# UNIVERSIDAD TECNOLÓGICA NACIONAL

# FACULTAD REGIONAL CÓRDOBA

## INGENIERÍA EN SISTEMAS DE INFORMACIÓN

## Cátedra: Ingeniería de Software

## Trabajo Practico N1

Ciclo lectivo 2020

Curso 4k1- Turno Mañana

GRUPO N° 7

73545. Brito, Carolina Mariel

74904. Gelos, Juan Agustín

76535. Merlo, Luciano Ariel

75413. Monastyrski, Carlos Alberto

77556. Rosales, Gabriel Francisco

Fecha de Entrega: 4/8/2020

# Enunciado

| | |
|---|---|
| Unidad: | Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software |
| Consigna: | Asocie a cada valor del manifiesto ágil el o los principios ágiles que considere que están asociados. |
| Objetivo: | Comprender valores y principios del Manifiesto Ágil y los principios Lean, expuestos en clase, para aplicarlos a ejemplos concretos de gestión de proyectos de software. |
| Propósito: | Familiarizarse con los valores y principios del Manifiesto Ágil. |
| Entradas: | Conceptos teóricos sobre el tema, desarrollados en clase. Manifiesto Ágil.<br>Bibliografía referenciada sobre el tema. |
| Salida: | Comparación de los valores del manifiesto ágil con los principios del manifiesto ágil, y justificación asociada. |
| Instrucciones: | 1. Cada grupo discute individualmente qué valor del manifiesto ágil está asociado con qué principio ágil.<br>2. Cada grupo presenta al resto del curso sus conclusiones, justificando las relaciones presentadas.<br><br>El práctico será evaluado en el aula. |

## Manifiesto Ágil

VALORES

- Individuos e interacciones por sobre procesos y herramientas
- Software funcionando por sobre documentación detallada
- Colaboración por sobre negociación con el cliente
- Responder a cambios por sobre seguir un plan

PRINCIPIOS

- La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes
- Recibir cambios de requerimientos, aun en etapas finales
- Releases frecuentes (2 semanas a un mes)
- Técnicos y no técnicos trabajando juntos TODO el proyecto
- Hacer proyectos con individuos motivados
- El medio de comunicación por excelencia es cara a cara
- La mejor métrica de progreso es la cantidad de software funcionando
- El ritmo de desarrollo es sostenible en el tiempo
- Atención continua a la excelencia técnica
- Simplicidad - Maximización del trabajo no hecho
- Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados
- A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar

http://agilemanifesto.org/iso/es/manifesto.html

## Desarrollo

Valor: Individuos e interacciones por sobre procesos y herramientas

Principios Asociados:

- **Recibir cambios de requerimientos, aun en etapas finales**. Las empresas operan en un entorno global que cambia rápidamente. Es necesario responder ante nuevas oportunidades, al cambio que se produzca en el surgimiento de nuevos productos y servicios competitivos. Ya que el entorno es cambiante es muy difícil derivar un conjunto completo de requerimientos de software estable, los requerimientos iniciales cambian inevitablemente, ya que a los clientes les resulta imposible predecir como un sistema repercutirá en su practica operacional cotidiana. Solo después de desplegarlo, y que el usuario tenga uso y practica del mismo, se podrán aclarar los requerimientos reales. Si no se cambian los requerimientos el software puede quedar obsoleto al momento de entregarse.
- **Técnicos y no técnicos trabajando juntos TODO el proyecto**. Se involucran a los clientes en el proceso de desarrollo para conseguir una rápida retroalimentación para los requerimientos cambiantes. Como el usuario se involucra en el desarrollo y evolución del software, estos pueden proponer cambios y nuevos requerimientos para una nueva versión del software.
- **Hacer proyectos con individuos motivados**. Los RR HH además de estar informados y calificados para el trabajo deben estar motivados. Esto se puede lograr mediante la alineación de los objetivos de crecimiento personal y los objetivos de la empresa. Además, los objetivos grandes e inalcanzables desmotivan al personal, por lo cual es conveniente dividir los proyectos en una sucesión de pequeños objetivos accesibles en tiempo y forma. Es necesario que la personalidad de los miembros del equipo sea adecuada para una participación intensa.
- **El medio de comunicación por excelencia es cara a cara**. Aunque el compromiso que se establece en la comunicación verbal es menor a la escrita, la información que se transmite se asimila de manera más sencilla y se puede notar la comunicación no Verbal de las personas para reconocer su postura con respecto al proyecto.
- **A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar.** El equipo de trabajo puede no seguir un proceso lineal y estricto en el proceso de desarrollo de un sistema, sino que, con el tiempo y a través de un análisis de su funcionamiento hasta el momento, es capaz de ajustar su forma de trabajar y rectificar en los errores que pudo haber cometido, y de la interacción directa con el cliente resolver problemas de requerimientos. Utiliza procesos empíricos por ello de cada entrega obtiene experiencia, la cual les servirá para sucesivas iteraciones.

