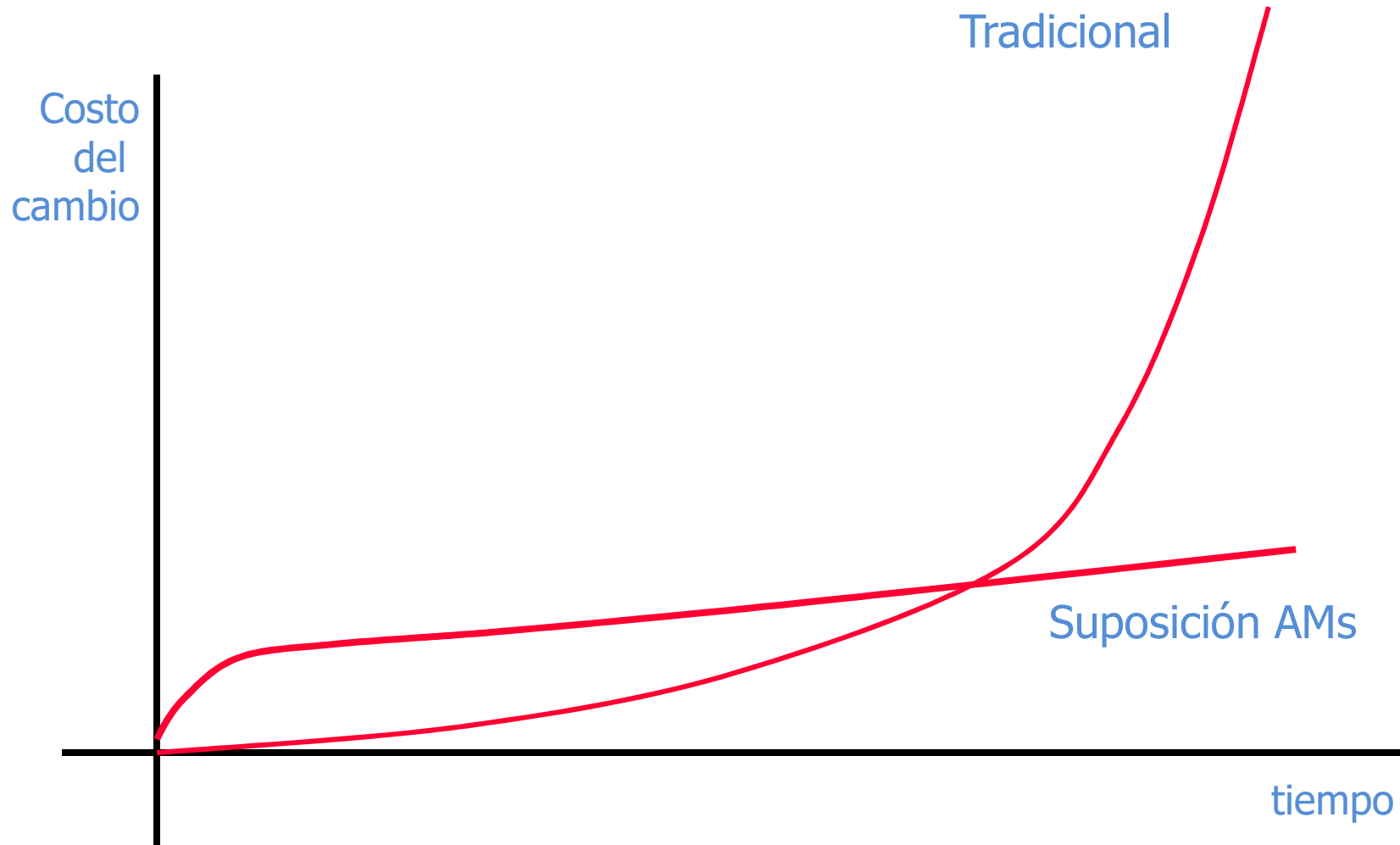
Metodologías ágiles. Qué promueven

- *Las metodologías ágiles promueven un proceso de gestión de proyectos que fomenta:*
 - el trabajo en equipo
 - un conjunto de mejores prácticas de ingeniería que permiten la entrega rápida de software de alta calidad
 - un enfoque de negocio que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía

Desarrollo ágil de Software

- **Desarrollo ágil de software**: grupo de metodologías de desarrollo de software que se basan en principios similares incluidos en el **Manifiesto ágil**.
- Los procesos ágiles, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Importancia de las MA



Metodologías ágiles. Evolución

- **Desarrollo ágil de software >>>** Evolucionó a mediados de los años 1990 como parte de una reacción contra los métodos de “peso pesado” (muy estructurados y estrictos).
- Inicialmente >>> Métodos de "peso liviano".
- **En 2001 >>>** En reunión de 17 figuras de métodos livianos en Utah se adopta el nombre de "**Metodologías ágiles**"
 - Se crea el Manifiesto ágil
 - Poco después se forma la "alianza ágil" (promueve el DA).
- Métodos similares se crearon antes del 2000: Scrum, Crystal Clear, Feature driven development, Programación Extrema o XP

Metodologías ágiles.

Desarrollo iterativo incremental. Principios y prácticas

Metodologías ágiles. Características

- Los métodos ágiles se caracterizan a veces como diametralmente opuestos a métodos disciplinados.
 - Esta distinción es errónea, ya que esto implica que los métodos ágiles no son planificados ni disciplinados.
- Distinción más apropiada: **Los métodos pueden ser desde “adaptativos” a “predictivos”**.
 - Los **métodos adaptativos** se centran en la adaptación rápida a los cambios.
 - Los **métodos predictivos**, por el contrario, se centran en la planificación del futuro en detalle
- *Los métodos ágiles se acercan más al lado adaptativo*

Métodos Iterativos y evolutivos

- Las metodologías ágiles son un subconjunto de los métodos iterativos y evolutivos.
- Las primeras ideas sobre dichos métodos datan de los años 60s

