INSTITUTO POLITECNICO NACIONAL

CENTRO DE ESTUDIOS CIENTIFICOS Y TECNOLOGICOS

“JUAN DE DIOS BATIZ”

MATERIA: METODOS AGILES DE PROGRAMACION

PROFESOR: TORRIJOS JAIMES ROBERTO CARLOS

EMPRESA: WRANDE

“Pensemos en Wrande”



ALUMNOS: ***Arenas Carrillo Ricardo Xochipilli***

***Flores Rodríguez Ángel Ariel***

***Mora Guerrero León Sergio***

***Zapata Manilla Carol Michelle***

***TRABAJO: TAREA 4. MANIFIESTO AGIL***

*ESPECIALIDAD: PROGRAMACION*

GRUPO: 6IM7

TURNO: MATUTINO

SEMESTRE: SEXTO

***Introducción***

En la actualidad, las metodologías ágiles se convierten en un modelo para los iniciados en el desarrollo de software, estas metodologías presentan algunas ventajas ante las metodologías pesadas, pero son limitadas por el tamaño del proyecto y el número de programadores que pueden intervenir. Sin embargo, resultan muy atractivas para el desarrollo de aplicaciones en empresas de software que estén iniciando o para el desarrollo de software por módulos, sin descuidar la calidad y garantizando la actualización de la documentación.

***Antecedentes históricos***

En una disciplina naciente la falta de técnicas o metodologías, hicieron evidente el exceso en los tiempos empleados, los sobrecostos, la insatisfacción de los usuarios o clientes; en definitiva, la baja calidad de Scientia et Technica Año 382 XIII, No 34, Mayo de 2007. Universidad Tecnológica de Pereira los programas. Esto fue lo que finalmente desembocó en la llamada crisis del software. Como respuesta a esta crisis, en la segunda mitad de la década de los 60 se popularizó el concepto de ingeniería de software. Al formalizarse una metodología para la elaboración de programas bajo conceptos de ingeniería, esta tarea se transformó en un proceso riguroso propendiendo por el cumplimiento de los cronogramas, los presupuestos, y lo más importante, la satisfacción del cliente. Durante el proceso de evolución de la ingeniería del software, se la ha concebido y definido de diversas formas, al respecto la IEEE (IEEE93), dice: “La Ingeniería de Software consiste en la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software”. En general la ingeniería de software establece los métodos, los principios y las reglas para obtener software confiable de excelente calidad, a partir de un enfoque sistémico.

El 17 de febrero de [2001](https://es.wikipedia.org/wiki/2001) diecisiete críticos de los modelos de mejora del [desarrollo de software](https://es.wikipedia.org/wiki/Desarrollo_de_software) basados en procesos, convocados por [Kent Beck](https://es.wikipedia.org/wiki/Kent_Beck), quien había publicado un par de años antes Extreme Programming Explained, libro en el que exponía una nueva metodología denominada [Extreme Programming](https://es.wikipedia.org/wiki/Extreme_Programming), se reunieron en [Snowbird, Utah](https://es.wikipedia.org/w/index.php?title=Snowbird,_Utah&action=edit&redlink=1" \o "Snowbird, Utah (aún no redactado)) para tratar sobre técnicas y procesos para desarrollar software. En la reunión se acuñó el término “Métodos Ágiles” para definir a los métodos que estaban surgiendo como alternativa a las metodologías formales ([CMMI](https://es.wikipedia.org/wiki/CMMI), [SPICE](https://es.wikipedia.org/wiki/SPICE)) a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.

Los integrantes de la reunión resumieron los principios sobre los que se basan los métodos alternativos en cuatro postulados, lo que ha quedado denominado como Manifiesto Ágil.

Hasta 2005 han sido frecuentes las posturas radicales entre los defensores de los modelos de procesos y los defensores de modelos ágiles, quizá más ocupados en descalificar al otro que en estudiar sus métodos y conocerlos para mejorar los propios.

