



Universidad Tecnológica Nacional
Facultad Regional de Córdoba
Ingeniería en Sistemas de Información
Cátedra de Ingeniería de Software

Profesora: Ing. MELES, Silvia Judith.

Profesor: Ing. ROBLES, Joaquín Leonel.

Profesora: Ing. CRESPO, María Mickaela.

<p>PRÁCTICO 14 COMPARACIÓN DE ENFOQUES PARA LA GESTIÓN DE DESARROLLO DE PRODUCTOS DE SOFTWARE</p>
--

CURSO: 4K1

Grupo N°: 8

Carrizo, Lucas. Legajo: 72818.

Castellina, Bruno. Legajo: 72193.

Lerda Venchiarutti, Martín. Legajo: 71655

Lozada Bonelli, Juan. Legajo: 72665

Fecha de Entrega: 4 de noviembre de 2020

Introducción

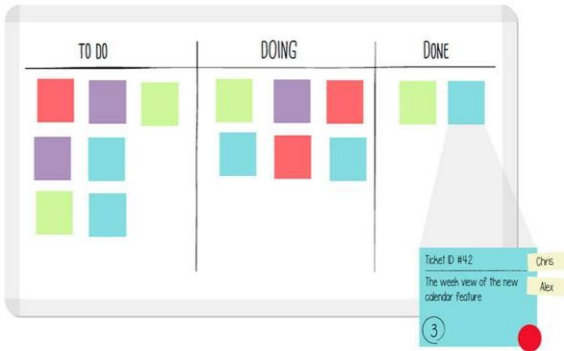
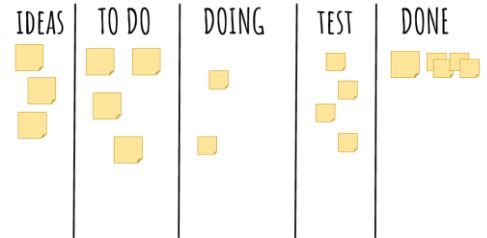
En el presente trabajo práctico se realizará un abordaje de distintas metodologías utilizadas para la gestión del desarrollo de productos de software: el Proceso Unificado de Desarrollo como metodología tradicional, y SCRUM y Kanban como metodologías ágiles.

Tomando las distintas características que cada una posee, con sus alcances, aplicaciones y limitaciones, se las comparará sistemáticamente en base a un conjunto de criterios sobre la que todas tienen aspectos en común o difieren, con el objetivo de evaluar las ventajas y desventajas de cada una.

Desarrollo

<div>Enfoque de gestión</div> <div>Característica</div>	Tradicional (PUD)	Scrum	Kanban
Roles	Roles específicos para cada tarea.	Roles que permiten organizar el equipo de trabajo (Scrum Master, Development Team, Product Owner)	No hay roles obligatorios predefinidos, por lo que no se crean nuevas posiciones en la organización. En la práctica se agregaron los roles Service Request Manager y Service Delivery Manager. Utiliza el mismo modelo organizacional de la empresa, buscando una estructura flexible y poco jerarquizada.
Reuniones	Reuniones no programadas (no están definidas en el proceso)	Reuniones programadas, con objetivos y tiempos definidos (Sprint Planning, Sprint Review, Sprint Retrospective, Daily Scrum, Product Backlog Refinement)	Reuniones no prescriptas (no están definidas en el enfoque).
Artefactos	Se definen artefactos por iteración, los cuales forman parte o son finalmente el producto desarrollado	Artefacto por sprint, por producto y de alcance variable. Estos artefactos son de carácter organizativo del proceso	No posee artefactos predefinidos
Documentación	Maneja una gran cantidad de documentación por cada uno de los artefactos y tareas en todo el ciclo de vida.	Al ser una metodología ágil, se centra en documentar lo mínimo e indispensable, centrando todo el esfuerzo en el desarrollo del producto.	Al ser una metodología ágil, se centra en documentar lo mínimo e indispensable, centrando todo el esfuerzo en el desarrollo del producto

