

## Ingeniería de Software

### Trabajo Práctico N° 1

#### Dinámica de Manifiesto Ágil

Grupo N° 2

Curso: 4K4

#### Integrantes:

- Casas, Gonzalo 81960
- Hernández, Daniel 55591
- Hidalgo, Nicolás 78934
- Martínez, Javier Alejandro 62339

# Consigna

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

## Desarrollo

### 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

## Individuos e interacciones por sobre procesos y herramientas

Principios Asociados:

### **Técnicos y no técnicos trabajando juntos TODO el proyecto**

El foco está en las personas, aquellas involucradas en el proyecto, son las que van a colaborar e interactuar durante el desarrollo del proyecto para construir un producto de software de calidad.

Cada persona se interrelacionan dentro del equipo desde su especialidad y sus conocimientos, en todo momento, y no un proceso donde las actividades marcan roles específicos que deben aparecer en ciertas etapas.

### **Hacer proyectos con individuos motivados**

Para que un equipo trabaje de manera excepcional, se deben desenvolver en un buen ámbito de trabajo, personas desmotivadas difícilmente serán proactivos, esto solo se logra cuando ponemos primero los individuos y sus interacciones por sobre los procesos y herramientas.

### **El medio de comunicación por excelencia es cara a cara**

Se valora la interacción entre personas cara a cara, porque esta filosofía considera que es la mejor forma de comunicar las ideas por sobre el uso exclusivo de las herramientas. La interacción cara a cara nos permite expresarnos de mejor manera, se considera como la forma más eficiente para comunicarse.

### **Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados**

Le damos más importancia al trabajo en conjunto, delegando la responsabilidad de la toma de decisiones a las personas que van a realizar el trabajo. Se autoorganizan, deciden cómo van a trabajar, deciden como equipo qué herramientas y actividades del proceso son las más adecuadas para lograr sus objetivos. Deciden cuál y cómo es la mejor manera de construir el producto a medida que lo conocen.

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

Se permite que las personas decidan cómo trabajar. El equipo se inspecciona para ver qué aspectos deben cambiar, continuar o mejorar de la manera en que construyen software.

## **Software funcionando por sobre documentación detallada**

Principios Asociados:

### **La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes**

Si nuestro principal foco está puesto en ir creando y construyendo un producto de software, desde lo que es prioritario para el cliente, sin intentar conocer todos los detalles al comienzo, podremos ir realizando entregas de alto valor que satisfagan al cliente desde etapas tempranas de un proyecto, de esta manera se postergan tareas, para impedir desperdiciar esfuerzo como el de detallar todas las funcionalidades desde un comienzo o generar modelos en funcionalidades que no son requeridas aún.

### **Releases frecuentes (2 semanas a un mes)**

Una de las variables en los proyectos es el tiempo, y lo que plantea la filosofía ágil es que el tiempo que el equipo entrega software funcionando es una constante, entonces puede el equipo negociar en términos del alcance, según lo que el equipo considera que puede desarrollar. De esta forma logramos también entregar software funcionando (alto valor y calidad) en menor tiempo posible.

### **La mejor métrica de progreso es la cantidad de software funcionando**

La generación del producto esta basada en un conjunto de esfuerzos, y poder determinar cuanto de nuestro esfuerzo se ha materializado en software funcionando, es importante para conocer el desempeño del equipo.

La cantidad de software funcionando nos indica cuán cerca estamos de nuestro objetivo o cuánto hemos recorrido.

Es importante también lograr un equilibrio entre la documentación generada que es parte del software funcionando. Mantener modelos que no aportan valor, detallar demasiado, generar modelos solo por que los indica el proceso son actividades que implican esfuerzos que no se ven reflejados en software funcionando.

### **El ritmo de desarrollo es sostenible en el tiempo**

Cuando en cada iteración el equipo conoce cuánto trabajo puede generar, permite al equipo poder comprometerse a entregar una cantidad de software funcionando, siempre priorizando junto al cliente las funcionalidades que se van a desarrollar y esto permite al equipo conocer los requerimientos sobre los que se van a hacer foco y a partir de ese grupo generar los modelos, especificaciones y gestión necesarios.

### **Atención continua a la excelencia técnica**

El software entregado al cliente debe cumplir con sus expectativas, por eso es fundamental que el software sea de calidad, ya que de otra manera no podríamos cumplir con este valor que indica claramente “Software funcionando”. La entrega de valor tiene relación directa con la calidad.

### **Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados**

El equipo es quien mejor conoce al producto y es por lo que un equipo auto organizado sabrá decidir cuales arquitecturas, diseños y requerimientos son los apropiados para el software, como resultado podemos obtener un software completado por una documentación apropiada a lo largo de todo su ciclo de vida.

