

Propuesta metodológica para el desarrollo de entornos de soporte para educación a distancia en ambientes Web

Autores

Imwinkelried Angel (imwinkelried@unlam.edu.ar)

Occhipinti Pedro (opedro@unlam.edu.ar)

Orthusteguy Fernando (forthus@unlam.edu.ar)

Rusticcini Alejandro (arusticcini@unlam.edu.ar)

Departamento de Pedagogía Universitaria –UNLAM-

Resumen

Durante el desarrollo de un campus virtual destinado a dar soporte a cursos a distancia dictados bajo la modalidad semipresencial, nos encontramos con una serie de dificultades que condujeron a la definición de una metodología de trabajo basada en métodos ágiles y programación extrema.

En el presente documento expondremos dicho enfoque metodológico, adecuado para el desarrollo y mantenimiento de entornos de soporte para educación a distancia en ambientes Web.

Este documento está dirigido a aquellas áreas o equipos de teletrabajo que desarrollen entornos y ambientes de soporte a educación a distancia, así como a aquellas personas que participen en el desarrollo de proyectos Web o cumplan roles de gestión de proyectos siguiendo métodos ágiles.

Palabras claves: Gestión; Teletrabajo; Educación; Distancia; Requerimientos; Configuración; Programación; Extrema;

Antecedentes

En el año 2000 la Universidad Nacional de La Matanza (UNLM) decide, entre otros cambios de índole pedagógico-académico, dictar en forma “Transversal” (común a todas las carreras de la Universidad), las materias de nombre “Computación I” y “Computación II”, que contienen dos niveles de aprendizaje acerca de aplicaciones de oficina (ofimática), para abastecer una población de 2500 alumnos por cuatrimestre aproximadamente. Esta oferta académica nace con una fuerte limitación en los cupos de inscripción, dada por las capacidades de los laboratorios equipados para el dictado de dichas asignaturas, dejando a muchos alumnos sin la posibilidad de cursar..

A mediados del 2001, como una estrategia de resolución para este cuello de botella se comienza a diseñar un sistema informático que pueda soportar el modelo de enseñanza a distancia.

Para llevarlo adelante, se constituyó, un equipo interdisciplinario que abordara la problemática, constituido por docentes de nuestra casa de estudios. Los mismos tuvieron a su cargo, la investigación y el desarrollo necesarios para la implantación del sistema de gestión de educación a distancia.

Paralelamente se comenzó con el desarrollo de los contenidos generales, para que los alumnos de la materia Computación I y II dispongan del material teórico y práctico adecuado para la cursada semipresencial de estas asignaturas.

Posteriormente se amplió el alcance de la plataforma para dictar otros cursos de tipo semipresencial, y para dar soporte a asignaturas que se dictan en modalidad presencial.

El sistema se encuentra actualmente operativo y funcionando. Desde Marzo del año 2002, registrando desde entonces una inscripción de más de dos mil alumnos.

En la actualidad está brindando soporte para las siguientes asignaturas:

Cátedra	Departamento Dirección
Computación I	Pedagogía
Computación II	Pedagogía
Diseño y Arquitectura Web	Ingeniería
Comunicación de Datos en Internet	Ingeniería
Sistemas de Computación II	Ingeniería
Programación I	Ingeniería
Educación a Distancia	Postgrado
Tópico de Redes	Postgrado
Taller/Curso para Tutores	MIEl

Arquitectura

El sistema de gestión de educación a distancia consta de tres componentes principales, a saber:

