



Universidad Nacional de La Matanza

ESCUELA DE FORMACIÓN CONTINUA

Licenciatura en Gestión de Tecnologías
Ingeniería de Software

**Trabajo Práctico 1: Investigación de Agile Comparación
Metodología Spotify vs Scrum.**

PROFESORES:

ING. ROSA SANABRIA

ING. FLAVIO GARRIDO

1° CUATRIMESTRE - AÑO 2021

GRUPO	APELLIDO Y NOMBRE	DNI	NOTA
	Alonso Leandro	41.693.526	
	Aquino Jannett	95.893.314	
	Castillo Emiliano	40.187.592	
	Vazquez Tomás	43.735.806	

Metodologías Ágiles y su evolución

Objetivo del TP:

Que el alumno identifique las diferencias, ventajas, desventajas entre las dos metodologías ágiles.

Material de estudio necesario:

Investigación

Utilizar la información de Agile provista por la cátedra e investigación propia.

Metodología:

La idea es realizar una investigación sobre las dos metodologías, actividades en común, ventajas y desventajas de cada una.

El formato del trabajo es libre, pero debe entregarse en PDF.

Introducción

Antes de adentrarnos a abarcar sobre las metodologías ágiles Spotify y Scrum, creemos que es muy necesario recordar el concepto "Metodología Ágil".

Por esto mismo, la primera pregunta que nos hacemos es: ¿Qué es una metodología ágil?

La metodología ágil es un enfoque de desarrollo de software que se basa en la colaboración, la adaptabilidad y la entrega continua de productos de calidad.

Se caracteriza por la división del trabajo en iteraciones cortas llamadas "sprints", en las que se desarrollan y entregan funcionalidades en un período de tiempo determinado.

Además, fomenta la comunicación constante entre los miembros del equipo, la flexibilidad para adaptarse a cambios en los requisitos del proyecto y la entrega de valor al cliente de forma temprana y frecuente.

Ahora bien, ¿En qué se diferencia del resto de metodologías existentes?

Enfoque en la adaptabilidad: La metodología ágil se basa en la capacidad de adaptarse a los cambios y requerimientos del cliente a lo largo del proyecto, mientras que las metodologías tradicionales tienden a tener un enfoque más rígido y poco flexible.

Colaboración y comunicación constante: En ágil, se fomenta la comunicación continua entre los miembros del equipo y con el cliente, favoreciendo la colaboración y la toma de

decisiones conjuntas. En cambio, en las metodologías tradicionales la comunicación suele ser más jerárquica y formal.

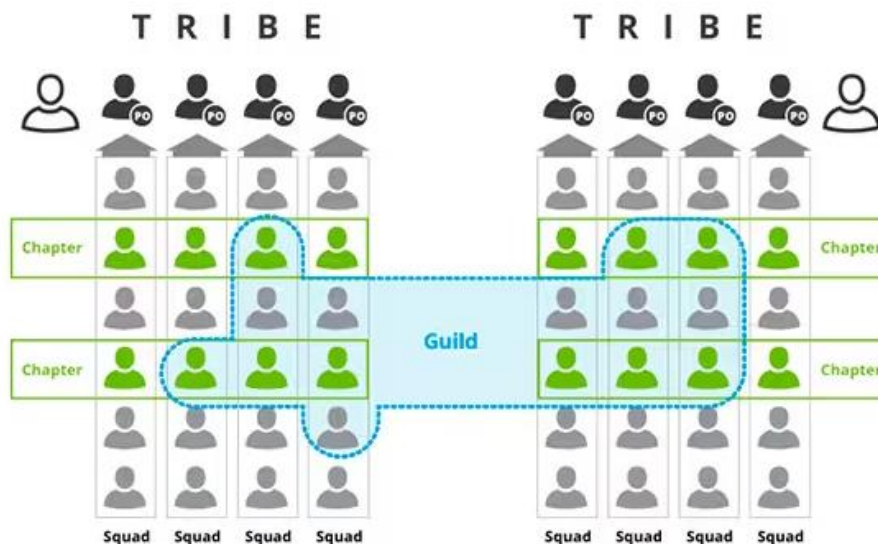
Entrega incremental: Agile se basa en la entrega continua de funcionalidades al cliente de forma incremental, lo que permite obtener retroalimentación temprana y ajustar el producto en base a ella. En contraste, las metodologías tradicionales tienden a entregar el producto final al final del proceso de desarrollo.