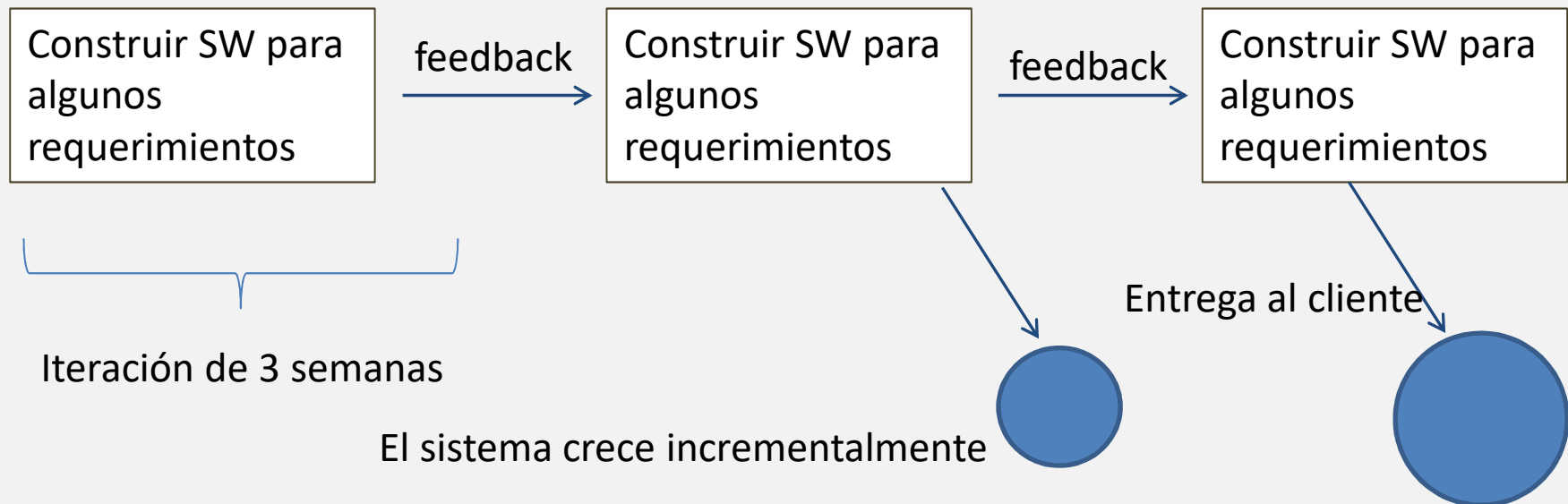


Desarrollo Iterativo

- El Desarrollo Iterativo es un enfoque para construir software en el cual el ciclo de vida completo se compone de iteraciones en secuencia.
- Cada iteración es un mini-proyecto compuesto de actividades tales como: requerimientos, análisis, diseño, programación y prueba.
- La meta al final de cada iteración es un **entregable** (*release*) de la iteración, es decir un sistema estable, integrado y parcialmente completo.
 - ✓ Algunos entregables son internos, compartidos en el grupo de desarrollo.
 - ✓ El entregable de la iteración final es el producto completo para entregar al mercado o a los clientes

Desarrollo Iterativo e Incremental (DII)

■ Evolución del sistema en iteraciones



- ✓ El sistema crece incrementalmente iteración por iteración agregando nuevas características (*desarrollo incremental*).
- ✓ El concepto de crecimiento de un sistema vía iteraciones es llamado Desarrollo Iterativo e Incremental. Constituye el corazón de los métodos ágiles

DII manejado por riesgos y por el Cliente

- **DII promueve el Planeamiento Iterativo Manejado por los riesgos y por el Cliente**
- DII “**risk-driven**” promueve elegir los elementos más dificultosos (de mayor riesgo) en las primeras iteraciones
- DII “**Client-driven**” implica que la elección de las características para la próxima iteración se definen con el cliente. El Cliente tiene control y decisión durante el desarrollo del proyecto, dado que interviene en el mismo iteración por iteración

DI time-boxed (de tiempo prefijado)

- **Timeboxing de operaciones** es una práctica para fijar la fecha final de las iteraciones y no permitir que cambie.
- Si no se puede cumplir , en lugar de aplazar la fecha, se reduce el ámbito de desarrollo para la iteración, de modo que el crecimiento del sistema sea estable
- La mayoría de los métodos iterativos ágiles recomiendan iteraciones entre 1 y 6 semanas.

DII. Desarrollo evolutivo y adaptativo

- **Desarrollo evolutivo iterativo** implica que los requerimientos, plan, solución, etc. **son refinados durante el curso de las iteraciones**, más que “congelados” en una gran especificación antes de comenzar el desarrollo de las iteraciones.
- **Desarrollo adaptativo** es un término relacionado. Implica que **los elementos se adaptan en respuesta al feedback de usuarios, test, desarrolladores, etc.**