Asignación de tareas	Las tareas a realizar son asignadas directamente al rol que le corresponda, debido a que existen roles bien definidos con tareas especificadas.	Equipos autoorganizados que se asignan las tareas del sprint (especificadas en el Sprint Backlog) en base a las capacidades de sus miembros. No siempre el mismo integrante realiza las mismas actividades.	En base a los eventos a cumplir, cada miembro del equipo definirá sus propias tareas a realizar. La selección de tareas debe realizarse de tal forma que exista poca probabilidad de que ocurran cuellos de botella u obstáculos de ejecución.
Organización del trabajo en el tiempo	El proceso se divide en 4 fases principales (inicio, elaboración, construcción y transición) dentro de las cuales ocurren iteraciones en las que se desarrollan actividades, en mayor o menor proporción, de cada workflow o flujo de trabajo.	Promueve un flujo de trabajo basado en iteraciones o ciclos (sprints) de los que surgen entregas parciales del producto. Los equipos se organizan en sprints de un tiempo definido (sin posibilidad de cambiar su duración), dentro de los cuales se distribuirán las tareas	El trabajo es organizado en base a los eventos, los cuales no tienen una restricción fija de tiempo, sino que se centra en que el flujo de trabajo sea continuo. La única restricción es la cantidad máxima de elementos dentro de cada etapa del WIP.
Gestión de Requerimientos	Los cambios en los requerimientos pueden ser asimilados durante cualquier iteración, siempre y cuando estos cambios no sean significativos o tengan un alcance moderado	Dentro de un sprint, no es recomendable que los requerimientos sufran modificaciones, ya que la adaptación al cambio, sin importar el alcance de dicho cambio, se da una vez finalizado cada sprint.	Durante el transcurso del evento, los nuevos cambios se traducen en modificaciones de tareas definidas anteriormente, los cuales no tendrán impacto más allá de esas tareas
Comunicación	La comunicación con el cliente se da en las primeras fases del proceso y va disminuyendo a lo largo del tiempo. Se va perdiendo la capacidad de	Se realizan reuniones diarias con el equipo donde puede estar además el Product Owner, por lo que la comunicación es continua, fluida y se genera una gran retroalimentación con el cliente y entre los integrantes del	Si bien el enfoque no define nada específico, es recomendable realizar reuniones de manera frecuente para mantener un avance del proyecto y una comunicación continua con el cliente y

	retroalimentación.	equipo. Al finalizar cada sprint se realiza una revisión de tareas con el Product Owner y un análisis a nivel equipo de los resultados del sprint.	dentro del equipo.
Herramientas visuales	No utiliza herramientas visuales para la gestión del desarrollo.	<p>Scrum hace uso de un tablero simple, con los ítems del Product Backlog definidos para el sprint y las actividades que forman parte de cada uno. En el tablero se pueden ver las actividades a realizar, en progreso y realizadas. Una vez que el sprint finaliza el tablero se descarta y se construye uno nuevo para el siguiente sprint.</p> 	<p>El enfoque se centra en el Tablero Kanban en el cual existen tarjetas con cada una de las tareas a realizar, divididas en base al estado en que se encuentren, pudiendo agregar el rostro del encargado en cada tarea. Este tablero se mantiene a lo largo del proyecto, y es en él donde se ve el flujo continuo de las tareas y las limitaciones de cada etapa.</p> 

Conclusión

Una vez finalizado el presente trabajo práctico, podemos realizar una conclusión a nivel global de cada uno de los conceptos abordados. En principio, la comparación no fue realizada únicamente con los enfoques vistos en la cátedra sino que también fueron analizados otros enfoques vistos en la cátedra de análisis y diseño de sistemas.

Los aspectos que tuvimos en cuenta para realizar la comparación permitieron demostrar los aspectos claves de cada uno de los enfoques, sus fortalezas y debilidades y las situaciones en la que es más conveniente utilizarlos. Si bien las metodologías tradicionales las abordamos teniendo como referencia el proceso unificado de desarrollo (visto en cátedras de años anteriores), estamos en conocimiento de la existencia de otras con características diferentes pero que no suelen ser las más utilizadas, por lo cual decidimos considerar principalmente el PUD.

Que una empresa o equipo de desarrollo de software utilice cada una de estas metodologías depende de la forma de trabajar, de la estructura de la empresa, los requisitos del software y el tipo de cliente. En principio, teniendo en cuenta las características de las metodologías tradicionales, es recomendable utilizarlas actualmente para la gestión de proyectos pequeños y bien definidos, con pocas variables, acontecimientos previsibles y un trabajo con alcance limitado. Por otro lado, en aquellos proyectos donde se requiera la entrega gradual de funcionalidad a los clientes, con una generación de mayor valor agregado proveniente del ajuste a sus necesidades en todo momento, es mayormente conveniente utilizar metodologías ágiles como Scrum y Kanban que permitan lograr dichos objetivos más eficaz y eficientemente. Si bien Scrum y Kanban proponen diferentes formas de lograr los objetivos, ambos tienden finalmente al principio de “lograr el mayor valor agregado para el cliente”

Bibliografía

- Material brindado por la cátedra de Diseño de Sistemas, UTN FRC.
- Scrum Guide. Recuperado de <https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>
- Kanban y Scrum – Obteniendo lo mejor de ambos. Recuperado de <https://uv.frc.utn.edu.ar/mod/resource/view.php?id=108093>
- Scrum y Kanban, ¿qué me ofrece cada metodología?. Recuperado de <https://obsbusiness.school/es/blog-project-management/scrum/scrum-y-kanban-que-me-ofrece-cada-metodologia>
- Scrum vs Kanban vs Scumban. Recuperado de <https://ora.pm/es/blog/scrum-vs-kanban-vs-scrumban>
- Gestión de proyectos informáticos tradicional o ágil. Recuperado de <https://obsbusiness.school/es/blog-project-management/metodologia-agile/gestion-de-proyectos-informaticos-tradicional-o-agil>
- Primeros pasos con Kanban. Recuperado de <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos>