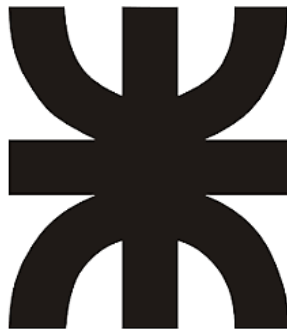


UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional de Córdoba



Ingeniería en Sistemas de Información

Ingeniería de Software

Docentes

ING COVARO, Laura Inés (Adjunto);

ING. MASSANO, María Cecilia (JTP)

ING CRESPO, Mickaela (Ayudante 1ra)

Integrantes

ANGLADA, Martín Legajo: 58159 martinanglada@gmail.com;

CHECA, Nicolás Legajo: 58351;

MARRO, Yanina Legajo: 47594 yani_marro@hotmail.com;

PICOSSI, Matias Legajo: 54941;

TOLEDO, Rodrigo Legajo: 75441.

Grupo N°: 7

Curso: 4K3

ÍNDICE

<i>PRÁCTICO 1 - Dinámica de Manifiesto Ágil (Evaluable)</i>	2
Unidad Nro. 2: Gestión Ágil de Productos de Software	2
Consigna	2
Objetivo	2
Propósito	2
Entradas	2
Salida	2
Instrucciones	2
Manifiesto Ágil	3
Resolución propuesta	4

PRÁCTICO 1 - Dinámica de Manifiesto Ágil (Evaluable)

Unidad Nro. 2: Gestión Ágil de Productos de Software

Consigna:

Asocie a cada valor del manifiesto ágil el o los principios ágiles que considere que están asociados.

Objetivo:

Comprender valores y principios del Manifiesto Ágil, expuestos en clase, para aplicarlos a ejemplos concretos de gestión de proyectos de software.

Propósito:

Familiarizarse con los valores y principios del Manifiesto Ágil.

Entradas:

Conceptos teóricos sobre el tema, desarrollados en clase. Manifiesto Ágil. Bibliografía referenciada sobre el tema.

Salida:

Comparación de los valores del manifiesto ágil con los principios del manifiesto ágil, y justificación asociada.

Instrucciones:

1. Cada grupo discute individualmente qué valor del manifiesto ágil está asociado con qué principio ágil.
2. Cada grupo presenta al resto del curso sus conclusiones, justificando las relaciones presentadas.

El práctico será evaluado en el aula.

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

- A. La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes
- B. Recibir cambios de requerimientos, aun en etapas finales
- C. Releases frecuentes (2 semanas a un mes)
- D. Técnicos y no técnicos trabajando juntos TODO el proyecto
- E. Hacer proyectos con individuos motivados
- F. El medio de comunicación por excelencia es cara a cara
- G. La mejor métrica de progreso es la cantidad de software funcionando
- H. El ritmo de desarrollo es sostenible en el tiempo
- I. Atención continua a la excelencia técnica
- J. Simplicidad - Maximización del trabajo no hecho
- K. Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados
- L. A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar

Resolución propuesta

Valor	Principio	Fundamentación
Colaboración por sobre negociación con el cliente Software funcionando por sobre documentación detallada	La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes	<p>Uno de los objetivos <u>intrínsecos</u> que se busca a la hora de crear un producto de software es lograr la satisfacción del cliente que se manifiesta cuando el producto cumple con los requisitos que aquel solicitó.</p> <p>Creemos que en la construcción del software entregar funcionalidades en iteraciones cortas permite establecer una retroalimentación del cliente lo más pronto posible, ya que al hacerlo genera valor continuo porque permite al equipo de desarrollo y al cliente validar y aprobar lo que se entrega, modificar e incluso detectar cuestiones que serán útiles en las siguientes iteraciones. Este feedback temprano hace que el producto final se termine adaptando a lo que el cliente necesita logrando esa satisfacción.</p>
Responder a cambios por sobre seguir un plan	Recibir cambios de requerimientos, aun en etapas finales	<p>El contexto (económico, social, político, etc.) en el que se desarrolla el producto de software puede cambiar y eso puede ocasionar que los requerimientos cambien.</p> <p>Sabemos que ésto puede ocurrir en cualquier momento durante el proyecto y para que el producto final sea útil para el cliente (esto es que sea competitivo y genere beneficios en el mercado) se debe atacar. Al trabajar en iteraciones cortas y con el feedback constante del cliente, se puede introducir el nuevo requerimiento como una funcionalidad nueva en la siguiente iteración.</p>
Colaboración por sobre negociación con el cliente Software funcionando por sobre documentación detallada	Releases frecuentes (2 semanas a un mes)	<p>Entregar software funcional de forma frecuente permite que el cliente pueda comprobar que el/los requerimientos/s asociados se cumplan. Esta retroalimentación puede ser usada por todos los involucrados del proyecto para ajustar y evaluar la velocidad y capacidad del equipo, la calidad y avance de las entregas, la estimación en las tareas de forma frecuente y manejable para todos los involucrados del proyecto.</p>
Individuos e interacciones por sobre procesos y herramientas	Técnicos y no técnicos trabajando juntos TODO el proyecto	<p>Consideramos los no técnicos como las personas que conocen el negocio y que involucra a los usuarios que van a usar el producto, los stakeholders, etc. y los técnicos, el equipo de desarrolladores, analistas, testers, arquitectos, etc.</p> <p>Creemos que la colaboración conjunta y de forma cotidiana entre ambas partes acrecienta el éxito del proyecto al haber una motivación, interés e involucramiento de ambas partes.</p>
Individuos e interacciones por sobre procesos y herramientas	Hacer proyectos con individuos motivados	<p>La motivación influye en la productividad de las personas. Una <u>persona productiva</u> tiene mayor posibilidad de realizar una tarea de forma eficiente. En vez de instruir a los miembros del proyecto en el seguimiento sistemático de un plan riguroso y poco flexible, creemos que se logra generar mayor motivación en confiar que los miembros del proyecto van a utilizar todas las herramientas que tienen a disposición para realizar su actividad de la mejor manera y encontrar mejores soluciones. Esto genera beneficios en el proyecto a largo plazo porque produce equipos autoorganizados y productivos logrando que el proyecto sea un éxito independientemente del plan y/o proceso que se utilice.</p>

