



TRABAJO PRÁCTICO N° 1:

Manifiesto Ágil

Cátedra: Ingeniería de Software.

Año: 2021.

Grupo: 13.

Profesores: Covaro, Laura Ines (Adjunto).
Robles, Joaquin Leonel (JTP).
Crespo, María Mickaela (Ayudante 1ra).

Integrantes:

- 80753 - Alemán Fernando Ebert - ebertalem@gmail.com
- 66685 - Alfonzo Paola Janet - janet942009@gmail.com
- 43710 - Andriano Garcia Stella Maris - stella.andriano@gmail.com
- 61123 - Caminos Julieta Daiana - caminosjulietadaiana@gmail.com
- 76150 - Domínguez Ariel - arielod98@gmail.com
- 67404 - Monzon Esteban Nahuel - nahuelmonzon93@gmail.com
- 75147 - Salutto Alexis Leonel - alexissalutto1@gmail.com
- 78713 - Verón Federico Javier - veronfederico45@gmail.com

MANIFIESTO ÁGIL

VALORES

1. Individuos e interacciones por sobre procesos y herramientas
2. Software funcionando por sobre documentación detallada
3. Colaboración por sobre negociación con el cliente
4. Responder a cambios por sobre seguir un plan

PRINCIPIOS

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

1. Individuos e interacciones por sobre procesos y herramientas

5. Hacer proyectos con individuos motivados: la motivación se traduce en calidad y ganas de desarrollar el producto.
6. El medio de comunicación por excelencia es el cara a cara: La comunicación en persona es más eficiente para resolver problemas, ya que la mayoría de gestos que hacemos ayudan mucho.
11. Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados: Con organización temprana y sólida se puede prever cómo actuar para no perder tiempo en un futuro.
12. A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar: La interacción entre los integrantes del equipo es de vital importancia a fin de lograr mejorar en los aspectos que consideren débiles.
4. Técnicos y no técnicos trabajando juntos TODO el proyecto: Interactúan intercambiando ideas y solucionando los problemas que se les presentan entre todos, para satisfacer al cliente eficientemente.

2. Software funcionando por sobre documentación detallada

1. La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes: Buscar satisfacer al cliente y no perder tiempo en documentación. Las entregas tempranas ayudan mucho a percibir la forma de pensar del cliente y sus requerimientos.
3. Releases frecuentes (2 semanas a un mes): Hay que tener prioridad en entregar versiones tempranas y no perder tiempo haciendo documentación que probablemente en un futuro quede obsoleta.
7. La mejor métrica de progreso es la cantidad de software funcionando: Si el software funciona es un indicio de que se está avanzando, y si no se pierde tanto tiempo en documentación se avanza más rápido.
8. El ritmo de desarrollo es sostenible en el tiempo: Se busca mantener un ritmo constante de desarrollo en el equipo de trabajo para generar la sinergia necesaria para mantener un software funcional, además de sostener las entregas al cliente para que se sienta conforme y así poder corregir cambios que puedan surgir.
9. Atención continua a la excelencia técnica: Para mantener un software funcionando nuestro equipo de trabajo debe estar formado por personas competentes técnicamente. Si el equipo desarrolla código, bien diseñado y testeado, se podrá incrementar la funcionalidad y agregar cambios de manera sustentable, sin tener que refactorizar constantemente el trabajo realizado.
10. Simplicidad - Maximización del trabajo no hecho: Realizar el trabajo necesario para que el software funcione, no agregar funcionalidad que quizás sea innecesaria, al menos en ese momento.

3. Colaboración por sobre negociación con el cliente

1. La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes: entregarle al cliente prototipos del sistema funcionando ayuda a probar y ver si es lo que realmente quiere, ayudando a descubrir nuevos requerimientos e incentivando su participación activa.
3. Releases frecuentes (2 semanas a un mes). Porque en cada release se entrega una versión de software funcional donde el cliente puede interactuar y así obtener un feedback más detallado.
4. Técnicos y no técnicos trabajando juntos TODO el proyecto: más allá de un acuerdo contractual se integra al cliente como un miembro del equipo dentro del proyecto, su colaboración nos permitirá una mayor retroalimentación con el negocio durante todo el desarrollo.

4. Responder a cambios por sobre seguir un plan

1. La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes: Con entregas tempranas tenemos respuestas tempranas del cliente, permitiendo tomar decisiones según su experiencia con el producto.
2. Recibir cambios de requerimientos, aun en etapas finales: Las prioridades o necesidades de nuestro cliente pueden cambiar a lo largo del proyecto, ya sea por una oportunidad de negocio o por una nueva tecnología en el mercado. Es importante que nos adaptemos a

esos cambios, y no aferrarnos a un plan ya definido, porque nuestra prioridad es satisfacer al cliente y entregar software que realmente le dé valor al negocio.

12. A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar: En este caso este principio nos enuncia la importancia de una mejora continua, en donde el equipo de trabajo no queda sujeto de forma estricta a un proceso ya definido, sino que a través de auto-evaluaciones regulares se realizan los cambios necesarios para maximizar el desempeño.