En el Manifiesto Ágil, firmado por [Kent Beck](https://es.wikipedia.org/wiki/Kent_Beck), [Mike Beedle](https://es.wikipedia.org/w/index.php?title=Mike_Beedle&action=edit&redlink=1), [Arie van Bennekum](https://es.wikipedia.org/w/index.php?title=Arie_van_Bennekum&action=edit&redlink=1" \o "Arie van Bennekum (aún no redactado)), [Alistair Cockburn](https://es.wikipedia.org/w/index.php?title=Alistair_Cockburn&action=edit&redlink=1" \o "Alistair Cockburn (aún no redactado)), [Ward Cunningham](https://es.wikipedia.org/wiki/Ward_Cunningham), [Martin Fowler](https://es.wikipedia.org/wiki/Martin_Fowler), [James Grenning](https://es.wikipedia.org/w/index.php?title=James_Grenning&action=edit&redlink=1), [Jim Highsmith](https://es.wikipedia.org/w/index.php?title=Jim_Highsmith&action=edit&redlink=1" \o "Jim Highsmith (aún no redactado)), [Andrew Hunt](https://es.wikipedia.org/w/index.php?title=Andrew_Hunt&action=edit&redlink=1), [Ron Jeffries](https://es.wikipedia.org/w/index.php?title=Ron_Jeffries&action=edit&redlink=1), [Jon Kern](https://es.wikipedia.org/w/index.php?title=Jon_Kern&action=edit&redlink=1), [Brian Marick](https://es.wikipedia.org/w/index.php?title=Brian_Marick&action=edit&redlink=1), [Robert Cecil Martin](https://es.wikipedia.org/w/index.php?title=Robert_Cecil_Martin&action=edit&redlink=1), [Steve Mellor](https://es.wikipedia.org/w/index.php?title=Steve_Mellor&action=edit&redlink=1), [Ken Schwaber](https://es.wikipedia.org/w/index.php?title=Ken_Schwaber&action=edit&redlink=1), [Jeff Sutherland](https://es.wikipedia.org/w/index.php?title=Jeff_Sutherland&action=edit&redlink=1) y [Dave Thomas](https://es.wikipedia.org/wiki/Dave_Thomas" \o "Dave Thomas), se expone:

1. Individuos e interacciones sobre procesos y herramientas
2. Software funcionando sobre documentación extensiva
3. Colaboración con el cliente sobre negociación contractual
4. Respuesta ante el cambio sobre seguir un plan

***Principios***

1. Nuestra mayor prioridad es satisfacer al cliente  
   mediante la entrega temprana y continua de software  
   con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas   
   tardías del desarrollo. Los procesos Ágiles aprovechan  
   el cambio para proporcionar ventaja competitiva al   
   cliente.
3. Entregamos software funcional frecuentemente, entre dos  
   semanas y dos meses, con preferencia al periodo de   
   tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores  
   trabajamos juntos de forma cotidiana durante todo  
   el proyecto.
5. Los proyectos se desarrollan en torno a individuos   
   motivados. Hay que darles el entorno y el apoyo que   
   necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar   
   información al equipo de desarrollo y entre sus   
   miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de   
   progreso.
8. Los procesos Ágiles promueven el desarrollo   
   sostenible. Los promotores, desarrolladores y usuarios  
   debemos ser capaces de mantener un ritmo constante   
   de forma indefinida.
9. La atención continua a la excelencia técnica y al   
   buen diseño mejora la Agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de  
    trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños  
    emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre  
    cómo ser más efectivo para a continuación ajustar y  
    perfeccionar su comportamiento en consecuencia.

***Valores***

El manifiesto hace énfasis en cuatro valores principales que deben soportar el desarrollo de software:

**a.** **Los individuos e interacciones por encima de los procesos y las herramientas**: para garantizar una mayor productividad, las metodologías ágiles valoran el recurso humano como el principal factor de éxito. Reconocen que contar con recurso humano calificado con capacidades técnicas adecuadas, facilidades para adaptarse al entorno, trabajar en equipo e interactuar convenientemente con el usuario, da mayor garantía de éxito que contar con herramientas y procesos rigurosos. Las metodologías ágiles reconocen que es más importante construir un buen equipo de trabajo que las herramientas y procesos. Procura primero conformar el equipo y que éste defina el entorno más conveniente de acuerdo con las necesidades y las circunstancias.

