



UNAH-VS

Universidad Nacional
Autónoma de Honduras
en el Valle de Sula

Universidad Nacional Autónoma de Honduras en el valle de Sula

TEORÍA DE SISTEMAS

INVESTIGACIÓN METODOLOGÍAS ÁGILES APLICADAS EN LA
INDUSTRIA DEL SOFTWARE.

Catedrático:

Ing. Tania Melissa Pineda Godoy

Alumno:

Jimmy Xavier Triminio Hernández - 20122008197

FEBRERO 2023

Contenido

1	Introducción	2
2	Objetivos	3
3	Investigación Metodologías Ágiles Aplicadas en la Industria del Software.	3
3.1	Metodología Ágil	3
3.2	Principios básicos de la metodología ágil	3
3.3	Valores de la metodología ágil	4
3.3.1	El Proceso de desarrollo ágil	6
3.4	Comparativa entre SDLC y la metodología ágil	6
3.4.1	Implementación mediante Metodologías ágiles	7
4	Conclusiones	8
5	Bibliografías	9

1 Introducción

La industria del software está cada vez más interesada en el uso de metodologías ágiles para desarrollar software de calidad. Estas metodologías ofrecen una serie de beneficios, como el incremento de la productividad, la reducción de costos, la mejora de la calidad del producto y la rapidez con la que se llevan a cabo los proyectos. Esto ha llevado a un aumento en la investigación sobre el uso de metodologías ágiles en la industria del software. En este documento, se presenta una breve introducción a la investigación de metodologías ágiles aplicadas a la industria del software. Se discutirán los conceptos básicos de la metodología ágil, los beneficios y desventajas de su uso, así como los últimos avances en el tema. Se concluirá con una lista de recursos útiles para aquellos interesados en el tema.

2 Objetivos

1. Conocer las habilidades para trabajar de manera ágil en un entorno empresarial.
2. Comprender los principios básicos de la metodología ágil.
3. Conocer los valores de la metodología ágil.
4. Conocer el proceso de la metodología ágil.
5. Aprender cómo implementar técnicas y herramientas para aplicar la metodología en el trabajo.

3 Investigación Metodologías Ágiles Aplicadas en la Industria del Software.

3.1 Metodología Ágil

Metodología Ágil es un conjunto de prácticas y técnicas para la gestión de proyectos que ofrecen rapidez, eficiencia y flexibilidad. Su objetivo principal es aumentar la calidad del producto final, ofreciendo resultados inmediatos y ahorrando tiempo y recursos. Esta metodología se caracteriza por el uso de sprints cortos, donde los equipos trabajan en pequeños incrementos de código para entregar resultados rápidamente. Además, se enfoca en la colaboración, el auto-aprendizaje y la retroalimentación continua. Esto se logra a través del uso de herramientas ágiles como el seguimiento de tareas, reuniones de standup y retrospectivas. Esto permite a los equipos entregar productos de mayor calidad de manera más rápida.

Los métodos ágiles son una colección de metodologías innovadoras para el desarrollo de sistemas las cuales se centran en los usuarios los cuales son en principio el cliente o usuario final, de cada uno de los productos finales, obtenidos de aplicar estas metodologías para desarrollo de *software* especializados.

3.2 Principios básicos de la metodología ágil

Para poder explicar este método es imprescindible ceñirse a estos 12 principios fundamentales:

1. Perseguir las *satisfacción del cliente e informarle periódicamente* del estado del proyecto
2. Los nuevos *cambios y requisitos son bienvenidos* y se valoran como modificaciones positivas
3. La división del trabajo se realiza en fases *fases temporales productivas* divididas en semanas, quincenas, etc
4. Posibilidad de *medir el progreso*
5. La forma de ejecutar los proyectos debe *garantizar en sí misma la continuidad del proyecto* (desarrollo sostenible)
6. *El equipo debe trabajar de forma coordinada y en conjunto*, utilizando el *método Scrum* ¹, como una práctica efectiva y esencial para la correcta organización y desarrollo del trabajo.
7. *Las conversaciones entre los integrantes del equipo y/o cliente deben llevarse a cabo en persona*, para comunicar de forma eficaz los mensajes.
8. Es necesario *infundir motivación y confianza a los miembros* que forman parte del proyecto para obtener procesos exitosos.
9. *Excelencia técnica y buen diseño*. En la metodología ágil, la calidad trabajo y la presentación forman parte del conjunto.
10. Se impone la *ley de la simplicidad*. Las tareas deben ser lo más sencillas posibles. En caso de no poder simplificar se tendrá que dividir en iteraciones para reducir su nivel de complejidad.
11. *Equipos auto-gestionados*. Aunque es necesario que existan una figura que monitorice los equipos de trabajo, éstos deben ser capaces de organizarse por sí mismos.
12. *Adaptación a las circunstancias cambiantes*. Es imprescindible que los profesionales que ejecute los proyectos puedan adaptarse a las distintas circunstancias y modificaciones que puedan surgir durante el proceso.

3.3 Valores de la metodología ágil

El método ágil no solo se basa en el resultado final, sino que también en los valores, principios y prácticas. Los *valores* y *principios* establecidos son esenciales para la programación ágil; estos crean el contexto para la colaboración entre programadores y clientes. La metodología ágil es una metodología de desarrollo de *software* que se basa en valores, principios y prácticas básicas. Los cuatro valores de la metodología ágil son: comunicación, *simpleza*, *retroalimentación* y *valentía*.

¹**Método Scrum** es una técnica de metodología ágil ampliada que ofrece una forma de conectar varios equipos que necesitan trabajar juntos para ofrecer soluciones complejas.

Como a menudo hay tensión entre los que hacen los desarrolladores a corto plazo y los que lo comercialmente deseable a largo plazo, es importante establecer los valores que formarán la base para actuar en conjunto en un proyecto de *software*. Estos valores son los siguientes:

1. Comunicación

En todo esfuerzo humano existe la posibilidad de una mala comunicación. Los proyectos de sistemas que requieren de una constante actualización de una constante actualización y diseño técnico son especialmente propensos a dichos errores. Si sumamos a ello tiempo de entrega ajustados, jerga especializada y el estereotipo de que los programadores prefieren hablar con las máquinas en vez de las personas, terminamos con el potencial de toparnos con serios problemas de comunicación.

2. Simpleza

Cuando trabajamos en un proyecto de desarrollo de *software*, nuestra primera tendencia es abrumarnos con la complejidad y tamaño de la tarea. La *simpleza* para el desarrollo de *software* significa que debemos empezar con las cosas más simples que podamos realizar. Y el valor de la simpleza nos pide hacer las cosas más sencillas hoy, a sabiendas de que tal vez mañana tengamos que cambiarla un poco.

ocurrir en cuestión de segundos, minutos, días, semanas o meses, esto dependiendo de lo que se requiera, de quién se esté comunicando y de lo que se pretende hacer con la retroalimentación.

3. Retroalimentación

La retroalimentación es una forma de control de un sistema. Como sistemas, todas organizaciones usan la planeación y el control de para administrar sus recursos con efectividad. La retroalimentación se recibe desde el interior de la organización y de los entornos exteriores. Cualquier cosa externa a los límites de una organización se considera un entorno. Numerosos entornos con diversos grados de estabilidad constituyen el medio en el que existen las organizaciones.

4. Valentía

El valor de la valentía tiene que ver con un nivel de *confianza* y *confort* que debe existir en el equipo de desarrollo. Significa no tener miedo de desperdiciar una tarde o un día de programación y empezar de nuevo si no si todo está bien. Significa poder estar en contacto con los instintos de uno mismo en relación con lo que funciona y lo que no.

Valentía también significa responder a la retroalimentación concreta, actuando con base en las corazonadas de sus compañeros de equipo cuando ellos piensan que tienen una forma más simple y mejor de obtener su objetivo.

