

El Product Owner

Gestión de productos con Metodologías Ágiles

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

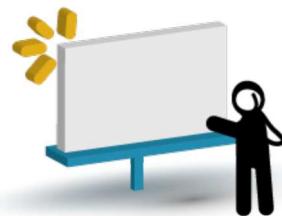
**Centro de
e-Learning**

p. 2

Unidad 1 - El rol del Product Owner (Dueño del Producto)

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Presentación:

En el entorno actual de las empresas el enfoque tradicional de gestión de proyectos encuentra cada vez más limitaciones, lo que lleva a la utilización de un enfoque más "ágil", con el objetivo de entregar valor más rápido y con menores costos.

Dentro de los distintos marcos de trabajo ágil, Scrum es el modelo de gestión más aceptado y documentado a nivel mundial desde hace más de una década, por lo cual vuelve una materia obligatoria su conocimiento para todo aquel que gestiona la construcción de productos.

En toda organización que busca adaptarse al contexto actual de cambios, debe existir el rol de Product Owner o dueño del producto, el cual es clave para definir y priorizar el trabajo a realizar para maximizar los beneficios al cliente y a la organización.



Objetivos:

Que los participantes:

- Aprendan el detalle del paradigma ágil.
- Conozcan las diferencias entre el enfoque tradicional de gestión de proyectos con el enfoque iterativo adaptativo.
- Entiendan Scrum, el marco de trabajo ágil más utilizado actualmente, sus roles, artefactos y ceremonias.
- Conozcan las características del Product Owner y que rol cumple en las organizaciones actuales.



Bloques temáticos:

1. El Paradigma Ágil
2. El framework Cynefin
3. El Marco de Trabajo Scrum
4. El Rol del Dueño del Producto (Product Owner)
5. Escalamiento del Rol de Product Owner
6. ¿Quién debe ocupar el rol según el tipo de proyecto?
7. Problemas comunes



Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

* *El MEC es el modelo de E-learning colaborativo de nuestro Centro.*



Tomen nota:

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.



1. El Paradigma Ágil

¿Qué entendemos por gestión Ágil?

En la actualidad Ágil (o Agile¹ por su nombre en inglés) es una de las palabras que indudablemente todo profesional de la industria de la tecnología o nuevos productos/ servicios ha escuchado decir, y posiblemente también sea una palabra que despierta de los más diversos entendimientos.

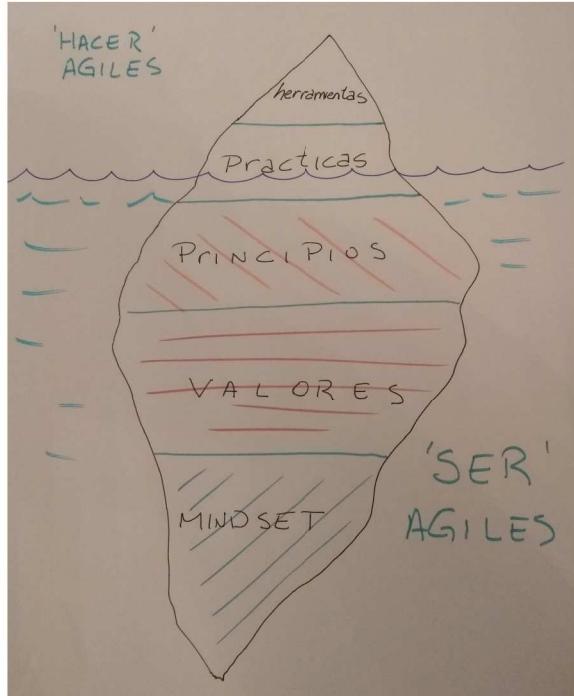
Primero vamos a decir que el agilísimo no es una metodología en sí mismo, sino un paradigma alternativo de como afrontamos el mundo y gestionamos la construcción de productos y servicios². Con el enfoque Ágil (agile mindset) se busca ayudar a los equipos a responder a la impredecibilidad inherente a todo nuevo emprendimiento a través de trabajo iterativo incremental con foco en feedback y aprendizaje empírico.

Es un enfoque alternativo al desarrollo secuencial y predictivo (comúnmente conocido como cascada o waterfall) en el cual los equipos solo tienen una oportunidad de entender correctamente cada aspecto del trabajo. Al seguir el paradigma ágil, cada aspecto (requerimientos, diseño, etc.) es continuamente revisado/cuestionado durante el ciclo de vida, con el objetivo de mantener el enfoque en la relevancia crítica del negocio. Esto no es algo recientemente descubierto, ya en la década del '70 autores como Winston Royce ("Managing the Development of Large Software Systems"³), criticaban utilizar al proceso secuencial estilo línea de producción para el desarrollo de producto.

1 <https://www.agilealliance.org/agile101/>

2 <https://www.101ways.com/2007/02/10/what-is-agile-10-key-principles/>

3 http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf



El agilismo se lo puede conocer por herramientas, que provienen de prácticas. Pero todo esto se sustenta en un mindset, con 4 valores y 12 principios.

El Manifiesto Ágil

En 2001 un grupo de reconocidos autores de libros, prácticas y marcos de trabajo se reunieron en un resort de Colorado (EE.UU.) para dar forma a lo que conocemos como manifiesto ágil, el cual expresa lo siguiente⁴:

Individuos e interacciones	sobre	Procesos y herramientas
Software que funciona	sobre	Documentación exhaustiva
Colaboración con el cliente	sobre	Negociación de contratos
Responder ante el cambio	sobre	Seguimiento de un plan

Esto es... Aunque consideramos importante los elementos indicados a la derecha priorizamos los elementos de la izquierda.

1. Individuos e interacciones sobre procesos y herramientas

⁴ <http://agilemanifesto.org/>

No se niega la necesidad de procesos y herramientas: los procesos ayudan al trabajo, sirven de guía de operación; y posiblemente las herramientas mejoran la eficiencia y soporte a los procesos. Pero... Los procesos y las herramientas **deben ser una ayuda** para guiar el trabajo, **deben adaptarse** a la organización, a los equipos y a las personas; **y no al revés**. Sin personas con el conocimiento y una actitud adecuada, los procesos y las herramientas no generan resultados.

2. Producto funcionando sobre documentación exhaustiva

La perspectiva subyacente al *Manifiesto Ágil* establece el éxito desde el punto de vista de la **entrega de valor para el negocio**. Conceptualmente la documentación que se suele elaborar en los proyectos no aporta valor directo al negocio. El valor para el negocio se habilita únicamente cuando se entrega un resultado. Adicionalmente ver el producto funcionando facilita fuertemente el feedback en comparación con la lectura de un documento.

Es probable que se requiera alguna documentación para poder llegar a este objetivo (o seguir produciendo en el futuro), pero el *Manifiesto Ágil* reconoce que al fin y al cabo la única medida real de avance de un proyecto es la entrega de valor, y por esta razón debe que ser el objetivo y prioridad permanente de cualquier equipo.

3. Colaboración con el cliente sobre negociación contractual

Las prácticas ágiles cobran particular relevancia para el desarrollo de productos difíciles de definir con detalle en el principio o cuando los requisitos pueden ser muy volátiles. En tales casos suelen fracasar o a menos complejizar el uso de gestiones basadas en modelos contractuales cerrados que en general terminan generando **intercambio de “culpas” o responsabilidades** por las diferencias entre el plan original y la realidad.

En el desarrollo ágil **se busca sumar al cliente** como un miembro más del equipo, que se integra y colabora en el grupo de trabajo. Se trabaja en general con un marco contractual de alto nivel sobre el cual se construye una relación de confianza basada en los resultados logrados.

4. Respuesta ante el cambio sobre seguir un plan

En entornos volátiles, donde el cambio es continuo e imprevisible, se requiere una capacidad para la evolución rápida y continua. El seguimiento de planes pre-establecidos suele dificultar el enfrentar este desafío con éxito, partiendo que suele ver el cambio como un problema a diferencia de la gestión ágil que busca adaptarse y beneficiarse del aprendizaje.



Principios del manifiesto Ágil

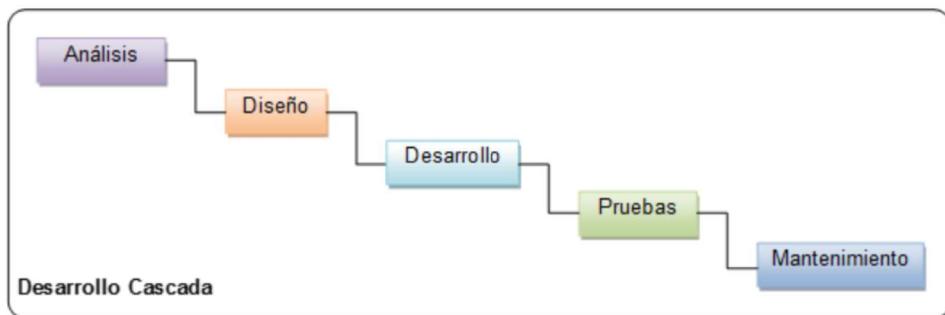
1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de productos con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos producto funcionando frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El producto funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

**Esta es una versión adaptada del manifiesto, para todas las industrias.*

** *Los valores y principios aquí mencionados, cuanto más se los cumpla, más agiles será su forma de trabajo. Se recomienda no tomar esto como un blanco o negro, sino como premisas a usar como guía.*

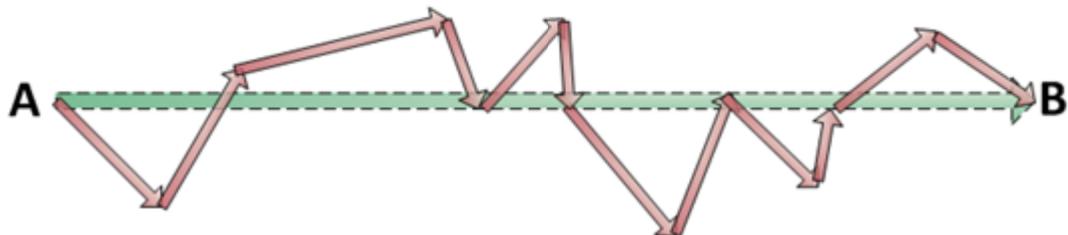
Adaptivo versus Predictivo

El enfoque predictivo u orientado a plan (Plan-driven) tiene su origen en otras disciplinas ingenieriles tales como civil y mecánica. En ellas es muy caro adaptar los cambios cuando la construcción ya empezó. Sumado a esta dificultad, se tiene una certidumbre bastante alta sobre el dominio y la problemática. Por lo cual dedicar un esfuerzo grande a planificar detalladamente permite un cronograma razonablemente fiable.



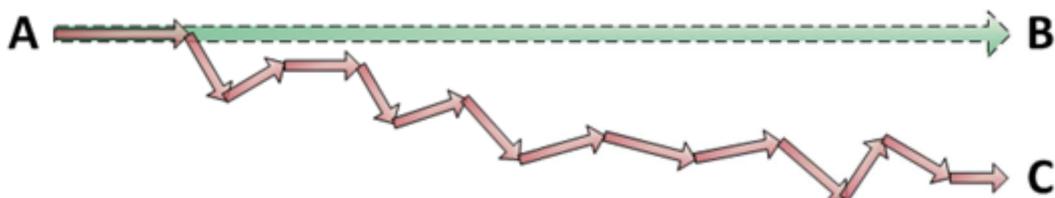
En cambio, al realizar desarrollos en un contexto de alta incertidumbre (de construcción, técnica, de satisfacción, de calidad, etc.) no se cumple con esas premisas. No hay garantía de que un buen diseño nos brinde a la construcción predecible y de calidad y para empeorar la situación, al adoptar el enfoque predictivo todo lo que no estaba planeado resulta en un problema y solo podemos esperar a tener un proceso para controlarlo y mitigarlo.