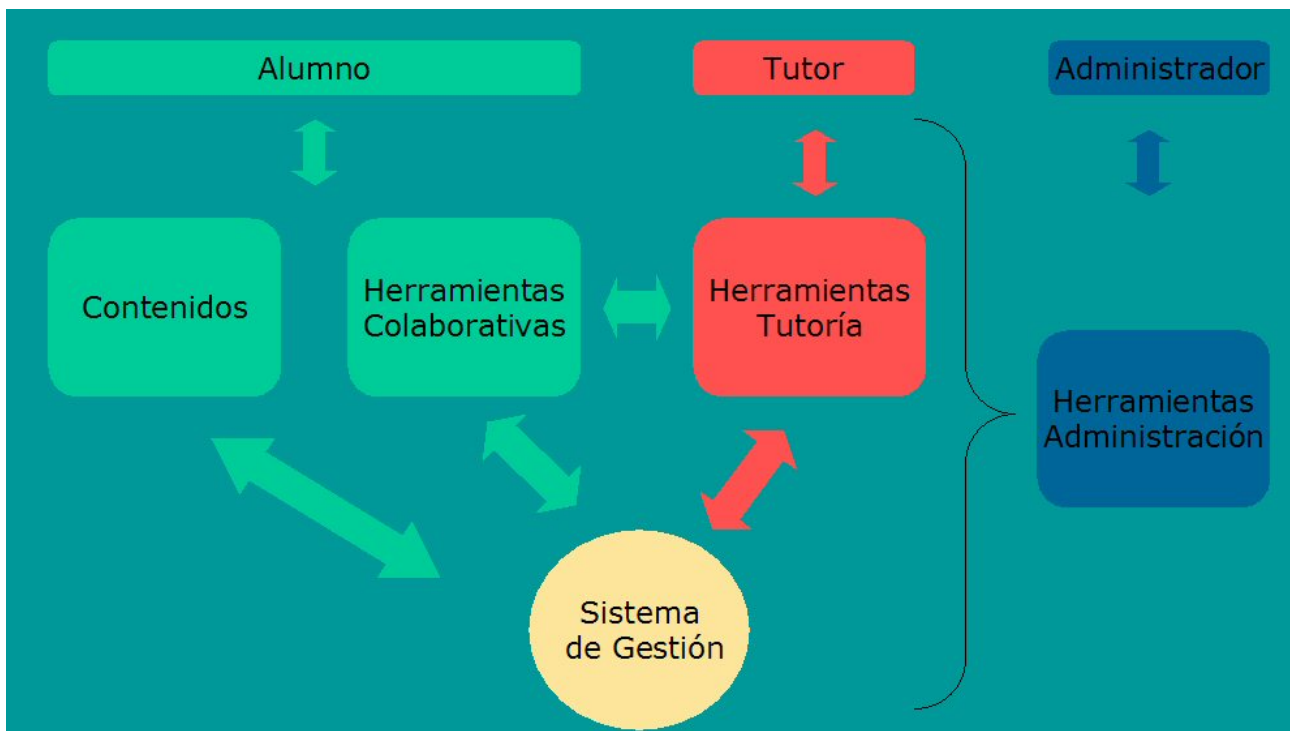
- Módulo Alumno
- Módulo Tutor
- Módulo Administración

El módulo alumno es el que permite a los cursantes tener acceso a los contenidos teóricos y prácticos de sus asignaturas y a su vez los provee de las herramientas colaborativas necesarias para mantener comunicación con sus tutores, e interactuar con los integrantes de un curso, como ser: foros, Chat, mensajería, glosario, calendario de eventos, etc.

El módulo tutor provee a quien cumpla este rol de los medios para realizar las tareas de seguimiento y control sobre los alumnos a su cargo, por ejemplo intercambio de prácticas, evaluación, calendario de actividades, mensajería, etc..

El módulo administración es el que permite la realización de tareas administrativas sobre los cursos en ejecución, por ejemplo altas/bajas/modificaciones de cursos, alumnos, tutores, etc.

Transversalmente a estos tres módulos el campus implementa dos funciones principales, la gestión de contenidos teórico-prácticos por un lado, y por el otro provee las herramientas básicas necesarias para mantener una comunicación entre los actores involucrados.



Características del proyecto.