3.3.1 El Proceso de desarrollo ágil

El modelado es una palabra clave en los métodos ágiles. El modelado ágil aprovecha la oportunidad de crear modelos que pueden ser lógicos, como los dibujos de los sistemas, o maquetas de tamaño natural como los prototipos. Un proceso ordinario de modelado ágil podría ser el siguientes:

1. Escuchar las historias de los usuarios por medio del cliente.
2. Dibujar un modelo de flujo de trabajo lógico para apreciar las decisiones de negocios representadas en la historia de un usuario.
3. Crear historias de usuarios con base en el modelado lógico.
4. Desarrollar algunos prototipos de visualización. Para ello hay que mostrar los clientes el tipo de interfaz que tendrán.
5. Usar la retroalimentación de los prototipos y los diagramas del flujo de de trabajo lógico para desarrollar el sistema hasta crear un modelo físico de datos.

3.4 Comparativa entre SDLC y la metodología ágil

La metodología ágil es un enfoque moderno para el desarrollo de *software* ², mientras que el SDLC ³ es un método tradicional. ambas se utilizan para desarrollo y mantener software de manera eficiente.

La principal diferencia entre estas dos Metodologías es el enfoque. La metodología ágil se centra en la adaptación, la iteraciones y la mejora continua, mientras que el SDLC se basa en una planificación detallada, una estructura formal y un proceso escrito. Otro gran diferenciador es que la metodología ágil se basa en equipo auto-organizado, mientras que el SDLC se basa en roles y responsabilidades claramente definidos.

Otras diferencias entre SDLC y la metodología ágil incluyen el tiempo de desarrollo, el control de calidad, el grado de documentación y la estabilidad del producto. La metodología ágil suele tener un tiempo de desarrollo más corto, un control de calidad más alto y una documentación más ligera. Por otro lado, el SDLC suele tener un mayor estabilidad del producto, ya que el proceso es más estructurado y formal.

Los investigadores (*Davis y Naumann, 1999*) desarrollaron un lista de (7) estrategias que pueden mejorar las eficiencia:

²Tesis Doctoral, *Metodología Ágil de Desarrollo de Software Enfocado a Trabajos de Grado en Ingeniería* - autor: Gustavo Armando Rivera Sanchez.

³SDLC:Ciclo de Vida del Desarrollo de Software

1. Reducir el tiempo de los errores de la interfaz.
2. Reducir el tiempo de aprendizaje del proceso y las pérdidas duales de procesamiento.
3. Reducir el tiempo y esfuerzo requeridos para estructurar las tareas y aplicar formatos a las salidas.
4. Reducir la expansión improductiva del trabajo.
5. Reducir el tiempo y costo del almacenamiento, la investigación de datos y del conocimiento.
6. Reducir el tiempo y costos de la comunicación y la coordinación.
7. Reducir las perdidas debido a la sobrecarga humana de información.

3.4.1 Implementación mediante Metodologías ágiles

1. Adoptar la programación en pareja.
2. Creación de prototipos y desarrollo rápido.
3. Fomentar las entregas pequeñas.
4. Limitar el alcance en cada entrega.
5. Permitir un cliente en el sitio.
6. Usar cajas de tiempo (timeboxing).
7. Apegarse a una semana de trabajo de 40 horas.

4 Conclusiones

Hemos visto que tanto los métodos ágiles como la interacción con los programadores, clientes, analistas y humano-computadora son fundamentales para el éxito de un proyecto. Los analistas de sistemas pueden utilizar prototipos para recopilar reacciones de los usuarios, sugerencias, innovaciones y planes de revisión para realizar mejoras al prototipo y modificar los planes del sistema con un mínimo de costo y de interrupciones. Al desarrollar un prototipo, se deben tener en cuenta cuatro alineamientos principales: trabajar en módulos administrables, crear el prototipo con rapidez, modificar el prototipo y hacer énfasis en la interfaz de usuario. Estas prácticas pueden ayudar a garantizar un proceso de desarrollo eficaz y eficiente que resulte en un sistema de calidad.

5 Bibliografías

1. ***Análisis y Diseño de Sistemas***
(*KENDALL & KENDALL*)
octava edición
2. ***Cognodata*** Principios de Metodología Ágil
www.cognodata.com
3. ***Tesis Doctoral***, *Metodología Ágil de Desarrollo de Software Enfocado a Trabajos de Grado en Ingeniería* - autor: **Gustavo Armando Rivera Sanchez**. Tesis Doctoral 2021