



Universidad Tecnológica Nacional

Ingeniería en Sistemas de la Información

PRÁCTICO N° 1 - Dinámica de Manifiesto Ágil

Cátedra de Ingeniería de Software

Grupo N°9

Curso: 4K3

Fecha: 23/03/2021

Integrantes:

- | | |
|--------------------------|---------------|
| • Acevedo Hernán | Legajo: 72380 |
| • Aresu Barella, Bruno | Legajo: 73446 |
| • Bardin, Tomás | Legajo: 72783 |
| • Boné, Nicolás | Legajo: 72624 |
| • D'uva, Micaela | Legajo: 74300 |
| • Frattin Juan | Legajo: 71939 |
| • Molina, Matias | Legajo: 75477 |
| • Pereira Duarte, Martin | Legajo: 76367 |

Docentes:

- Covaro, Laura Ines
- Robles, Joaquín Leonel
- Crespo, María Mickaela

Índice

Índice	2
Principios relacionados a cada valor	3
Individuos e interacciones por sobre procesos y herramientas	3
Software funcionando por sobre documentación detallada	4
Colaboración por sobre negociación con el cliente	5
Responder a cambios por sobre seguir un plan	5

Principios relacionados a cada valor

Individuos e interacciones por sobre procesos y herramientas

- **Técnicos y no técnicos trabajando juntos TODO el proyecto:**
En este principio se hace énfasis en la obtención de un equipo bien estructurado, con definiciones bien claras y con un conocimiento en común del trabajo a llevar a cabo. De esta forma conseguimos un equipo que se puede adaptar a diferentes circunstancias empleando procesos en forma de guías.
- **Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados:**
Para lograr el resultado propuesto de manera eficiente es necesaria la conformación de grupos de trabajo autoorganizados. Esto facilitará la comunicación entre los integrantes, generando confianza entre ellos. A partir de la organización propuesta por el grupo se genera un ambiente con personas motivadas para trabajar en diseños, arquitecturas e implementaciones con resultados más óptimos.
- **Hacer proyectos con individuos motivados. Hay que darles el entorno y el apoyo que necesitan y confiarles la ejecución del trabajo:**
La motivación aumenta de manera exponencial nuestro potencial de llevar a cabo tareas donde el intelecto, ya sea lógico o creativo, juega un papel tan importante como en el desarrollo de software. Es uno de los factores más importantes para que el trabajo se desarrolle de la mejor manera posible y la eficiencia se mantenga constante.
- **El medio de comunicación por excelencia es cara a cara :**
La intercomunicación (preferentemente “cara a cara” como declara este principio) nos permite expresar más detalladamente las palabras, permitiendo poner en común una idea o de buscar un alineamiento común, o incluso de poder percibir lo más relevante. Pero para sacar el máximo provecho, al aplicar este principio, se debe tener preparada previamente la reunión y los objetivos que se quieren conseguir.
- **Simplicidad - Maximización del trabajo no hecho:**
Sea cual sea la tarea a realizar, la transmisión de conocimientos debe ser sencilla con el fin de promover la simplicidad, evitaremos abarcar demasiados procesos/herramientas a la vez o aquellos tal vez complejos, y, en cambio avanzaremos poco a poco aportando valor y eliminando grandes esfuerzos que no lo aportan.

Software funcionando por sobre documentación detallada

- **La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes:**
La presentación de releases tempranos permite al equipo conocer los próximos pasos a seguir de acuerdo a la visión del cliente. Determina si se están cumpliendo los requerimientos definidos y le da una perspectiva al cliente de lo que va a recibir cuando el producto sea entregado. Así se logra una aceptación y adaptación del cliente hacia el producto final mucho más sencilla.
- **Releases frecuentes (2 semanas a un mes):** La utilidad de las entregas continuas se basan en que el cliente puede ir visualizando el avance del proyecto lo que se traduce en una constante retroalimentación con el equipo de trabajo, ayudando a alcanzar satisfactoriamente los objetivos.
- **La mejor métrica de progreso es la cantidad de software funcionando:**
Esta relación se debe a que los documentos o métricas más tradicionales tienen menos valor para el desarrollo que la comunicación que se puede lograr con el cliente a través de las presentaciones de los prototipos con software funcionando.
- **Simplicidad - Maximización del trabajo no hecho:**
Resulta de amplia importancia dedicar toda la atención pura y exclusivamente a aquellos requerimientos acordados con el cliente, para así evitar los excesos de documentación y el desarrollo de funcionalidades que no se pidieron. Se busca realizar solo lo que se haya acordado, ni más ni menos.
- **Atención continua a la excelencia técnica:**
Se debe apuntar a la excelencia técnica para que en un futuro el sistema pueda ser escalable además de mantenible dentro de su ciclo de vida. Este principio nos habla de la importancia de la maestría, control y conocimiento de los medios y las herramientas que utilizamos para poder llevar a cabo el proyecto y conseguir los objetivos planteados.
- **El ritmo de desarrollo es sostenible en el tiempo:**
Se relaciona con este valor ya que es necesario mantener cierta periodicidad en los avances del software, permitiendo las entregas reiteradas del mismo. Esto produce un ritmo continuo en el desarrollo del software.

Colaboración por sobre negociación con el cliente

- **La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes:**
Se relaciona con este valor dado que una mayor frecuencia de releases o entregas permite tener un feedback continuo de parte del cliente, permitiendo que este se involucre más en el desarrollo guiándolo mediante sus opciones.
- **Técnicos y no técnicos trabajando juntos todo el proyecto:**
Hace referencia al trabajo conjunto entre el equipo de desarrollo y los stakeholders o responsables del negocio, que se involucran constantemente en el proyecto proporcionando una entrada para mejorar la adaptabilidad del producto que está siendo desarrollado.
- **Hacer proyectos con individuos motivados:**
La motivación no solamente se refiere a quienes construirán el software, sino que también implica al cliente o dueños del negocio. Si quien encarga el proyecto no está interesado en el mismo, y no se involucra ni participa en el desarrollo, y simplemente espera que el equipo realice el trabajo sin comprometerse, entonces las metodologías ágiles no serían la forma adecuada de organizar ese proyecto, ya que el cliente no sería compatible con las mismas.

Responder a cambios por sobre seguir un plan

- **Recibir cambios de requerimientos, aun en etapas finales:**
hace hincapié a que los requerimientos pueden sufrir variaciones durante el desarrollo del producto debido a la constante cambio y evolución que se pueden presentar. Es mejor que los planes se antepongan a la evolución de los requerimientos en vez de seguir un plan original el cual puede padecer algún tipo de transformación o cambio al que estaba sujeto desde un principio
- **A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar:**
Es muy importante que el equipo pueda reunirse a evaluar el avance del sistema para tomar medidas en caso de que existan desvíos en el objetivos o impedimentos detectados (retrospectivas). No solo es importante el hecho de reflexionar, sino el de buscar una mejora efectiva por mínima que sea, además es algo que a todos nos aporta una gran satisfacción a nivel personal y a nivel profesional.
- **El ritmo de desarrollo es sostenible en el tiempo:**
En este principio se hace foco en alcanzar una alta velocidad en el desarrollo de software y de mantenerlo constante en el tiempo, pero durante la vida de un proyecto van surgiendo cambios o variables que no estaban claras o que no estaban definidas al principio durante la definición del plan de proyecto. Entonces debemos tener en cuenta esos cambios y considerarlos en el producto que se está desarrollando pero manteniendo la velocidad de desarrollo, es decir que esos cambios que van surgiendo durante el desarrollo del software no deben afectar a la velocidad de desarrollo con la que viene trabajando el equipo de trabajo.