



Universidad Tecnológica Nacional
Facultad Regional Córdoba
Ingeniería en Sistemas de información

Cátedra: Ingeniería de Software

Trabajo Práctico:

“Comparación de Enfoques para Gestión del Desarrollo de Productos de Software”

Curso: 4K2

Grupo n°: 9

- 74973 Isaías Avalor Borgiani
- 77629 Francisco Maurino
- 77020 Rodrigo Díaz Mac William
- 64797 Mariana Fagandini

Docentes:

- Adjunto: Meles, Silvia Judith
- JTP: Massano, María Cecilia
- Ayudante 1ro: Robles, Joaquín Leonel

Índice

Índice	2
Introducción	3
Enunciado	4
Desarrollo	5
Conclusión	8
Bibliografía	9

Introducción

A lo largo del presente trabajo práctico, el cual tiene por objetivo la comparación de los diferentes enfoques para la gestión del desarrollo de productos de software, se propone un análisis comparativo de algunas características de los tres enfoques tratados durante el desarrollo de la asignatura. Tomando como principales puntos de comparación los tipos de procesos utilizados, ciclos de vida empleados, gestión de requerimientos, roles y artefactos resultantes, así como las métricas utilizadas a nivel general por cada uno.

Previo a comenzar el análisis, es conveniente situar en un contexto histórico el surgimiento de cada enfoque. En primer lugar, los enfoques tradicionales, surgieron como un intento de superar la “Crisis del software” durante la segunda mitad del siglo XX, donde se optó por la adopción progresiva de modelos y procesos cada vez más rigurosos y prescriptivos, donde se destina gran esfuerzo a controles, planificación y documentación exhaustiva, absorbiendo una importante cantidad de recursos del proyecto, al tiempo que se limita el interés sobre los talentos que conforman los equipos.

En contraposición a esto, el enfoque Ágil surge a principios del siglo XXI con la finalidad de quitar cierta rigurosidad del proceso de desarrollo, apoyándose en 12 principios y 4 valores del llamado “Manifiesto Ágil” que sustenta su filosofía, la cual se centra en la entrega frecuente de valor al cliente con el fin de obtener realimentación temprana por parte de éste.

Por último, la filosofía Lean tiene sus orígenes en un contexto totalmente desligado al software, ya que surgió en la industria automotriz en Japón durante la década del 40, particularmente en Toyota, donde su director Taichii Ohno buscó aplicar una serie de principios que mejoraran la productividad. Estos principios apuntan a eliminar desperdicios durante el proceso, aprender continuamente, entregar valor al cliente, con la máxima calidad posible y en el menor tiempo. Actualmente, su principal representante en la industria del software es Kanban, cuyas características principales se abordarán durante el trabajo.

Para el desarrollo del análisis se optó por los siguientes marcos de trabajo o procesos particulares para representar cada enfoque de gestión: el enfoque tradicional estará representado por el Proceso Unificado de Desarrollo (PUD), el enfoque ágil por SCRUM, y finalmente Lean será representado por Kanban.

Enunciado

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Presentar los diferentes enfoques para la gestión del desarrollo de productos de software, considerando los enfoques de gestión basada en procesos definidos (Tradicional), Scrum y Kanban y realizar una tabla comparativa que explique cada enfoque.
Objetivo:	<p>Que el estudiante sea capaz de abordar la temática, no solo a partir de lo brindado por el docente durante las clases, sino también a través de investigación bibliográfica.</p> <p>Que consiga identificar aspectos que puedan ser comparados para obtener una decisión acertada en la elección de un determinado enfoque de gestión, en el momento en que lo necesite.</p>
Propósito:	Aplicar los conceptos de Gestión del Desarrollo de Productos de Software basados en procesos definidos y basados en procesos empíricos
Entradas:	<p>Conceptos teóricos sobre los temas desarrollados en clase.</p> <p>Bibliografía sobre el tema.</p> <p>Artículos científicos de interés.</p>
Salida:	<p>Se evaluará que:</p> <ul style="list-style-type: none"> • Introduzca brevemente el trabajo a presentar y los enfoques de gestión con la información de interés. • Realice un cuadro comparativo de los diferentes aspectos a contrastar. • Presente una conclusión de elaboración propia del grupo en relación al tema abordado.
Instrucciones:	<ul style="list-style-type: none"> • En grupos trabajarán la consigna. • Preparan un informe que contenga el trabajo • Subirán en el aula virtual en la sección indicada por el docente
Observaciones:	Deben referenciar el material bibliográfico utilizado.