El dominio del negocio en que desarrollamos nuestra plataforma de educación a distancia está caracterizada por las siguientes premisas:

- ❑ Requisitos inciertos o volátiles
- ❑ Desarrolladores responsables y motivados, organizados en equipos celulares de no más de tres personas.
- ❑ Usuarios comprometidos, que entienden y se involucran no solo en la definición de sus necesidades, sino también en el desarrollo y el perfeccionamiento de la solución.
- ❑ La dedicación de los miembros del equipo de trabajo es parcial y no coincidente con la disponibilidad horaria de los usuarios requerentes, o con otros miembros del equipo.
- ❑ La masa de requerimientos a analizar y procesar supera la capacidad operativa del grupo de trabajo.
- ❑ No se está sujeto a las limitaciones de un alcance o contrato fijo propios de un proceso predictivo
- ❑ Se esta sujeto a un marco regulatorio para el mantenimiento del sistema de gestión

Estas premisas coinciden mayormente con los factores que definen un proceso adaptable, por lo que basamos nuestra metodología en la aplicación de un método adaptativo o ágil en vez de predictivo.

Principios (Agile Manifiesto)

Se define como métodos ágiles a aquellos en los cuales las estrategias de desarrollo de software promueven prácticas adaptativas e iterativas, centradas en la comunicación intensiva entre las personas participantes, orientadas hacia la entrega de soluciones en tiempos relativamente breves, y que requieren que el usuario referente del dominio del negocio se involucre en forma directa.

El Manifiesto Ágil establece:

Individuos e interacción	Sobre	procesos y herramientas
software que funciona		documentación abarcadora
colaboración con el cliente		negociación contractual
Responder a los cambios		seguimiento de un plan

Aunque hay valor en los elementos a la derecha, valoriza más los de la izquierda.

Del manifiesto ágil se derivan los siguientes principios:

1. La prioridad más alta es satisfacer al cliente a través de la entrega temprana y continua de software funcional.
2. El dinamismo del dominio del negocio promueve el entusiasmo del equipo de desarrollo para responder a nuevos desafíos en forma constante.
3. Los procesos ágiles se pliegan al cambio en procura de una ventaja competitiva para el cliente.
4. Orientar los desarrollos a escalas de tiempo lo mas breve posibles, preferentemente de tres a cuatro semanas.
5. Los usuarios referentes del negocio y los desarrolladores deben trabajar juntos cotidianamente a través de todo el proyecto.
6. Construir proyectos basados en individuos altamente motivados. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software funcional es la medida primaria de progreso.
8. Los procesos ágiles promueven el dinamismo en el desarrollo. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de trabajo.
9. La simplicidad (el arte de maximizar la cantidad de trabajo que no se hace) es esencial.
10. Las mejores arquitecturas, requerimientos y diseños emergen de equipos que se auto-organizan.
11. A intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo, y ajusta su conducta en consecuencia.

Basándonos en estos principios y dada la particularidad de la conformación de nuestros equipos de trabajo, sin la posibilidad de realizar tareas en un entorno físico y temporal común, replanteamos la comunicación “cara a cara” y el “trabajar juntos” por espacios virtuales de encuentro, aprovechando las herramientas que desarrollamos para su aplicación a la enseñanza a distancia (e-mail, foros, Chat, calendario) e integrándolas a un sistema de gestión para conformar el repositorio de conocimiento del proyecto.

De esta manera hacemos transparente la aplicación de las prácticas de la derecha del cuadro del manifiesto ágil, que naturalmente se tiende a no aplicar.

Marco Regulatorio

La modalidad de Educación a Distancia, de antigua data en la Educación de nuestro país fue, sin embargo, recientemente contemplada por nuestra Legislación en la Ley 24.521 - Artículo 74, delegando en el entonces Ministerio de Cultura y Educación el dictado de las normas necesarias y designándolo como órgano de aplicación de las disposiciones emanadas por dicha Ley.

El Ministerio de Educación, Ciencia y Tecnología de la República Argentina ha dado un paso adelante para cubrir un fuerte vacío legal, al aprobar la Resolución N° 1717/04, estableciendo un nuevo marco normativo que intentará controlar la oferta de la educación a distancia y mejorar su calidad académica.

En el punto “5.5. Tecnologías de información y comunicación” de dicha resolución, se establece que “Los centros de apoyo deben tener un responsable académico y -según el modelo sustentado-, un responsable tecnológico. Es necesario que cuenten, además, con normativas específicas de uso y mantenimiento.”

Enfoque Metodológico

En la definición de la *normativa para el mantenimiento de la plataforma tecnológica*, consideramos tres pilares metodológicos que la sustenten:

- ❑ Gestión de Requerimientos e Incidentes
- ❑ Gestión de Proyectos
- ❑ Gestión de Cambios

La Gestión de Requerimientos enfoca el mantenimiento correctivo necesario para subsanar los incidentes detectados en el ambiente productivo; así como el mantenimiento adaptativo necesario para ajustar la plataforma a los cambios en las reglas de negocio, en la normativa y/o el marco regulatorio y al mantenimiento evolutivo que implementa mejoras técnicas en el sistema orientadas a incrementar la calidad del servicio.

La Gestión de Proyectos enfoca la asignación de equipos de trabajo para atender los requerimientos, y el seguimiento y control del desarrollo de las correspondientes soluciones.

La Gestión de Cambios enfoca al registro de los cambios que se efectúan en los componentes del software de la plataforma tecnológica en los ambientes de desarrollo, pruebas y producción; así como las familias de versiones que componen la solución a los requerimientos en dichos ambientes.

Tratamiento de Requerimientos/Incidentes

Los requerimientos nacen básicamente de “historias de usuario”, basándose en la experiencia de uso para obtener un registro de necesidades insatisfechas o de defectos de producto.

Los usuarios que pueden registrar requerimientos y/o incidentes en el sistema de gestión del proyecto, son principalmente los tutores (docentes) de las asignaturas que se dictan en modalidad semipresencial, y los docentes de los cursos dictados en modalidad presencial, quienes además de sus propias necesidades, reciben y filtran los requerimientos de sus respectivos alumnos.

La registración establece diferentes atributos para los incidentes que para los requerimientos.

Los requerimientos son evaluados y asignados a un equipo de desarrollo, integrado por el solicitante, junto con un analista-programador y un diseñador gráfico, quienes definen los aspectos funcionales, así como de navegabilidad y facilidad de uso de la solución.

Los coordinadores establecen requerimientos de seguimiento y control, basados en indicadores y reportes.

Las prioridades de asignación de requerimientos se determinan por un comité integrado por la coordinación del proyecto y los coordinadores de las asignaturas que participan de esta modalidad.

Los incidentes son asignados al equipo de desarrollo, desplazando las otras tareas ya asignadas. Como regla general, el máximo nivel de prioridad corresponde a los incidentes detectados en el ambiente productivo, y el segundo nivel a los detectados en las pruebas; siendo estos cambios de prioridad decisión directa del equipo de desarrollo

Las pruebas pueden ser efectuadas por el programador (unidad, integración y técnicas) y por el usuario (aceptación).

Se cuenta además con la colaboración de algunos alumnos que conforman equipos de pruebas. De la observación de su desempeño durante la ejecución de las pruebas, se tomaron decisiones de diseño relativas a la interfase hombre máquina. Por ejemplo, un alumno discapacitado visual nos llevó a rediseñar aspectos de accesibilidad de la plataforma.

El requerimiento se cierra con la aceptación de la solución por parte del usuario solicitante, y con la puesta en producción de la solución.

Ciclo de Desarrollo

El ciclo de desarrollo consta generalmente de tres iteraciones:

1. definición conceptual
2. prototipo evolutivo
3. solución definitiva en producción

La primera para formalizar la necesidad a partir de “historias de usuario”, en la primer discusión grupal establecer el alcance y el plan de trabajo, así como el plan de prueba de la solución.

La segunda iteración para revisar el prototipo de la solución y someterlo a los casos del plan de prueba, discutir el diseño y proponer ajustes.

La tercera iteración apunta a conformar la solución y a ponerla en producción.

Encontramos adecuado adoptar para la actividad de desarrollo, las prácticas de la Programación Extrema (XP), dado que de las prácticas encuadradas en métodos ágiles, es la que más se ajusta a las características de nuestros equipos de trabajo.

XP se basa en cuatro valores fundamentales: comunicación, simplicidad, retroalimentación y entusiasmo.

Comunicación

Tal como afirma Fowler “Como los desarrolladores son profesionales capaces en su propia disciplina, necesitan poder trabajar como iguales con otros profesionales de otras disciplinas.

Gran parte de esto, claro está, se debe a la naturaleza del desarrollo adaptable. Como la premisa entera del desarrollo adaptable es que las cosas cambian rápidamente, se necesita un contacto constante para avisar a todos de los cambios.”¹ Las herramientas integradas a la plataforma favorecen la comunicación en cada una de las fases del proyecto, conformando un espacio virtual de trabajo (sincrónico y asincrónico) en el que los usuarios pueden establecer sus requerimientos y someterlos a revisión para refinarlos, definir con el equipo el plan de desarrollo que los satisfará, y comprobar el software producido con el objeto de formalizar la aceptación (o las observaciones de

¹(La Nueva Metodología -<http://www.programacionextrema.org/articulos/newMethodology.es.html>)

tipo correctivo o adaptativo y/o ampliaciones o nuevos req.) en un ambiente controlado, previo a la implantación en el ambiente productivo.

Los usuarios interactúan con los desarrolladores a través de un canal de chat especialmente habilitado al efecto, o se notifican de las novedades o los cambios en los respectivos foros.

De este modo, al forzar la comunicación por medios formales, registramos los requerimientos en un sistema de gestión, que permite identificar y establecer la trazabilidad de cada requerimiento durante la evolución del proyecto.

Simplicidad

Todo el diseño en una fase determinada se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. Esto garantiza mantener el foco de la solución en lo simple, sin contaminarla con otras abstracciones que la vuelvan compleja.

El riesgo está en que se pierda perspectiva del objetivo del sistema total, debido a las limitaciones en la capacidad de abstracción de los usuarios y los programadores. Aquí es donde resulta crucial el rol de Líder-Coordenador, para mantener la visión del proyecto en un nivel de abstracción más alto.

Retroalimentación

La retroalimentación es el núcleo de un entorno de desarrollo iterativo en la definición conceptual la verificación y comprobación son muy importantes para refinar los requerimientos, dado que si no se puede encontrar una manera de probar la solución a un requerimiento, es probable que no se haya entendido o no se pueda solucionar

En las siguientes fases de desarrollo la retroalimentación provee la información necesaria para el progreso de la solución hasta su implementación.

Entusiasmo

El cambio constante y los requerimientos no planeados, plantean un ambiente de desarrollo estimulante donde el desafío de encontrar soluciones simples e ingeniosas forma parte del día a día del desarrollador. Esta característica mantiene equipos de desarrollo ágiles y en constante reinvención.

Uno de los mayores problemas en los equipos de desarrollo formales es el anquilosamiento y burocratización de los mismos, hemos encontrado que lo que muchas veces se plantea como una contra en el desarrollo (requerimientos impredecibles) se puede convertir en una ventaja al tener equipos de desarrollo en constante movimiento.

La capacitación continua es un efecto secundario deseado en este tipo de ambientes, el desarrollador tiene la necesidad de capacitarse formal o informalmente para poder responder al desarrollo cambiante.

Repositorio de Conocimiento

Utilizando los foros y las consultas sobre la información registrada, los desarrolladores aumentan su accesibilidad al conocimiento, posibilitando el desarrollo de mayores potencialidades en un proceso de aprendizaje continuo. Esto disminuye los costos de formación por programador.

Tal como afirma Josh Kerievsky “Si un equipo realmente aprecia los valores de Retroalimentación y Comunicación de la Programación Extrema, será el aprendizaje continuo una consecuencia directa de dichas prácticas”²

El aprendizaje continuo es parte del espíritu de la programación extrema, y los equipos aprovechan la ventaja de los aspectos didácticos que hacen a la utilización de los medios electrónicos de comunicación dispuestos en la plataforma tecnológica.

Luego de cada implementación de una versión mayor del sitio se realizan reuniones para reflexionar y discutir de lo aprendido a partir del desarrollo. Para los temas técnicos, se busca obtener un detalle de deficiencias del equipo en el aspecto técnico y se planifica su solución (capacitación, migración de servidores, etc.), para asuntos de la interacción personal o desarrollo del proceso se realiza una retrospectiva de revisión de fin de proyecto (análisis postmortem), donde se da participación a todo el equipo (desarrolladores, coordinadores, usuarios) para entender:

- Que sucedió
- Que se pudo Aprender
- Qué podría hacerse de otro modo la siguiente vez.

Coincidimos con Kerievsky que “El aprendizaje continuo no es una parte nueva de la Programación Extrema, sino una parte medular, implícita pero no articulada directamente. Es una práctica que puede ayudar a un equipo de Programación Extrema a convertirse pronto en un gran equipo de Programación Extrema”

Separación de ambientes y control de cambios

Para dar soporte a nuestro enfoque metodológico de gestión de desarrollo y mantenimiento, implementamos métodos de control de versiones de manera de poder hacer gestión de los productos desarrollados.

Esto es muy importante porque permite a equipos teleconectados usar un repositorio común de código para realizar las tareas necesarias de desarrollo. Asimismo Organizamos nuestro entorno de soporte para educación a distancia, separando los ambientes de desarrollo, prueba y producción.

Cuando un requerimiento es asignado a un equipo, se transforma en un proyecto, cuyos componentes se identifican separadamente con una etiqueta para el control de cambios (ECC), y se promueven al ambiente de desarrollo para su modificación, analizando previamente si no impactan en desarrollos en ejecución. Así toda pieza de código en modificación cuenta con su correspondiente ECC permitiendo realizar trazabilidad de modificaciones y rastreo de cambios.

Asimismo, durante el período de ejecución de pruebas y corrección de defectos, es muy posible que se requiera modificar componentes que están en el ambiente de prueba y que ya se han comenzado a modificar en el ambiente de desarrollo. Esto se debe a que la velocidad para corregir defectos suele ser mayor a la de desarrollos programados. Para ello es necesario contemplar la realización de los cambios preservando la última versión en desarrollo y replicando el cambio por la incidencia en la nueva versión de desarrollo.

² . <http://industriallogic.com/xp/ContinuousLearning.pdf>

De este modo, a medida que se va iterando, se van refinando los componentes y se van acordando versiones, que se van ramificando y fusionando en familias derivadas de la etiqueta de cambio original.

Los cambios que se van aplicando se mantienen separados en cada ambiente, y se tiene trazabilidad y control pudiendo, de ser necesario, volver atrás un cambio a una versión determinada. Este criterio se aplica no solo al código sino también a las historias de usuario de los requerimientos, y a los casos de prueba relacionados con ellas.