## **Colaboración por sobre negociación con el cliente**

Principios Asociados:

### **La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes**

El equipo junto con la colaboración permanente con el cliente identifica el valor y la prioridad, esto se usa como entrada para guiar el desarrollo de entregables. Si esos entregables son porciones pequeñas, de alto valor y prioridad que se logran “Terminar” en poco tiempo el cliente logra satisfacer sus expectativas, aún en las etapas más tempranas de un proyecto.

### **Recibir cambios de requerimientos, aun en etapas finales**

Los cambios de requerimientos existen constantemente. Debemos generar procesos que acepten esos cambios sin implicar grandes impactos. Entrar en una discusión con el cliente por algún cambio que requiera generaría desgaste, descontento y pérdida de tiempo. La colaboración permanente con el cliente permite gestionar el cambio de manera tal que el cliente es capaz de reconocer los esfuerzos necesarios para cada cambio y puede visualizar qué aspecto del software puede llegar a cambiar a medida que se materializa.

### **Releases frecuentes (2 semanas a un mes)**

El equipo se compromete con el cliente a generar entregables en períodos cortos, la colaboración con el cliente se ve reflejada al comienzo para definir un alcance, en medio para lograr un desarrollo del producto y al final donde se obtiene un feedback y comunicación, que luego determinará la planificación de los próximos releases en base a lo que le da valor al negocio.

### **Técnicos y no técnicos trabajando juntos TODO el proyecto**

Un aspecto muy importante para que un proceso basado en la filosofía ágil funcione es que todos los involucrados del proyecto deben participar a lo largo del proyecto de desarrollo software. El cliente debería ser como un participante más en todo el proyecto. Debido a que no se obtiene una descripción detallada al comienzo de los requerimientos, el cliente debe de estar presente para poder colaborar con el equipo en la identificación y especificación de los requerimientos. Es el cliente quien conoce lo que le da valor al producto. De esta manera entre el cliente y el equipo técnico van a ir relevando los requerimientos a medida que avanza el proyecto con el objetivo de no buscar tener un plan fijo.

### **El ritmo de desarrollo es sostenible en el tiempo**

Al trabajar de manera regular, el cliente tiene una expectativa más clara de lo que puede esperar del equipo, y el equipo puede comprometerse con el cliente.

La idea del desarrollo sostenible es que podamos darle al cliente una cierta cantidad de software funcionando cada un determinado intervalo de tiempo, de manera tal que podamos obtener una retroalimentación por parte del cliente y ajustar el alcance para el próximo release para continuar de acuerdo con el tiempo pactado con el cliente.

En conclusión las variables de negociación con el cliente siempre se basan en el alcance del siguiente entregable, el cual tiene como dijimos antes, una expectativa clara de cuánto y cuándo podrá obtenerlo.

### **Simplicidad - Maximización del trabajo no hecho**

No gastar tanto esfuerzo en detalles. No enredarse en la complejidad. Trabajar con objetivos claros y simples nos permite entregar software funcional al cliente de manera eficaz. La colaboración, feedback permanente y frecuente con el cliente nos permite saber a qué darle prioridad en cada iteración, de esta manera no invertimos esfuerzo en requerimientos que no le dan mucho valor al cliente o que pueden cambiar.

## **Responder a cambios por sobre seguir un plan**

Principios Asociados:

**Recibir cambios de requerimientos, aun en etapas finales:**

Lo único que se mantiene constante son los cambios, por lo que generar una planificación detallada desde el inicio no es lo mejor en esas circunstancias. Por eso los equipos que siguen una filosofía ágil deben planificar teniendo en cuenta que los requerimientos cambian.

### **Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados**

Las planificaciones detalladas contaban de un grupo de actividades específicas que cada equipo debía realizar, en cambio los equipos auto organizados determinan qué actividades son las adecuadas para generar un software de calidad.

Se confía en el criterio del equipo para tomar decisiones significativas en vez de seguir un plan de actividades estricto

### **Simplicidad - Maximización del trabajo no hecho**

Una forma de maximizar el trabajo no hecho es modelar sólo los requerimientos que se estén a punto de implementar. De esta forma no implicaría un gran esfuerzo responder a los cambios de requerimientos, puesto que seguramente aún no se han desarrollado.

La planificación del trabajo se puede realizar sobre los ítems sobre los que se trabajará en cada iteración.

### **La mejor métrica de progreso es la cantidad de software funcionando**

Las métricas nos permiten reconocer desviaciones y tomar acciones en consecuencia. Si los equipos son capaces de implementar métricas que le permitan medir la cantidad de software funcionando podrán generar acciones que permitan responder mejor a los cambios.