En la imagen podemos ver justamente que se requiere llegar al punto B como fue planeado (Tiempo, Dinero y Alcance) para tener un proyecto exitoso.



Esto nos puede dar la paradoja de cumplir con los criterios de éxito de un proyecto (entrega en tiempo, con el presupuesto y alcance acordado) pero que no le da el valor esperado al usuario.

A diferencia de esto, si tomamos una estrategia de adaptarnos al cambio a medida que aprendemos durante el proceso siendo esto una posibilidad de tomar mejores decisiones, nos da como resultado que el mismo punto de destino puede cambiar, y esto no es un problema sino un resultado deseable del conocimiento obtenido.



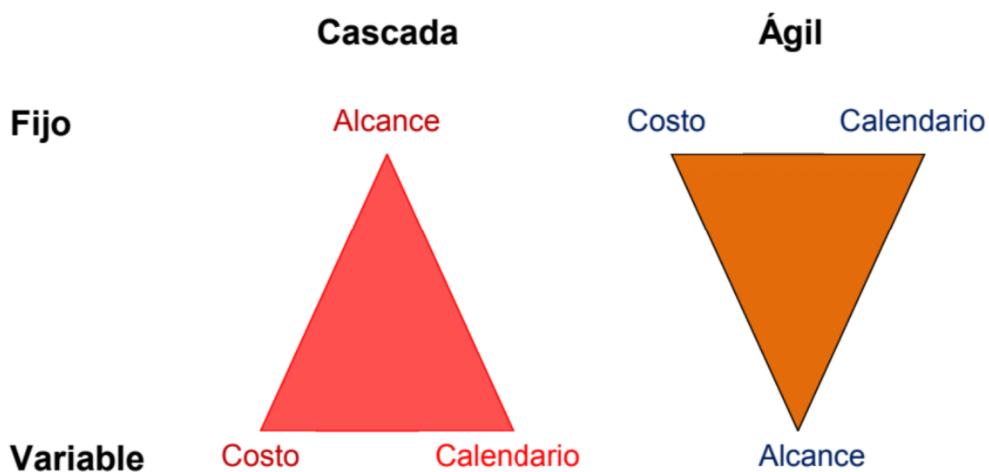


Si podemos usar una metodología que nos permite cambiar, podemos explotar el aprendizaje obtenido en el curso del proyecto, sin necesariamente incurrir en mayores costos, y obteniendo otro resultado (C) que será un producto más valioso que el diseñado (B).

"El negocio cambia. La tecnología cambia. El equipo cambia. Los individuos cambian. El problema no es el cambio per se, porque el cambio va a ocurrir; el problema es la inhabilidad para adaptarse cuando llega". Kent Beck

Variables en los proyectos

Cuando hablamos de proyectos, la teoría indica 3 variables a tener en cuenta (normalmente conocido como triángulo de gestión de proyectos o triple restricción): Alcance, Costo y Tiempo. En los modelos predictivos (siendo el método cascada el más conocido) en general se asume el alcance como una constante, siendo las otras variables (tiempo y costo) las que pueden ajustarse. En cambio, en el enfoque adaptativo (o Ágil) se invierte la pirámide dejando como fijos los costos y fechas (iteraciones) y la variable de ajuste está en el alcance a implementar, el cual se acuerda entre las necesidades y posibilidades técnicas.



Marcos de trabajo agiles

Existen múltiples enfoques que llevan los valores ágiles y durante las últimas 4 décadas varios marcos fueron surgiendo y utilizándose con distintos niveles de éxito y aceptación. La mayoría



promueven el trabajo en equipo, colaboración, transparencia, adaptabilidad y trabajo en conjunto con el usuario durante todo el ciclo de vida del software como base.

Entre las más populares podemos destacar las siguientes:

Marco de trabajo	Descripción	Creadores
Scrum	Scrum es un framework para la gestión y desarrollo ágil basado en un proceso iterativo e incremental. Toma conceptos de metodologías desarrolladas previamente y en la actualidad es la más utilizada a nivel mundial. Hoy agnóstico a la industria.	Jeff Sutherland, Ken Schwaber
Extreme Programming (XP)	XP organiza equipos de desarrollo de software con un ritmo sostenible de trabajo, con desarrolladores trabajando de a pares que realmente disfrutan de su trabajo y con un cliente disponible a quien se entrega periódicamente. El diseño del sistema está claramente descrito para todos a través de una metáfora, se sostiene por una suite de pruebas del desarrollador y del cliente, y se mantiene entendible por un refactoring constante y un respeto del estándar de programación bajo responsabilidad colectiva del equipo. Algunos de estos principios son trasladables a otras industrias	Kent Beck
Lean Software Development	La metodología de desarrollo de software Lean es una translación de los principios y prácticas de la manufacturación Lean hacia el dominio del desarrollo de software. Adaptado del sistema de producción Toyota, adopta los principios siguientes: eliminar los desperdicios, ampliar el aprendizaje, decidir lo más tarde posible, reaccionar tan rápido como sea posible, potenciar el equipo, crear la integridad y ver todo el conjunto. Todos estos principios son trasladables a otras industrias.	Mary & Tom Poppendieck
Kanban	Kanban tiene como objetivo la entrega eficiente de	David Anderson

	valor al cliente. Promueve el concepto Lean de flujo para entregar valor continuamente de manera predecible. El trabajo y los procesos se visualizan para identificar actividades y problemas. Se limita el trabajo en curso para motivar calidad, foco y finalización de las tareas. La intención en la eficiencia del proceso controlando el trabajo en curso. (sea cual sea la industria).	
Feature Driven Development (FDD)	FDD es un proceso de desarrollo de software iterativo e incremental. FDD engloba varias buenas prácticas reconocidas de la industria del software en una metodología coherente. Estas prácticas están todas derivadas de una perspectiva de funcionalidades valoradas por el cliente. Su principal objetivo es la entrega de software tangible funcionando en el tiempo adecuado. Algunos principios trasladables a otras industrias no software.	Jeff De Luca
Ágile Unified Process (ÁgileUP)	ÁgileUP es una versión simplificada del <i>Rational Unified Process</i> (RUP). Describe un enfoque simple y fácil de entender para desarrollar software utilizando técnicas y conceptos ágiles sin salir del modelo RUP. El enfoque aplica técnicas ágiles como <i>Test Driven Development</i> (TDD), <i>Ágile Model Driven Development</i> (AMDD), Gestión de Cambio Ágil y Refactoring de Base de Datos, para mejorar la productividad. Algunos principios trasladables a otras industrias no software	Scott Ambler
Crystal	Crystal es una familia de metodologías de desarrollo de software (trasladable a otras industrias). Están enfocadas en el éxito del proyecto a través de las mejoras del trabajo de las personas involucradas. Cualquiera sea el tamaño y las prioridades del proyecto, hay una metodología	Alistair Cockburn



	de la familia que funcionará para reducir el papeleo y la burocracia y será práctica para los parámetros de ese proyecto. Son adaptable en el sentido que se puede empezar con mecanismos muy limitados y simples y agregar a medida que sea necesario mecanismos más complejos y escalables.	
--	---	--

En este curso utilizaremos en mayor medida conceptos de **Scrum** por ser el marco de trabajo ágil más documentado y explotado desde hace más de una década⁵. A título de ejemplo se pueden nombrar algunas empresas que aplicaron o aplican Scrum: Microsoft, Yahoo, Google, Electronic Arts, High Moon Studios, Lockheed Martins, Philips, Siemens, Nokia, Capital One, BBC, Intuit, Nielsen Media, First American Real Estate, BMC Software, Ipswitch, John Deere, Lexis Nexis, Sabe, Salesforce.com, Time Warner, eduScrum, Banco Galicia.

Vale la pena destacar que, desde ya hace más de una década, el paradigma ágil ha dejado de ser algo exclusivo de la industria de software para ingresar a todo el resto de las industrias y sectores. Siendo aplicado desde la gestión de las personas, las finanzas, construcción, etc.

⁵ Según la encuesta "Agile Dev Survey" del año 2016 el 54% de las organizaciones que utilizan algún entorno de trabajo ágil utiliza Scrum, 11% algún híbrido, 9% un híbrido propio y 7% Scrumban



Caso de estudio: Facebook vs My Space

Previo a la existencia de Facebook, Twitter (y otras redes sociales) MySpace fue una de las precursoras de las redes sociales, permitiendo funcionalidades no esperadas por el público general de esa época (creciendo tanto que superó a Google en cantidad de visitas en Estados Unidos) y en tan solo 2 años fue adquirida en 580 millones de dólares por News Corp.

Justamente News Corp intentó marcar su impronta con una “Gestión profesional” para dirigir la evolución de MySpace, con una planificación detallada del roadmap de las próximas implementaciones de acuerdo a su análisis del mercado.

Este enfoque probó su debilidad al momento de competir con el recién llegado (por ese entonces) Facebook con una gestión “Espacio en blanco” (White Space), dejando que el mercado decidiera donde el negocio debería ir.

Los fundadores siguieron empujando la tecnología para hacer lo que los usuarios querían. Si alguien tenía una idea para “networking” en algo, Facebook desafiaba a sus equipos de desarrollo para hacerlo posible. Su análisis se basó en escuchar el feedback de los usuarios para decidir lo que incorporaban en cada release, lo cual llevó a más y más usuarios que brindaron mayor experiencia y rapidez en la evolución.

Justamente ese es el espíritu de este tipo de gestión. *No hay reglas, no hay planes detallados, no hay predicciones del comportamiento del mercado* o de los usuarios, en particular no se trata de ser más inteligentes que los usuarios para determinar lo que deben o no hacer. No se prejuzgan ideas o límites tecnológicos o en analizar y planificar el avance correcto. Por el contrario, se experimentó lo que podía florecer en el mercado, y conocer aquello que lo hacía. Contrariamente en MySpace, los líderes tenían un plan, no eran universitarios sin experiencia en los negocios. Los líderes crean diapositivas describiendo a donde el negocio se dirigirá, donde invertirán, como podrán tener el máximo ROI con proyecciones que funcionaran y los planes para lograrlo. Ellos desarrollaron el plan, actuaron en consecuencia. Creyeron predecir el futuro, decidieron y ejecutaron... Equivocándose a lo grande, por no tener la ‘agilidad’ para adaptarse⁶.



Podríamos mencionar además otros casos como Blockbuster vs Netflix o Blackberry vs Android. Los invitamos a buscar otros casos que considere representativos de éxito / fracaso del enfoque y compartirlos en el foro.

⁶ <https://www.theguardian.com/technology/2019/mar/18/myspace-loses-all-content-uploaded-before-2016>

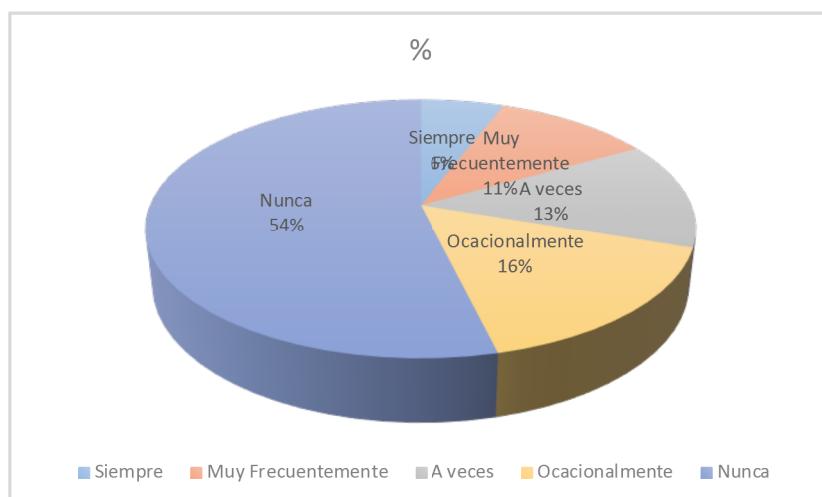


Estado actual de la gestión de proyectos

En el año 1994 Standish Group presentó el lapidario informe conocido como Chaos Report, el cual sobre una base de datos de casi 50 mil proyectos en todo el mundo (principalmente iniciativas de IT y software) en cual la tasa de éxito general de los proyectos estaba en el 16%; definiendo éxito como un cumplimiento razonable del tiempo proyectado, el costo y principalmente la satisfacción del cliente.

	2011	2012	2013	2014	2015
Exitoso	29%	27%	31%	28%	29%
Con Problemas	49%	56%	50%	55%	52%
Fracasados	22%	17%	19%	17%	19%

En los siguientes años, si bien los números mejoraron aún se mantienen tasas un tanto preocupantes sobre los resultados de los proyectos. Para sumar gravedad al estado actual, el estudio expresa que más del 60% de las funcionalidades de los sistemas son rara vez o incluso nunca usadas. Lo cual denota una inversión de las empresas en funciones que no tenían (o ya no tienen) importancia, valor, utilidad, etc.



Otra categorización en el reporte "Chaos" (2015) muestra la mayor probabilidad de éxito que tienen los proyectos chicos sobre los de mayor envergadura (con tipo de gestión):

Tamaño	Tipo de Gestión	Exitosos	Con Problemas	Fallidos
Todos los	Agil	39%	52%	9%

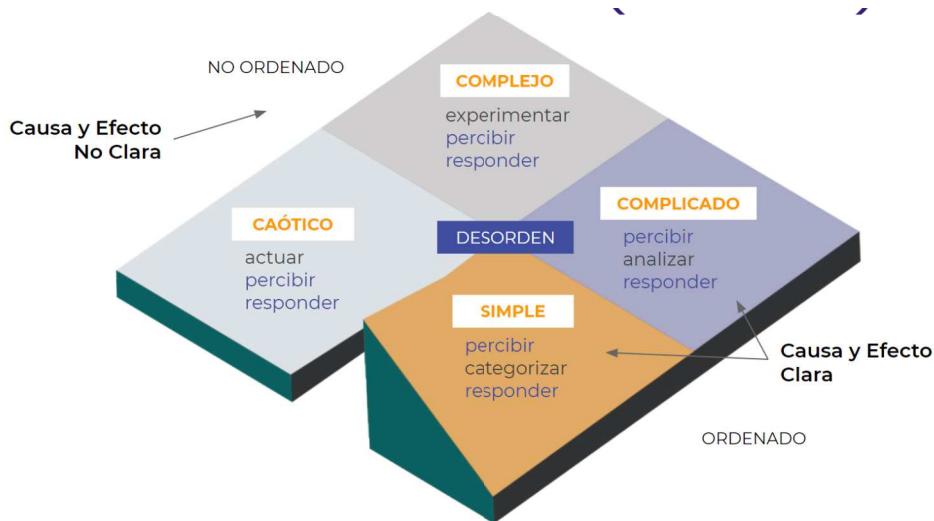


tamaños				
	Tradicional	11%	60%	29%
Grande	Agil	18%	59%	23%
	Tradicional	3%	55%	42%
Medianos	Agil	27%	62%	11%
	Tradicional	7%	68%	25%
Pequeños	Agil	58%	38%	4%
	Tradicional	44%	45%	11%