Conclusión

Luego de habernos encontrado con una serie de dificultades y al intentar corregirlas concluimos que debíamos cambiar la metodología para el desarrollo y la programación, en este recorrido coincidimos: Como cita Carlos Reynoso en “Métodos Heterodoxos en Desarrollo de Software” la Documentación de “Microsoft Solutions Framework 3.0 eXtreme Programming (XP)” “Los métodos Ágiles,..., son estrategias de desarrollo de software que promueven prácticas que son adaptativas en vez de predictivas, centradas en la gente o en los equipos, iterativas, orientadas hacia prestaciones y hacia la entrega, de comunicación intensiva, y que requieren que el negocio se involucre en forma directa....las metodologías ágiles están muy alineadas tanto en los principios como en las prácticas para el desarrollo de software en ambientes que requieren un alto grado de adaptabilidad.”

Teniendo en cuenta el ambiente de desarrollo de los sitios de soporte de Educación a Distancia basados en tecnología Web y la particularidad de nuestros equipos de trabajo descritos con anterioridad, aprovechamos las herramientas ya desarrolladas para conformar un espacio virtual de encuentro común y de trabajo integrándolas a un nuevo sistema de gestión de Requerimientos, Proyectos y Cambios, para conformar un repositorio de conocimiento del proyecto. Tendiendo de esta manera a transparentar las practicas de la derecha del cuadro del manifiesto ágil, que naturalmente se utilizan o minimizan, coincidiendo con Martin Fowler en “The New Methodology” “Estos nuevos métodos buscan un justo medio entre ningún proceso y demasiado proceso, proporcionando simplemente suficiente proceso para que el esfuerzo valga la pena. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos”

Aumentando la accesibilidad al conocimiento del grupo de trabajo, posibilitando el desarrollo de mayores potencialidades en un proceso de aprendizaje continuo. Disminuyendo los costos de formación. Tratando de materializar la metodología y la herramienta según refiere Josh Kerievsky “Si un equipo realmente aprecia los valores de la retroalimentación y comunicación de la programación extrema, será el aprendizaje continuo una consecuencia directa de dichas practicas”.

Referencias bibliográficas

- Advanced Development Methods. “Controlled chaos: Living on the Edge”,
<http://www.controlchaos.com/ap.htm>, 1996
- 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 y Dave Thomas. “Agile Manifiesto”.<http://agilemanifesto.org/> – El Manifiesto Ágil.
- Kent Beck. “Interview with Kent Beck and Martin Fowler”. Addison-Wesley,
<http://www.awprofessional.com/articles/article.asp?p=20972&redir=1>, 23 de Marzo de 2001.
- Kent Beck y Martin Fowler. *Planning Extreme Programming*. Reading, Addison Wesley, 2000.
- Lee Copeland. “Developers approach Extreme Programming with caution”. *Computerworld*, p. 7, 22 de Octubre de 2001.
- Martin Fowler. “Is design dead?”. *XP2000 Proceedings*,
<http://www.martinfowler.com/articles/designDead.html>, 2001.
- Microsoft Solutions Framework Process Model, v. 3.1.
<http://www.microsoft.com/technet/itsolutions/techguide/msf/msfpm31.mspix>, 2002.
- Microsoft Solutions Framework – MSF Project Management Discipline.
<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/tandp/innsol/default.asp>, 2002.
- Microsoft Solutions Framework Version 3.0 Overview,
<http://www.microsoft.com/technet/itsolutions/techguide/msf/msfovrvw.mspix>, 2003.
- Joshua Kerievsky, a Senior Consultant with Cutter Consortium's [Agile Software Development & Project Management](#)
- Carlos Reynoso, Revisión técnica de Nicolás Kicillof , Métodos Heterodoxos en Desarrollo de Software
Versión 1.0 – Abril de 2004

Direcciones Web de métodos ágiles

<http://c2.com/cgi/wiki?ExtremeProgrammingRoadmap> – Materiales de XP en el sitio de Ward Cunningham, creador del Wiki Wiki Web.

<http://www.extremeprogramming.org> – Página semi-oficial de XP.

<http://www.jimhighsmith.com> – Páginas de Jim Highsmith y Adaptive Software Development.

<http://www.refactoring.com> – Página de Martin Fowler consagrada a refactorización.