

GRUPO 9

PRÁCTICO 1 - Dinámica de Manifiesto Ágil y Lean (Evaluable)

UTN 5/8/2020

INTEGRANTES:

- Martín Sciarra, 69650
- Solís Luna Santiago José, 65105
- Carogana Martín Nahuel, 65252
- Juan cruz Díaz, 65924
- Guadalupe Vocos, 63631
- Luna, Carlos , 52734

Salida:	Comparación de los valores del manifiesto ágil con los principios del manifiesto ágil, y justificación asociada.
Instrucciones:	<ol style="list-style-type: none"> 1. Cada grupo discute individualmente qué valor del manifiesto ágil está asociado con qué principio ágil. 2. Cada grupo presenta al resto del curso sus conclusiones, justificando las relaciones presentadas. <p>El práctico será evaluado en el aula.</p>

Valores del manifiesto ágil	Principio del manifiesto ágil	Justificación
<ul style="list-style-type: none"> • Individuos e interacciones por sobre procesos y herramientas. 	<ul style="list-style-type: none"> • Técnicos y no técnicos trabajando juntos TODO el proyecto. • Hacer proyectos con individuos motivados. • El medio de comunicación por excelencia es cara a cara. • Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados. • A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar. 	<p>El objetivo fundamental es satisfacer al cliente ya que éste es quién decide sobre el proyecto. Los usuarios y sus necesidades son la principal entrada para ir guiando el proceso de desarrollo y fundamental para definir el resultado final, su calidad y funcionalidad.</p> <p>El usuario debe estar en contacto permanente en el desarrollo para ir descubriendo nuevas necesidades, puntos fuertes y débiles del software. Construir junto con el usuario cara a cara nos permite reducir tiempo, ir adaptándose a los cambios que surgen e ir probando las primeras funcionalidades. Reduce enormemente el riesgo de recibir un producto que no es el esperado.</p> <p>Los equipos y su interacción permiten que tengan claras las tareas que debe afrontar cada día. Esto permite que la desviación del objetivo se reduzca al mínimo.</p> <p>El Desarrollo de software es una actividad de base intelectual y las personas somos seres con inteligencia emocional, por lo que la motivación aumenta de manera increíble nuestro potencial de llevar a cabo tareas y producir un software de calidad.</p>

<ul style="list-style-type: none"> • Software funcionando por sobre documentación detallada. 	<ul style="list-style-type: none"> • La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes. • Releases frecuentes. (2 semanas a un mes) . • La mejor métrica de progreso es la cantidad de software funcionando. • Simplicidad - Maximización del trabajo no hecho . • Atención continua a la excelencia técnica. 	<p>El objetivo es tener software funcional a lo largo de todo el proceso de desarrollo, e ir progresando en el mismo lo mas rapido posible, asi manteniendo satisfecho al cliente que puede ver los cambios constantemente, y con frecuencia. Para mantener el software funcionando y los releases constantes se debe evitar realizar trabajo no solicitado, o que no influya en el futuro inmediato del producto, sino ir realizando el mismo a medida que es necesario.</p>
<ul style="list-style-type: none"> • Colaboración por sobre negociación con el cliente. 	<ul style="list-style-type: none"> • La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes. • El medio de comunicación por excelencia es cara a cara. 	<p>Parte fundamental de la colaboración con el cliente, por sobre la negociación es poder interpretar los requisitos de la mejor manera y para ello es fundamental que el trato entre las partes sea cara a cara para poder interpretar no solo lo que dice, si no como lo expresa. Poder introducir al cliente en los release temprano es fundamental para saber si el trabajo que hicimos va acorde con lo que el cliente necesita.</p>
<ul style="list-style-type: none"> • Responder a cambios por sobre seguir un plan. 	<ul style="list-style-type: none"> • El ritmo de desarrollo es sostenible en el tiempo . • Recibir cambios de requerimientos, aun en etapas finales. • Releases frecuentes (2 semanas a un mes) . 	<p>Para poder permitir un desarrollo sostenible en el tiempo es necesaria la capacidad del equipo de responder a cambios para adaptar el plan de trabajo. De esta forma se puede sostener en el tiempo el desarrollo del software y permitir entregas frecuentes.</p> <p>El equipo debe adaptarse a los cambios de requerimientos, aún en etapas finales, para garantizar la máxima satisfacción del cliente y agregar valor a su negocio.</p> <p>Tan importante es poder alcanzar una alta velocidad de desarrollo, como poder mantenerla estable, sino, sólo estaremos cubriendo una necesidad puntual.</p>