Individuos e interacciones por sobre procesos y herramientas	El medio de comunicación por excelencia es cara a cara	<p>El recurso más importante del proyecto son las personas. Creemos que la comunicación “cara a cara” genera mayor fluidez en el traspaso de información en comparación a que si se lo hiciera por medio de documentos que es más estática.</p> <p>La comunicación cara a cara tiene mayor ventaja ya que permite un intercambio rápido y entendible de la información: se pueden generar debates que enriquezcan el desarrollo del proyecto y del producto; conocer por medio del lenguaje no verbal (gestos, tonos de la voz, etc.) lo que piensa la persona, crear lazos y empatía que fomenten la motivación y colaboración, etc.</p> <p>La documentación sólo debe ser un medio para registrar y consultar lo que se discutió en las reuniones presenciales. </p> <p>6.1</p>
Software funcionando por sobre documentación detallada	La mejor métrica de progreso es la cantidad de software funcionando	<p>El objetivo central de un proyecto de software y lo que origina su concepción es entregar un producto de software funcionando y que cumpla con las expectativas y necesidades del cliente; No hay mejor métrica para indicar si están implementados y funcionan todos los requerimientos comprometidos. La documentación gira en torno al producto y proyecto y si bien puede servir para obtener conclusiones, lo que realmente refleja el éxito del proyecto es el software final funcionando.</p> <p>6.2</p>
Software funcionando por sobre documentación detallada	El ritmo de desarrollo es sostenible en el tiempo	<p>En el desarrollo ágil durante todo el ciclo de vida del proyecto, la actividad en el desarrollo del software de los involucrados debe ser equilibrada para evitar la disminución en la productividad.</p> <p>En cada iteración podemos evaluar la velocidad y capacidad del equipo lo que nos permite planificar la próxima en función de ello, logrando que el ritmo de trabajo sea constante en cada iteración. </p>
Software funcionando por sobre documentación detallada	Atención continua a la excelencia técnica	<p>Se logra <u>aumentar la calidad del software</u> si: existe un equipo con óptimas actitudes técnicas y de análisis; el software desarrollado implementa los patrones correctos de diseño; el código cumple con los estándares de programación y está debidamente documentado; se aplican pruebas, etc.</p> <p>Si prestamos atención a cada iteración de desarrollo de software trabajando lo más eficiente posible, mejoramos la productividad en las siguientes, evitando deudas técnicas. </p>
Colaboración por sobre negociación con el cliente	Simplicidad - Maximización del trabajo no hecho	<p>El cliente tiene un rol central en el proyecto ya que se origina por una necesidad de éste y su participación es clave. El esfuerzo se debe centrar únicamente en satisfacer los requerimientos del cliente sin realizar cosas innecesarias. En el trabajo ágil se busca maximizar los recursos, minimizar el tiempo en tareas que no agregan valor al producto y al mismo tiempo asegurar la calidad. </p>
Individuos e interacciones por sobre procesos y herramientas	Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados	<p>Un equipo ágil con las aptitudes y actitudes de organizarse en conjunto fomenta la capacidad de reaccionar al cambio rápidamente y genera alto rendimiento dentro del proyecto. Es un equipo motivado y por ende su productividad es mayor encontrando mejores soluciones al comunicarse y ayudarse entre sí. </p>
Individuos e interacciones por sobre procesos y herramientas	A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar	<p>En las metodologías ágiles existen ceremonias breves y dinámicas con pautas flexibles y semiestructuradas que permiten evaluar y reflexionar el desenvolvimiento del equipo. No son procesos definidos, sino eventos que enfatizan y que permiten conocer cómo el equipo trabaja, qué habilidades o cosas hay que mejorar, cuáles fueron los problemas en una iteración y extraer mejoras para aplicarlas en la próxima, buscando aprender y mejorar el desempeño del equipo. </p> <p>6.3</p>

Fuente: elaboración grupal

Índice de comentarios

- 6.1 Xq es necesario mantener este registro? Qué valor le da al producto y al usuario?
- 6.2 Que quiere decir obtener conclusiones?
- 6.3 No explica la relacion