Valor: Software funcionando por sobre documentación detallada

Principios Asociados:

- **Recibir cambios de requerimientos, aun en etapas finales**. El contino desarrollo del software deberá permitir todas las posibles mejoras, cambios y/o solicitudes que el cliente necesite, antes de su entrega final, debido a que las necesidades del

proyecto o del software mismo así lo requieran. Además, una cuestión importante de los procesos agiles, es que estos aprovechan el cambio para proporcionar ventaja competitiva al cliente, lo cual aumenta el valor agregado del producto antes de ser completamente entregado. Debido a la velocidad de los cambios, la documentación excesiva puede quedar obsoleta, por ello, las prácticas ágiles enfatizan la importancia de escribir un código bien estructurado y destinar el esfuerzo en mejorar el código.

- **La mejor métrica de progreso es la cantidad de software funcionando**. La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor, de un producto funcionando. Es por esto por lo que, a la hora de desarrollar software, la documentación nunca nos debería representar un gasto desproporcional de tiempo que nos lleve a no poder progresar en el desarrollo del software. La documentación no deberá de retrasar ni tampoco impedir el desarrollo rápido del software que estamos construyendo.
- **Simplicidad - Maximización del trabajo no hecho**. Se centra en el arte de maximizar la cantidad de trabajo no realizado. Realizar los cambios sería extremadamente difícil en sistemas en donde existen muchos participantes. Para mantener la simplicidad, requiere trabajo adicional. Bajo presión de fechas de entrega, se vuelve dificultoso, modificaciones deseables en el sistema. Es necesario enfocarse en la simplicidad tanto del software como del proceso de desarrollo. Siempre que sea posible trabajar de manera activa para eliminar la complejidad del sistema. La documentación excesiva queda desactualizada frente a la velocidad de los cambios.
- **Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados.** Es mucho más eficiente en el desarrollo de un software, que el mismo no se limite a los procesos fijos o a una documentación demasiado estricta. La rápida respuesta a los problemas que aparezcan en el desarrollo, serán factores críticos a la hora de llegar a una arquitectura estable y completa. La auto organización aporta una mayor capacidad de reacción por parte del equipo ante situaciones complicadas, mejora la comunicación y el entendimiento entre los miembros, favorece el pensamiento colectivo y un mejor reparto de responsabilidades en cada situación.
- **El ritmo de desarrollo es sostenible en el tiempo.** Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante dentro de un marco de trabajo establecido de forma indefinida en el tiempo. Esto es esencial para la filosofía ágil, ya que un ritmo insostenible sería igual a un funcionamiento incorrecto en la forma de llegar al resultado final (equipo, proyecto, etc.). Desde el inicio del proyecto se debe asignar responsabilidades y tareas de manera que siempre se puedan cumplir. Es necesario mantener actualizado el documento de requerimientos del sistema y eventualmente dedicar un “pico” de documentación donde en vez de producir una nueva versión del sistema, el equipo genera documentación.
- **Atención continua a la excelencia técnica.** La atención continua a la excelencia técnica y al buen diseño incrementan la agilidad. Además de satisfacer los requerimientos del usuario, los aspectos técnicos deben ser excelentes, independientemente de su cantidad y complejidad. La calidad debe ser vista desde dos perspectivas, la del usuario y la del equipo desarrollador. Para el personal técnico resulta evidente que cuanta más calidad tenga el software en cuanto a diseño y estándares de implementación, más rendimiento obtiene en las tareas. Nuevamente, este principio enfatiza la importancia de escribir un código bien estructurado para suplir el problema de la falta de documentación. A partir de un código legible y bien hecho, los IDE’s pueden generar automáticamente documentación entendible.