**b. Software funcionando por encima de la documentación**: los profesionales relacionados con el desarrollo de software, aunque no es su fuerte producir documentos, reconocen su importancia, al igual que reconocen el tiempo y costo de mantener una documentación completa y actualizada. En este sentido, las metodologías ágiles respetan la importancia de la documentación como parte del proceso y del resultado de un proyecto de desarrollo de software, sin embargo, con la misma claridad hacen énfasis en que se deben producir los documentos estrictamente necesarios; los documentos deben ser cortos y limitarse a lo fundamental, dando prioridad al contenido sobre la forma de presentación. La documentación, en las metodologías ágiles procura mecanismos más dinámicos y menos costosos como son la comunicación personal, el trabajo en equipo, la auto documentación y los estándares.

**c. La colaboración del cliente por encima de la negociación del contrato**: clásicamente el usuario o cliente es quien solicita e indica qué debe hacer el software, y espera los resultados de acuerdo con sus exigencias o expectativas, en los plazos establecidos. Con frecuencia las dos partes, cliente y equipo de desarrollo, asumen posiciones distantes, con ingredientes de rivalidad y prevención al punto de tener que dedicar tiempo valioso a la tarea de redactar, depurar y firmar el contrato. En este sentido, y complementando el valor que se da al trabajo en equipo, las metodologías ágiles incluyen de manera directa y comprometida al cliente o usuario en el equipo de trabajo. Es un ingrediente más en el camino al éxito en un proyecto de desarrollo de software. Más que un ambiente de enfrentamiento en el cual las partes buscan su beneficio propio, evadiendo responsabilidades y procurando minimizar sus riesgos, bajo la filosofía de las metodologías ágiles se busca el beneficio común, el del equipo de desarrollo y el del cliente. La participación del cliente debe ser constante, desde el comienzo hasta la culminación del proyecto, y su interacción con el equipo de desarrollo, de excelente calidad. Es el cliente quien sabe qué es lo que necesita o desea, el más indicado para corregir o hacer recomendaciones en cualquier momento del proyecto.

**d. La respuesta al cambio por encima del seguimiento de un plan**: dada la naturaleza cambiante de la tecnología y la dinámica de la sociedad moderna, un proyecto de desarrollo de software se enfrenta con frecuencia a cambios durante su ejecución. Van desde ajustes sencillos en la personalización del software hasta cambios en las leyes, pasando por la aparición de nuevos productos en el mercado, comportamiento de la competencia, nuevas tendencias tecnológicas, etc. En este sentido, las metodologías pesadas con frecuencia caen en la idea de tener todo completo y correctamente definido desde el comienzo. No se cuenta entre sus fortalezas la habilidad para responder a los cambios. Por el contrario, en las metodologías ágiles la planificación no debe ser estricta, puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para los siguientes meses.

***Conclusión***

Las metodologías ágiles son una alternativa interesante para superar las debilidades de las metodologías convencionales, pero, al igual que los computadores no son en sí mismos la solución a los problemas de procesamiento de información, éstas no son la solución a todos los problemas que enfrenta el desarrollo de software. Con el surgimiento de las metodologías ágiles, el concepto de etapa se desvanece dando paso a la idea de actividades, las cuales pueden ser organizadas a comodidad del equipo de trabajo, en paquetes pequeños conservando las mismas labores e identidad de las etapas concebidas en las primeras metodologías.

***Bibliografía***

PRESSMAN, Roger S. Ingeniería del Software: Un enfoque práctico, Quinta edición, 601 páginas, McGraw-Hill, Madrid 2002.

/Dialnet-DelManifiestoAgilSusValoresYPrincipios-4809645.pdf

AgileManifiesto. (2006). Agile Manifiesto. 17/02/17, de AgileManifiesto Sitio web: http://agilemanifesto.org/iso/es/principles.html