Principios y prácticas de métodos ágiles

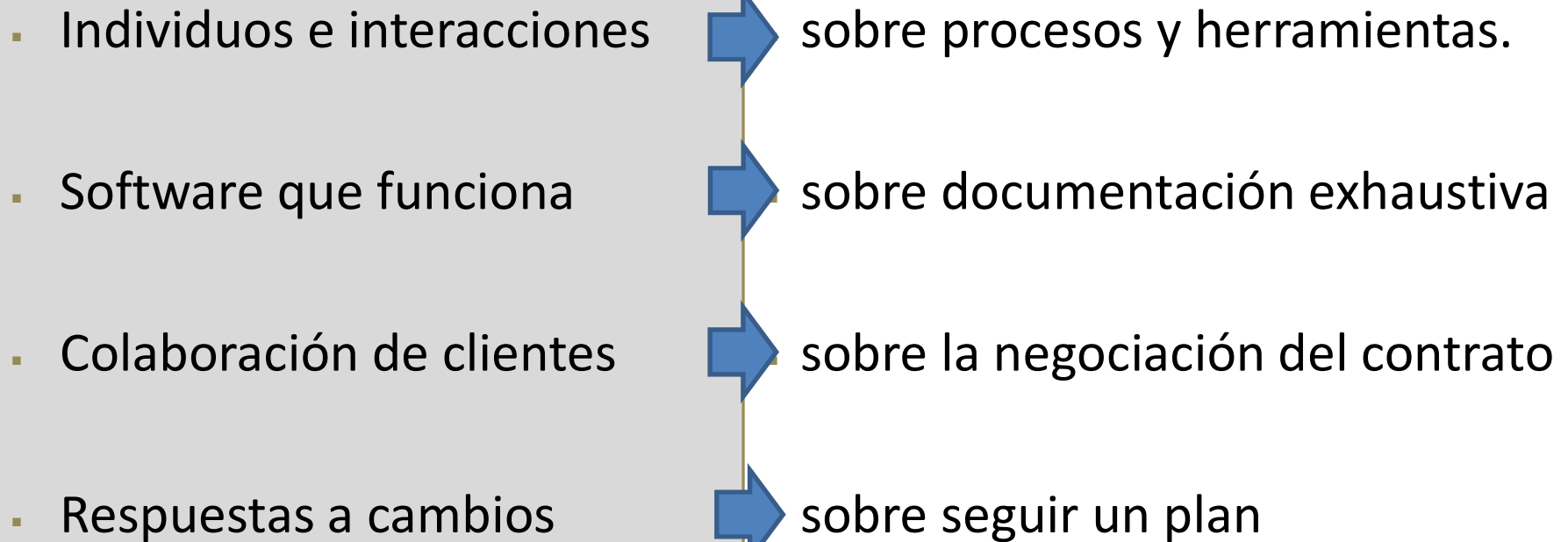




- **Los métodos de Desarrollo ágil** aplican desarrollo iterativo-incremental y evolutivo, planeamiento adaptativo e incluyen otros valores y prácticas que promueven agilidad, respuesta a los cambios rápida y flexible.
- **El lema de los métodos ágiles es “abrazar el cambio”**
- Los métodos ágiles comparten algunas prácticas como: iteraciones de tiempo-fijo corto, refinamiento evolutivo y adaptativo de planes y metas, y fundamentalmente el desarrollo iterativo

Manifiesto ágil (Reunión Utah - EEUU, 2001)

■ Metodólogos:

- Kent Beck (XP), Mike Beedle, Arie van Bennekum (DSDM), Alistair Cockburn (Crystal), Ward Cunningham (XP), Martin Fowler (XP), James Grenning (XP), Jim Highsmith (ASD), Andrew Hunt (Pragmatic Programming), Ron Jeffries (XP), Jon Kern (FDD), Brian Marick, Robert C. Martin (XP), Steve Mellor, Ken Schwaber (Scrum), Jeff Sutherland (Scrum) y Dave Thomas (Pragmatic Programming)

Manifiesto ágil

- 
- Individuos e interacciones  sobre procesos y herramientas.
 - Software que funciona  sobre documentación exhaustiva
 - Colaboración de clientes  sobre la negociación del contrato
 - Respuestas a cambios  sobre seguir un plan

Los elementos de la parte derecha son importantes, pero se valoran más los de la izquierda

Principios

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le reporte un valor
2. Se acepta que los requerimientos cambien, aún en etapas tardías del desarrollo.
3. Entregar frecuentemente software que funcione, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre una entrega y la siguiente
4. Los responsables del negocio y los desarrolladores deben trabajar juntos de forma cotidiana durante todo el proyecto
5. Construir proyecto en torno a individuos motivados.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo

Principios

7. El software que funciona es la medida principal de progreso
8. Los procesos ágiles promueven un desarrollo sostenible. Los desarrolladores y usuarios deberían mantener un ritmo constante
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad
10. La simplicidad es esencial
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos auto-organizados.
12. En intervalos regulares, el equipo reflexiona respecto de cómo llegar a ser más efectivo, y según esto ajusta su comportamiento

Metodologías ágiles. Cómo

Algunas prácticas básicas:

- Producir el primer delivery en semanas para tener rápido feedback.
- Mejorar la calidad constantemente.
- Test continuo, para detectar errores antes.
- Control del proceso:
- ✓ **Feedback frecuente para conocer la situación actual en intervalos regulares.**
 - El tiempo de iteración debe ser el más corto posible (entre 2 semanas y 1 mes).
- ✓ **El cliente tiene mayor control sobre el proyecto.**
 - El cliente y el equipo de desarrollo evalúan el progreso.
 - El cliente puede cambiar las funcionalidades del software de acuerdo a los cambios en el negocio.

Metodologías tradicionales vs. Agiles

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El <u>modelado</u> es accesorio, modelos desechable (algunos)	Más Artefactos. El modelado es esencial, mantenimiento de modelos
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un <u>contrato</u> 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 <u>proyectos pequeños</u> . 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 <u>arquitectura</u> se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los <u>aspectos humanos</u> : el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Se <u>esperan cambios</u> durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Métodos ágiles

Metodología	Acrónimo	Creación	Tipo de modelo	Característica
Adaptive Software Development	ASD	Highsmith 2000	Prácticas + Ciclo de vida	Inspirado en sistemas adaptativos complejos
Agile Modeling	AM	Ambler 2002	“Metodología basada en la práctica”	Suministra modelado ágil a otros métodos
Crystal Methods	CM	Cockburn 1998	“Familia de metodologías”	MA con énfasis en modelo de ciclos
Agile RUP	dX	Booch, Martin, Newkirk 1998	Framework / Disciplina	XP dado vuelta con artefactos RUP
Dynamic Solutions Delivery Model	DSDM	Stapleton 1997	Framework / Modelo de ciclo de vida	Creado por 16 expertos en RAD
Evolutionary Project Management	Evo	Gilb 1976	Framework adaptativo	Primer método ágil existente
Extreme Programming	XP	Beck 1999	“Disciplina en prácticas de ingeniería”	Método ágil radical
Feature-driven development	FDD	De Luca & Coad 1998 Palmer & Felsing 2002	“Metodología”	Método ágil de diseño y construcción
Lean Development	LD	Charette 2001, Mary y Tom Poppendieck	“Forma de pensar” – Modelo logístico	Metodología basada en procesos productivos
Microsoft Solutions Framework	MSF	Microsoft 1994	Lineamientos, Disciplinas, Prácticas	Framework de desarrollo de soluciones
Rapid Development	RAD	McConnell 1996	Survey de técnicas y modelos	Selección de <i>best practices</i> , no método
Rational Unified Process	RUP	Kruchten 1996	Proceso unificado	Método (¿ágil?) con modelado
Scrum	Scrum	Sutherland 1994 - Schwaber 1995	“Proceso” (framework de management)	Complemento de otros métodos, ágiles o no

Bibliografía

- Agile and Iterative Development. Larman C., Addison Wesley, 2003
- Extreme Programming Explained: Embrace Change, 2a ed. Beck K., Andres C. Addison Wesley, 2004
- El Proceso Unificado de Desarrollo de Software. Jacobson, Booch, Rumbaugh. Addison Wesley, 2000
- Applying UML and Patterns 3ª ed. Larman. Addison Wesley, 2004
- The Rational Unified Process More Easy. Kroll, Kruchten, Booch. 2004
- The Rational Unified Process. An Introduction. Kruchten. Ed. Addison Wesley, 2003
- SRUM y XP para programadores. Bahit E. SafeCreative ed., 2012