Auto-organización de equipos: En agile, se fomenta la auto-organización de los equipos, lo que significa que los miembros tienen autonomía para tomar decisiones y planificar su trabajo. En las metodologías tradicionales, suele haber una mayor jerarquía y control sobre las tareas y responsabilidades de cada miembro.

Estas son solo algunas de las diferencias fundamentales entre la metodología agile y otras metodologías tradicionales de desarrollo de software. Cada enfoque tiene sus ventajas y desventajas, y la elección de una u otra dependerá de las necesidades y características específicas de cada proyecto.

Metodología Spotify

La metodología Spotify es un enfoque utilizado por equipos de desarrollo de software, inspirado en las prácticas que emplea la empresa Spotify para organizar y gestionar su trabajo. Se basa en la idea de "squads" (equipos pequeños multifuncionales), "tribes" (conjuntos de squads que trabajan en áreas relacionadas), "chapters" (grupos de personas con habilidades similares en diferentes squads), y "guilds" (comunidades de práctica que comparten conocimientos y buenas prácticas).



Adicionalmente tienen poca o ninguna estandarización de procesos. No existe como tal un estándar y la creencia en el modelo es basarse más en que emerjan nuevas formas de trabajar en base a la experiencia, que en estandarizar las que deban ser usadas. Esto es,

si mucha gente de la compañía encuentra alguna herramienta muy cómoda y útil, y esa herramienta resulta ser fácil de implantar y del agrado de los equipos, entonces los otros squads terminarán usando esa misma herramienta. De esta manera, se consigue que algo dinámico se convierta en un estándar.

Modelo de trabajo en el sistema de Spotify

Como en cualquier modelo ágil, en Spotify hay roles y agrupaciones de estos. Al contrario que en Scrum, las agrupaciones de estos tienen diferentes dimensiones.

Squads (Equipos, Escuadrones)

Podría equivaler al Scrum Team de Scrum.

Consiste en un pequeño equipo multifuncional y auto-organizado de hasta 8 personas. Es un equipo con responsabilidad de principio a fin y trabajan juntos hacia una meta a largo plazo. Su clave es la autonomía.

Cada 'squad' tiene autonomía para decidir qué hacer y cómo hacerlo, así como la forma de trabajar juntos para conseguirlo.

A pesar de contar con autonomía, es necesario tenerlos alineados mediante la 'misión', la 'estrategia de producto' y usar metas a corto plazo.

Emerge la figura del líder, que es el encargado de comunicar el problema que necesita ser resuelto. Así mismo es responsabilidad del líder el explicar por qué es necesario resolver dicho problema.

Una vez puestos en situación, es el turno de los 'squads' el colaborar entre sí para encontrar la mejor solución.



Tribus

Las tribus son una estructura matricial bastante ligera que lo que pretende es agrupar a los 'squads' y mantenerlos bajo el foco de un producto. Podemos decir que agrupa a los 'squads' bajo un paraguas de negocio, por ejemplo, el desarrollo de funcionalidades relativas a la aplicación móvil que la compañía ofrece a sus clientes.

Las tribus se colocan bajo el mismo espacio físico y el número de squads por tribu no debe de exceder las 100, aunque generalmente hablamos de números bastante inferiores.

La estructura de la tribu está concebida para apoyar al grupo de 'squads' y 'chapters'. La tribu en 2012 estaba dirigida por un único líder, a menudo de ingeniería. El liderazgo de la tribu cambió y ya no es responsabilidad de un solo individuo, sino de un grupo, los líderes de la tribu, que está formado por personas representativas del producto, ingeniería, diseño, negocios, etc ..., cuya formación dependerá de las propias necesidades de la tribu.

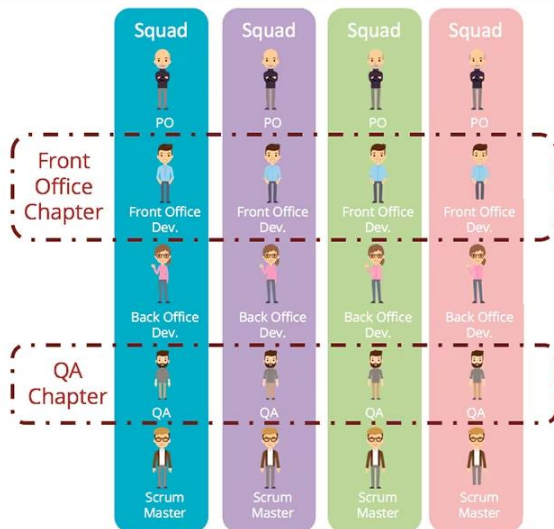
La clave para el liderazgo de la tribu es que, al igual que el liderazgo del capítulo, se trata de un liderazgo de servicio.



Chapter (División, Departamento, Sección)

El 'chapter' es otro tipo de agrupación cross-squad, que se focaliza en áreas de competencia, como por ejemplo, la calidad, el desarrollo front o back, el ux/ui, etc

Los 'chapters' están dirigidos por un 'chapter lead', cuyo propósito principal es apoyar y enfocar el crecimiento de los miembros del 'chapter', tanto como individuos como grupo



Guild (Gremio)

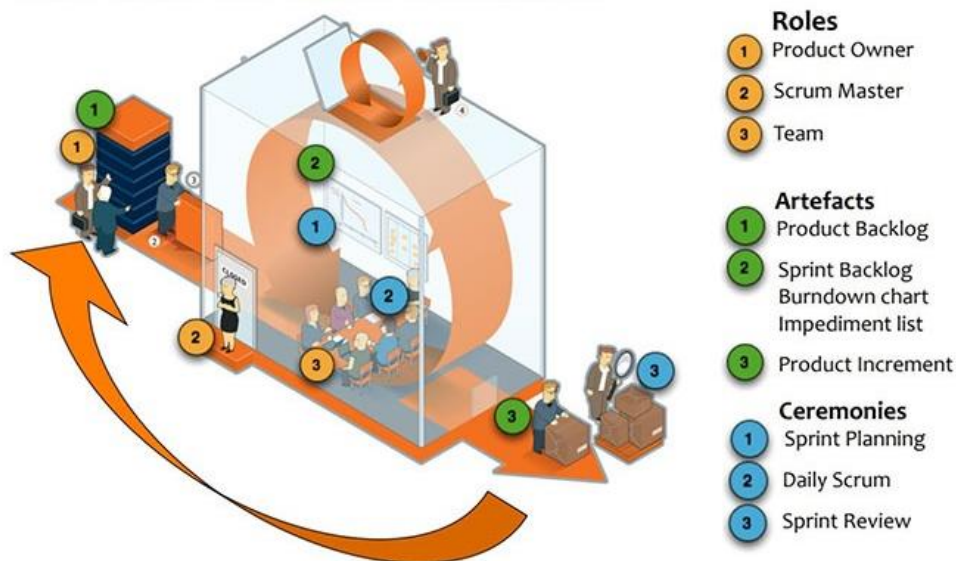
Esta organización, más parecida a una comunidad de interés, agrupa a personas de toda la compañía que quieren compartir conocimiento en un área específica. Es una comunidad abierta, cualquiera puede unirse al gremio o salir de él en cualquier momento. Las personas pueden pertenecer a uno o varios gremios. Cada persona puede decidir cuál de activo o inactivo es en relación a cada uno de los gremios a los que está adscrito.

Los gremios también son orgánicos. La intención original del gremio en 2012 era polinizar y cultivar las maestrías entre las tribus. En la actualidad ha progresado más allá de eso y los gremios se forman en torno a cualquier interés común, incluso aquellos que no tienen que ver con el propio producto o negocio.

Metodología Scrum

Es un framework que permite trabajar en una serie de interacciones en equipo. Las fases que definen y en las que se divide un proceso de SCRUM son las siguientes:

- **El quién y el qué:** identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto.
- **El dónde y el cuándo:** que representan el Sprint.
- **El por qué y el cómo:** representan las herramientas que utilizan los miembros de Scrum.



Roles en Scrum: quién y qué



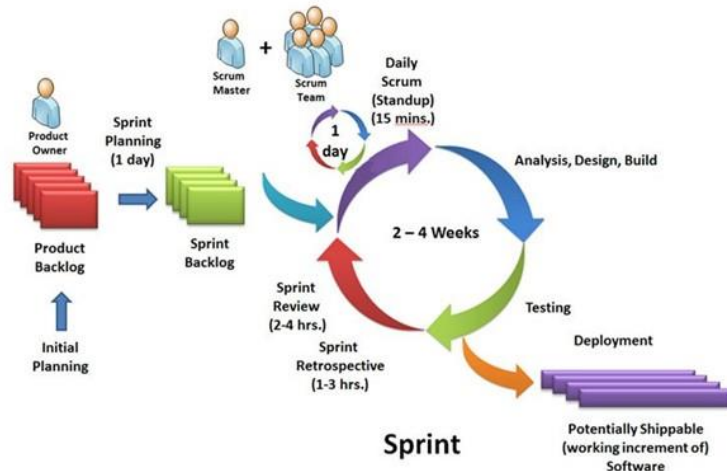
El equipo de Scrum consiste en tres diferentes roles:

El **Product Owner** es la “voz del cliente” y el responsable de desarrollar, mantener y priorizar las tareas en el backlog.

El **Scrum Master** es responsable de asegurarse que el trabajo del equipo vaya bien siguiendo las bases de Scrum. Además, se encarga de remover cualquier obstáculo que pueda encontrar el equipo de desarrollo.

El **Development Team**, son los encargados de escribir y probar el código.

El **Sprint**: dónde y cuándo



Copyright © 2011, William B. Heys

El Sprint es la unidad básica de trabajo para un equipo Scrum. Esta es la característica principal que marca la diferencia entre Scrum y otros modelos para el desarrollo ágil. Es una simple iteración llevada a cabo por los miembros del equipo. Un equipo puede completar varios sprints durante el desarrollo del proyecto.

Un Sprint inicia con un equipo que se compromete a realizar el trabajo y finaliza con la demostración de un entregable. El tiempo mínimo para un Sprint es de una semana y el máximo es de 4 semanas. Dentro del desarrollo de un Sprint se llevan a cabo ciertos eventos, estos reciben el nombre de Scrum Events o Eventos Scrum.

Herramientas Scrum: por qué y cómo

Para poder definir las respuestas a estas preguntas nos valemos de ciertas herramientas que Scrum nos provee. Estas son:

Product Backlog

Esto puede referirse a todo elemento que sea parte del proyecto: puede ser un bug, una referencia o parte de un requerimiento. Brindan información muy general del proyecto y muchas veces no son tomados como requerimientos oficiales.

User Stories

Es un elemento especial del product Backlog. Se llaman historias porque en ellas se proporciona información sobre cómo debe ser el comportamiento del requerimiento que se está trabajando. Su función es proporcionar información directa del cliente en caso de existir algún cambio. Generalmente estos sí son tomados como requerimientos oficiales.

Sprint Backlog

Es el conjunto de elementos tomados del Product Backlog que fueron priorizados, medidos y aceptados en las reuniones de Sprint Planning. Estos, en conjunto con sus respectivos User Stories, forman oficialmente los requerimientos a elaborar en cada uno de los Sprints que tendrá el proyecto.

The Taskboard

Este panel muestra las tareas que tienen asignadas los miembros del equipo. Esta tabla se divide en tres columnas que representan el estado de la actividad:

- a. Por hacer.
- b. Haciendo.
- c. Terminado.

Al inicio del Sprint todas están en la primera columna. Cuando una tarea pasa a la segunda columna, el Scrum Master y el Product Owner son notificados respecto a qué está haciendo cada miembro del equipo y cuánto tiempo lleva trabajando en dicha tarea. Al finalizar, ésta debe cambiarse a la última columna. Esto quiere decir que está listo para que QA haga las pruebas necesarias.

Definition of Done

Todo equipo eficaz y ágil tiene ciertos acuerdos que deben cumplirse antes de dar por finalizado un proyecto. Estos son:

- Todas las tareas están completas.
- Code Reviewed.
- Pruebas realizadas a cada elemento desarrollado.
- Revisión por parte de los clientes (que cumpla sus necesidades).
- La revisión de las condiciones de aceptación por parte del Product Owner.

Estas herramientas son útiles, no sólo durante un Sprint, sino también a lo largo del proyecto, pues ayudan al equipo a entender el porqué de cada actividad. Además, son visibles para el equipo y para los externos.

Spotify vs Scrum

Ventajas de la metodología Spotify:

- Fomenta la autonomía y la colaboración entre los equipos.
- Proporciona una estructura escalable para organizaciones grandes.
- Permite la especialización y la colaboración transversal a través de los "chapters" y "guilds".
- La metodología Ágil de Spotify permite a los equipos adaptarse rápidamente a los cambios en los requisitos del proyecto y a las necesidades del cliente.
- Spotify pone un énfasis particular en entender y satisfacer las necesidades del usuario final.

Desventajas de la metodología Spotify:

- Puede resultar compleja de implementar y gestionar, especialmente en organizaciones pequeñas.
- Requiere una cultura organizativa sólida y madura para funcionar eficazmente.
- La adaptación a las necesidades específicas de cada equipo puede ser complicada.
- Necesidad de comunicación y coordinación efectivas: Dado que los equipos de Spotify son autónomos.

Ventajas de Scrum:

- Proporciona un marco de trabajo simple y fácil de entender.

- Promueve la transparencia, la inspección y la adaptación continua.
- Favorece la entrega temprana y frecuente de productos, lo que permite obtener retroalimentación rápidamente.
- Scrum se centra en la entrega continua de valor al cliente al priorizar las características más importantes y abordarlas primero.

Desventajas de Scrum:

- Puede ser rígido en algunos aspectos, lo que dificulta su adaptación a contextos no puramente de desarrollo de software.
- Requiere una disciplina rigurosa para seguir todos sus eventos y artefactos correctamente.
- No proporciona una estructura explícita para escalabilidad en organizaciones grandes.

Comparativa y en qué caso utilizar cada uno:

La metodología Spotify puede ser más adecuada para organizaciones grandes y complejas que necesitan una estructura escalable para gestionar varios equipos y áreas funcionales.

Scrum, por otro lado, es más adecuado para equipos de desarrollo de software que buscan una manera simple y efectiva de organizar su trabajo en ciclos cortos y entregas iterativas.

Por ejemplo, podríamos utilizar Scrum en caso de que se tenga un sólido concepto del proyecto que se quiera realizar.

Por el contrario, Spotify está preparado para cambios y giros 360° ya que es más dinámico.

En resumen, la elección entre la metodología Spotify y Scrum dependerá del tamaño de la organización, la complejidad del proyecto y las necesidades específicas del equipo.

Fuentes

- <https://grupociusa.com/metodologia-scrum-que-es-y-sus-5-fases/>
- <https://www2.deloitte.com/es/es/pages/technology/articles/introduccion-modelo-agile-spotify.html>
- <https://slash.co/articles/scrum-vs-spotify-model-which-one-is-best-for-your-product-development/>