

PRÁCTICO 1 - Dinámica de Manifiesto Ágil



Cátedra: Ingeniería de Software

Curso: 4k3

Año: 2021

Grupo: 3

Integrantes:

- Rodrigo Virgolini (60377)
- Lautaro Bustos Raiz (72337)
- Ignacio Villane (62687)
- Benjamin Alexander Gómez Toledo (78928)
- Julian Mansilla (76191)
- Jorge Rodriguez (77106)

Consigna:

Asocie cada uno de los valores del manifiesto ágil con el o los principios ágiles que considere que están relacionados.

Valores	Principios	Justificación
Individuos e interacciones por sobre los procesos y herramientas	El medio de comunicación por excelencia es cara a cara	La relación entre los clientes y el equipo es de forma presencial, generando un vínculo y comodidad de trabajo. Además de trata de que sea equipos pequeños, facilitando lo anterior
	Técnicos y no técnicos trabajando juntos	El equipo involucrado en el proyecto trabajan de manera colaborativa , tantos desarrolladores y clientes, entonces esto genera una sinergia en el mismo
	Hacer proyectos con individuos motivados	Se generar un ambiente donde se le de oportunidades, respaldo y confianza para hacer sus tareas, permitiendo equipos autogestionados
	Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados	Esto implica que la participación y colaboración de todos los miembros genera sinergia.
	A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar	Permite al equipo reflexionar sobre los errores cometidos y como hacer mejoras
	La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes	El cliente es un individuo más y se busca su conformidad en vez de continuar con el proceso de construcción del software.
Software funcionando por sobre documentación detallada	La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes	Permite poder entregar nuevas funcionalidades en pocos periodos de tiempo, priorizando este trabajo por sobre el de documentar.

	Releases frecuentes (2 semanas a un mes)	Realizar entregas periódicas permiten que éstas sean un termómetro de nuestro trabajo, brindándole al cliente versiones del producto nuevas con nuevas funcionalidades.
	La mejor métrica de progreso es la cantidad de software funcionando	El software nos permite medir cuánto es que nuestro equipo va construyendo y queda funcionando correctamente
	Simplicidad - Maximización del trabajo no hecho	Concentrarse en lo más simple que se necesita en el momento actual, así no se agrega trabajo extra. Esto implica priorizar el desarrollo del próximo entregable y únicamente la documentación mínima necesaria para la actual etapa, posponiendo el resto para cuando se necesite.
	El ritmo de desarrollo es sostenible en el tiempo	Para lograr la entrega de software funcional a intervalos regulares de tiempo, el equipo necesita adoptar un ritmo constante de trabajo
	Atención continua a la excelencia técnica	Al tener este enfoque nos centraremos en que el software funcione correctamente
Colaboración por sobre negociación con el cliente	Recibir cambios de requerimientos, aun en etapas finales	Al establecer una comunicación casi constante con el cliente, permite ver nuevos requerimiento aun en etapas finales
	Técnicos y no técnicos trabajando juntos TODO el proyecto	El equipo involucrado en el proyecto trabajan de manera colaborativa , tantos desarrolladores y clientes, entonces esto genera una sinergia en el mismo
	El medio de comunicación por excelencia es cara a cara	La relación entre los clientes y el equipo es de forma presencial, generando un vínculo y comodidad de

		trabajo. Además de trata de que sea equipos pequeños, facilitando lo anterior
	Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados	Esto implica que la participación y colaboración de todos los miembros genera sinergia.
	La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes	Es necesaria la entrega constante de resultados al cliente para obtener feedback y que la colaboración con este sea efectiva
Responder a cambios por sobre seguir un plan	Recibir cambios de requerimientos, aun en etapas finales	Los frameworks que se usan permiten adaptarnos a nuevos requerimientos o cambios que se sumen en cualquier etapa, incluso al final
	Simplicidad - Maximización del trabajo no hecho	El maximizar el trabajo no hecho facilita en la mayoría de casos la introducción de cambios de requerimientos, ya que así minimizamos lo que se verá afectado por esos cambios. Esto en contraposición a por ejemplo realizar trabajo no necesario en el momento siguiendo un proceso definido (ej. documentación) para luego tener que realizar un doble esfuerzo corrigiendo al momento de ingresar cambios por parte del cliente.