

Dinámica de Manifiesto Ágil

79071 - Pieve Roiger, Ignacio
79906 - Cibello, Sofía Florencia
79420 - Donalisio, Juan Pablo
79876 - Spini, Leila Aylén
77749 - Parrucci, Lara
80022 - Gonzalez, Florencia

Valores Ágiles

Individuos e
interacciones

sobre

Software
funcionando

sobre

Colaborar con
el cliente

sobre

Responder al
cambio

sobre

Procesos y
herramientas

Documentación
extensiva

Negociación
Contractual

Seguir un plan

Individuos e interacciones por sobre procesos y herramientas

La prioridad es satisfacer al cliente a través de releases tempranas y frecuentes

La participación del cliente se hace más productiva a medida en que el software está siendo probado, revisado y aprobado constantemente por quien lo requiere y lo va a usar. De esta forma podemos dejar a un lado la desconfianza o insatisfacción producidas en el cliente.

Los responsables del negocio y los desarrolladores trabajan juntos

Considerando que somos un solo equipo con un mismo objetivo en común, las personas cumplen un rol importante. Se debe trabajar en conjunto con el cliente, tomándolo como parte del equipo de trabajo, involucrandolo en el proyecto y de esta forma obteniendo mejores resultados

Los proyectos se desarrollan por medio de individuos motivados en un entorno apropiado

Mantener a los individuos, tanto técnicos como clientes, contentos, cómodos y motivados nos permite tener una mejor productividad, obteniendo mejores resultados.

El medio de comunicación por excelencia es cara a cara

El equipo de trabajo debe apoyarse con un buen sistema de comunicación, ya sea entre los miembros del equipo de desarrollo como entre estos y el cliente. La mejor forma de lograr esto es, hablando personalmente en cuyo caso se evitan intermediarios en el proceso de comunicación.

Los equipos auto-organizados generan mejores arquitecturas, requisitos y diseños

Confiar en que la gente y el equipo de trabajo se pueden autoorganizar, dándoles autonomía a ellos les da confianza para trabajar y genera mejores resultados que siguiendo un plan definido.

A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar

El equipo de trabajo está todo el tiempo dispuesto a cambiar lo que sea necesario para mejorar. De esta forma resulta mejor debido a que existe la posibilidad de ir optimizando el desempeño del equipo de trabajo a través del tiempo.



Software funcionando sobre documentación detallada

Simplicidad - maximizar la cantidad de trabajo no realizado

Se debe evitar realizar trabajo innecesario o soluciones complejas ya que pueden consumir mucho tiempo y retrasar la salida del software funcional.

Release frecuentes (2 semanas - 2 meses)

No debe centrarse tanto en la documentación sino en el desarrollo, para poder anticipar el funcionamiento del producto final, con prototipos previos que ofrecen un 'feedback' del cliente. Esto se debe a que si documentar implica tener menos software funcionando es preferible dejarlo para después.

La mejor métrica de progreso es la cantidad de software funcionando.

Estos prototipos nos brindan información acerca de si se están cumpliendo los requerimientos de la mejor manera posible o como el cliente desea.

El ritmo de desarrollo es sostenible en el tiempo.

Se hacen release cada cierto tiempo, lo que permite que el desarrollo se realice más rápidamente y luego, se intenta mantener ese ritmo para cada release.



Colaboración con el cliente sobre negociación contractual

La prioridad es satisfacer al cliente a través de entregas tempranas y frecuentes

Las entregas tempranas permiten aumentar la confianza del cliente hacia el equipo desarrollador ya que, al ver software de manera más frecuente, este puede hacer cambios o ver que se esté haciendo lo acordado. Evita conflictos o errores que con el tiempo llegan a ser complicados de reparar

El medio de comunicación por excelencia es cara a cara

Negociar sin haber antes hablado claramente qué es lo que busca el cliente o qué es lo que puede entregar el desarrollador está destinado a un trabajo ineficiente y posiblemente con muchos errores. El conversar cara a cara demuestra interés en la conversación así como también ayuda a identificar problemas que surgen espontáneamente, el contacto humano debe estar presente por su capacidad de dar soluciones, por otro lado, se evitan malos entendidos, ya que el cliente puede expresarse más y dar más detalles sobre lo que necesita. De esta forma se obtiene un objetivo común a todos y se genera confianza.

Recibir cambios de requerimientos aun en etapas finales

El cliente es un miembro más del proceso del desarrollo de software, se integra y colabora con el grupo de trabajo. La retroalimentación del cliente de forma continua enriquece el producto final, y permite encontrar aquellas tareas que podrían darle mayor valor al producto y proporcionar ventaja competitiva.

Técnicos y no técnicos trabajando juntos todos el proyecto

Trabajar juntos es la definición de colaboración, y el contemplar tanto técnicos como no técnicos, hace referencia a que tanto el equipo de desarrollo como otros involucrados dentro de los cuales está el cliente trabajan en conjunto. Así, podemos saber lo que el cliente realmente quiere (y ayuda al cliente a reconocer qué es lo que él mismo quiere, también, que muchas veces no sabe).



Responder a los cambios por sobre seguir el plan

Atención continua a la excelencia técnica

El equipo de trabajo debe tener el conocimiento y las habilidades necesarias para que los cambios en el producto puedan darse de una forma rápida y eficaz. De no ser así se verá perjudicado especialmente en fases de mantenimiento creando así una deuda técnica por hacer mal las cosas.

Recibir cambios de requerimientos, aun en etapas finales

En la gestión de proyectos tradicional cualquier cambio en la etapa tardía generalmente significa un aumento del alcance y por lo tanto costes más altos.

En este caso los equipos tienen como objetivo aceptar la incertidumbre y reconocer que incluso un cambio tardío todavía puede tener mucho valor para el cliente final

La mejor métrica de progreso es la cantidad de software funcionando

Podemos tener fechas estimativas de entrega, o sea, un plan a seguir, pero si en esas entregas el software no cumple con las funcionalidades requeridas por el cliente, entonces ese producto no tendrá el mismo valor para el mismo, y menos el hecho de que no cumpla con los requerimientos mínimos necesitados para esa entrega.





Bibliografía



- Material proporcionado por la cátedra




Referencias



- <https://solvingadhoc.com/los-principios-del-manifiesto-agil-y-su-aplicacion/>
- <https://www.iebschool.com/blog/metodologia-agil-agile-scrum/>
- <https://kzi.mx/el-manifiesto-agil-valores-y-principios/>





FIN!

Grupo 1 - 4K1