Desarrollo

Características	Enfoque		
	Tradicional	Ágil (SCRUM)	Lean (Kanban)
Tipo de proceso	Está basado en procesos definidos. Los mismos plantean pasos a reproducir para obtener ciertos resultados esperados.	Empírico. Se basa en un proceso de aprendizaje, donde se inspecciona y se adapta regularmente. A pesar de esto, establece reglas y restricciones sobre cómo abordar el proceso, las cuales no pueden modificarse o pasarse por alto.	Al igual que Scrum, Kanban trabaja con Procesos Empíricos. Teniendo un enfoque para el cambio incremental y evolutivo. Aunque posee menor cantidad de reglas y restricciones que SCRUM
Ciclo de vida	Depende del proceso específico que se implemente podrían existir ciclos de vida secuenciales, iterativos, recursivos, etc El PUD (Proceso Unificado de Desarrollo) particularmente implementa ciclos iterativos e incrementales.	Requiere el uso de ciclos iterativos e incrementales para su correcto desarrollo. Se plantean intervalos regulares de tiempos denominados "sprints", en los cuales se busca generar un incremento de producto para el cliente al finalizar.	Ciclo de vida incremental, no necesariamente iterativo. La limitación de trabajo apunta a la cantidad de trabajo en progreso simultáneamente, pero no se exigen intervalos de tiempo en particular.
Formas de organización y seguimiento del proyecto	Se divide el flujo de trabajo en diferentes "workflows", con sus actividades específicas (en el caso de PUD se llevan a cabo durante 4 fases, con distinto énfasis).	Se utilizan diferentes ceremonias (Daily Meeting, Sprint Planning, Product Backlog Refinement, Sprint Retrospective y Sprint Review) para planeamiento, seguimiento del avance del proyecto, perfeccionamiento del proceso y del equipo.	Kanban no prescribe ceremonias o reuniones, la visibilización del proyecto se logra con el tablero Kanban, el cual es persistente a lo largo de todo el proyecto. Se simboliza el flujo de trabajo con diferentes columnas en donde se sitúan las tareas pendientes y en curso, pero nunca se reinicia o se borra el tablero.

Momento de relevamiento y especificación de requerimientos	Es necesario definir los requerimientos de antemano, al inicio del proyecto y con la mayor exactitud posible, para poder realizar las estimaciones necesarias. En el PUD esto ocurre durante el workflow de requerimientos	Utiliza el enfoque “just in time”, los requerimientos no se especifican de antemano, sino que se van encontrando a medida que el proyecto avanza y se obtiene realimentación. Como mínimo se necesitan contar con las características para iniciar el primer sprint.	No define un momento específico para la captura de requerimientos. Pueden surgir nuevos requerimientos producto del feedback obtenido. Los mismos simplemente se agregan al tablero.
Gestión de los requerimientos	Definición y documentación exhaustiva de los requerimientos. En el caso de PUD los mismos son plasmados a través de la ERS (especificación de requerimientos de software)	Requerimientos documentados de forma más ligera, mantenidos y priorizados en el Product Backlog, el cual es de naturaleza dinámica y cambia a lo largo del proyecto.	Se pueden agregar requerimientos durante el avance del proyecto siempre y cuando lo permita el límite de la cola “Pendiente”. A pesar de que se expresan con algún tipo de artefacto (ej: caso de uso), se los suele desglosar para su realización, representandolos con tarjetas, que pueden tener diferentes significados según su color o algún criterio. Se suele utilizar un product backlog para apilar los requerimientos
Cambios en los requerimientos	Los requerimientos son tratados de forma rígida durante el proyecto. No es factible aceptar cambios en etapas tardías, ya que si estos se modificaran, el software resultante no sería el solicitado inicialmente (escapa del alcance pactado).	Se admite la incorporación de nuevos requerimientos e incluso desestimar algunos previamente planteados, aunque no se permiten cambios durante el desarrollo de un sprint.	Los cambios pueden realizarse en cualquier momento, simplemente es necesario plasmar las tareas nuevas a realizar para satisfacer dichas modificaciones.

Roles y equipo de trabajo	Integrantes especializados con roles específicos. Grupos grandes y distribuidos en las diferentes áreas (Análisis, Diseño y Arquitectura, Desarrollo, Base de Datos, Testing).	Integrantes multidisciplinarios y auto-organizados con un rol específico de Scrum (Team member, Scrum master, Product Owner). Grupos de 5 a 9 personas.	Equipos auto gestionados. No se requieren roles específicos y los integrantes pueden ser tanto especializados como multidisciplinarios. En la práctica han surgido dos roles no como cargos, sino por el propósito de los mismos: el gestor de peticiones de servicio y el gestor de prestación de servicio.
Métricas	Existen 3 tipos de métricas distintas: de proceso, de proyecto y de producto. Dependen generalmente del tipo de proceso que se adopta, y de cada implementación en particular. Las métricas "básicas" podrían ser el tamaño del producto, cantidad de líneas de código, el esfuerzo, el tiempo y la cantidad de defectos, entre otras.	Velocidad (que tantos SP se completaron en el sprint/s anterior/es), Capacidad (cuántas horas se disponen, medidas en horas reales o SP) y el RTF (Running Tested Features, que mide la cantidad de características desarrolladas).	Se utiliza el "lead time" o tiempo de entrega (vista del cliente), el "cycle time" o tiempo de ciclo (vista interna), el "touch time" o tiempo de tocado y la eficiencia del ciclo de proceso (tiempo de tocado/tiempo total)
Artefactos	Mayor cantidad de artefactos, son específicos de cada workflow de trabajo. Por mencionar algunos: ERS, Modelo de casos de uso, casos de prueba, manuales de usuario, etc.	Se generan 3 artefactos diferentes. El "product backlog", el "sprint backlog", y el incremento de producto.	No hay artefactos predefinidos en Kanban. En parte se debe a que no es un proceso de desarrollo de software, sino un método para introducir cambios al mismo.

Conclusión

Tras la realización del presente trabajo fue posible interiorizarse en los aspectos claves que identifican a cada uno de los enfoques estudiados, lo que permitió profundizar el aprendizaje sobre los mismos producto de la investigación y el posterior análisis que se llevó a cabo.

De dicho análisis surge una de las diferencias más relevantes entre los enfoques, la cual radica en el tipo de proceso sobre el cual se aplican; por un lado, el enfoque tradicional fue concebido para aplicarse en procesos definidos, altamente predecibles y de bajo dinamismo. Por otra parte, los enfoques Lean y Ágil resultan más efectivos sobre procesos empíricos los cuales están caracterizados por un alto grado de variabilidad y escasa predictibilidad, donde el perfeccionamiento del proceso se basa principalmente en la experiencia obtenida.

Cabe destacar la notoria diferencia que existe entre la metodología tradicional respecto de Ágil y Lean, en el sentido de que la primera involucra un marco de restricciones y exigencias muy estricto respecto de la planificación y la documentación durante el proyecto, mientras que los últimos dos enfoques presentan un tratamiento mucho más liviano y menos exigente respecto de dichos aspectos, poniendo el foco de esfuerzos en la entrega frecuente de valor al cliente y su satisfacción. Aunque a pesar de las grandes similitudes entre estos dos enfoques, una de las diferencias más notables entre los representantes elegidos es que SCRUM es, en general, bastante más prescriptivo que Kanban e impone una mayor cantidad de restricciones, mientras que este último deja más libertad a la hora de implementar la metodología.

A modo de conclusión, es posible afirmar que los tres enfoques presentan debilidades y fortalezas que pueden maximizarse o minimizarse según las características del proyecto en particular que se esté gestionando y de la implementación concreta del enfoque (PUD, RUP, SCRUM, Kanban). Por lo tanto, a la hora de decidir qué enfoque es el más apropiado, resulta de vital importancia tener un conocimiento claro acerca del proyecto a gestionar y de los factores que influyen en él. Además es necesario tener en cuenta las características de cada equipo en particular para decidir qué enfoque se ajusta mejor a las capacidades de los miembros.

Bibliografía

<https://openwebinars.net/blog/diferencias-scrum-kanban/>

<https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

<https://ora.pm/es/blog/scrum-vs-kanban-vs-scrumban>

<https://cristinaramosvega.com/semajanzas-diferencias-kanban-scrum/>

<https://www2.deloitte.com/es/es/pages/technology/articles/kanban-vs-scrum.html>

<https://www2.deloitte.com/es/es/pages/technology/articles/que-es-metodologia-kanban.html#:~:text=Cuando%20se%20utiliza%20para%20el,las%20diferentes%20etapas%20del%20proceso.&text=El%20principio%20de%20tr%C3%A1s%20del%20Kanban,que%20tiene%20un%20rendimiento%20limitado>

<https://dialnet.unirioja.es/descarga/articulo/6777227.pdf>

Essential Kanban Condensed - David J. Anderson & Andy Carmichael.

Kanban and Scrum, making most of both - Henrik Kniberg & Mattias Skarin.