Valor: Colaboración por sobre negociación con el cliente

Principios Asociados:

- **Recibir cambios de requerimientos, aun en etapas finales**. El software se desarrolla en incrementos, y el cliente nos provee los requerimientos que se incorporarán en cada incremento. Por lo que vemos necesario mantener una comunicación clara, fluida y constante entre el equipo de trabajo y el cliente para que los cambios se vean reflejados en cada incremento del software. Establecer restricciones de costo, alcance y tiempo para determinar la calidad. El cliente debe definir el alcance mientras que el equipo de trabajo tiene que esforzarse por mantener las limitaciones de costo y de tiempo para cumplir con las expectativas del cliente.
- **Técnicos y no técnicos trabajando juntos TODO el proyecto.** Es ambicioso esperar que el cliente defina de manera definitiva todos sus requerimientos desde el comienzo y peor aún depender de ello para adelantar el proyecto. Los cambios en los requerimientos deben asumirse como parte del proceso de maduración del software, debe entenderse que cuando el cliente describe una necesidad lo hace desde su perspectiva de usuario y que sus conocimientos técnicos lo pueden limitar para hacerse entender completamente. Por lo tanto, las novedades en los requerimientos inducen al equipo de desarrollo a preferir los diseños flexibles, lo cual aumenta la satisfacción del cliente y redunda finalmente en beneficio del equipo de desarrollo dada la comodidad en el diagnóstico y ajustes que se requieren en la etapa de mantenimiento.
- **El medio de comunicación por excelencia es cara a cara.** En los donde se produce un equilibrio entre comunicación ascendente y descendente la dirección tiene en cuenta los mensajes explícitos o implícitos que emiten quienes trabajan a sus órdenes y puede aprovechar la creatividad de todos los integrantes del proyecto.
  La existencia de una comunicación fluida es signo de buena integración entre las personas; en estos supuestos, sin duda, se sabe trabajar en equipo y los conflictos interpersonales están bien resueltos.
- **La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes.** Esto es precisamente por qué el cliente tendrá una mejor comprensión de lo que quiere cuando vea el software en funcionamiento. Nosotros recibiremos información (feedback) que podremos utilizar para adaptarlo.
  Siempre y cuando sean suficientes para crear un incremento significativo (software en funcionamiento) preferiremos las iteraciones más cortas. Mas allá de establecer una negociación previa con el cliente, lo mejor es que el mismo vea los cambios que le propone el software y según su visión de este agregar la funcionalidad que el crea conveniente o necesaria para el sistema.

Valor: Responder a cambios por sobre seguir un plan

Principios Asociados:

- **Recibir cambios de requerimientos, aun en etapas finales**. Un enfoque basado en un plan debe poseer un administrador con una visión completa y equilibrada del proyecto, todo lo que debe diseñarse y los procesos de desarrollo. Pero lo mismo si

bien ya es muy difícil, no funciona con los métodos agiles, donde los requerimientos se desarrollan incrementalmente, donde el software se entrega en rápidos incrementos y donde los cambios a los requerimientos y el software son la norma. El desarrollo ágil debe administrarse de forma tal, que se busque el mejor uso del tiempo y los recursos disponibles.

- **Releases frecuentes (2 semanas a un mes).** A medida que el usuario se va familiarizando con el producto surgen nuevos requerimientos que no han sido especificados con anterioridad o propios del uso. Ante esto, los reléase proporcionan una forma para que el cliente tome contacto con el sistema, evalúe si se cumplen sus requerimientos o decida agregar nuevos.
- **Atención continua a la excelencia técnica.** Es mucho más importante y eficiente trabajar según los cambios que surjan, debido a que de esta forma se podrá generar un sistema acorde a los requerimientos pedidos y que conforme al cliente. Si se adopta un plan y que este no respeta los cambios que surjan en su evolución, el software que se crea no será fiel a los requerimientos reales del cliente y puede quedar obsoleto. Por ello, en cada cambio no hay que descuidar la calidad del producto.
- **A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar.** En intervalos regulares, el equipo reflexiona sobre cómo volverse más efectivo, entonces afina y ajusta su comportamiento como corresponde. El equipo de trabajo está todo el tiempo dispuesto a cambiar lo que sea necesario para mejorar. En cada tarea siempre existe la posibilidad de hacerlo mejor la próxima vez.

## Bibliografía


- https://baturamobile.com/blog/los-12-principios-de-manifiesto-agile-parte-ii/
- https://medium.com/valores-y-principio-agiles/valores-y-principios-agiles-dfdefaf2d414
- https://baturamobile.com/blog/los-12-principios-del-manifiesto-agil-parte-i/
- https://proagilist.es/blog/agilidad-y-gestion-agil/los-12-principios-agiles/
- https://obsbusiness.school/int/blog-project-management/metodologias-agiles/metodologia-agile-cuales-son-los-12-principios-de-su-modelo
- https://agilemanifesto.org/iso/es/manifesto.html
- Ian Sommerville - Ingenieria de Sofware Novena Edición. Pearson Educacion, Mexico 2011