2. El framework Cynefin

Con el objetivo de facilitar la decisión de la mejor estrategia para encarar un desafío nos es de gran utilidad conocer el dominio sobre el cual estamos trabajando. Con este objetivo de catalogar el lugar que “habitamos” fue creado el modelo Cynefin (hábitat en Gales). Las bases del modelo es la relación entre la causa y sus efectos y como resultado surgen los siguientes dominios (o áreas) de problemas:



Tipos de Dominios

Simple, son sistemas en los cuales se puede asociar fácilmente la causa y el efecto. Esta área abarca los dominios que son bien conocidos y documentados. No hay una interpretación o heurística y existen muchos recursos disponibles para realizarlo.

Por ejemplo, al comprar un nuevo televisor y configurarlo uno tiene un manual de usuario que permite ejecutar pasos específicos y poder realizarlo con facilidad, aunque sea la primera vez. Esta área permite el uso de instructivos, indicaciones y modelos de fácil dirección y puesta en práctica.

Sentir (la situación) luego Categorizar para Actuar

Complicado, Son sistemas donde hay una relación de causa-efecto, pero es difícil (o no directamente intuitivo) de detectar. Encontrar una solución a problemas de este dominio usualmente requiere de un juicio experto, experiencia y análisis. Fuera de eso estos entornos suelen ser estables y no sufrir muchos cambios (y en caso de ocurrir tienen un comportamiento

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



predecible). Ejemplos de estos sistemas son los autos, los cuales tienen problemáticas que pueden diagnosticarse y arreglarse con experiencia y práctica.

Sentir (el problema) luego Analizar para Responder

Complejo, Son sistemas donde no hay una clara relación entre causa y efecto porque cambian con el tiempo. Podemos tener una aproximación a través de experimentos e investigación del estado actual. Incluso el juicio experto no permite arribar necesariamente a una solución en todos los casos, pero puede orientarnos en una dirección de análisis.

Sistemas en este dominio están vivos, son orgánicos y cambian. Donde hay gente, usualmente son sistemas complejos. Ejemplos pueden ser la bolsa de valores, el cerebro, el sistema inmunológico, sociedades, etc.

Probar luego Sentir para Responder

Caótico, Sistemas en los cuales no hay una relación de causa-efecto. No se pueden detectar porque no existen. Solo podemos tener prácticas que nos permitan mitigar el impacto. Ejemplos de estos dominios son fuegos y desastres naturales.

Actuar luego Sentir para Responder

Desorden, Una situación en la cual no podemos definir en qué tipo de dominio nos encontramos. Debemos entender más del problema para poder categorizar y de esa manera definir nuestro plan de acción.

Relación con el desarrollo de productos/servicios

Vamos a dar algunos ejemplos de cuando tenemos sistemas de los distintos tipos de categorías:

- 1- Tenemos un sistema simple cuando vemos un primer nivel de soporte, en la cual el cliente llama para realizar una consulta sobre cómo se utiliza alguna funcionalidad, esto es explicado por el operador y resuelto en el momento (y hasta tal vez pudo haber sido resuelto por el mismo usuario de haber dedicado más tiempo o esfuerzo).



- 2- Entramos en el área de sistemas complicados cuando por ejemplo tenemos un problema menor en el arranque con el auto que podría ser resuelto por un mecánico o una persona con una noción, pero difícilmente por alguien que nunca realizó tareas de ese estilo.
- 3- Cuando pasamos a un contexto cambiante, en el cual no podemos definir de manera precisa las actividades y la duración que tienen estamos en el dominio complejo.
- 4- Obviamente también existen los sistemas caóticos, en los cuales no hay reglas si no solo soluciones ad-hoc para cada caso en los cuales se trata de sistematizar, estandarizar y entrenar de acuerdo a experiencias pasadas para tratar de facilitar acciones futuras. En estos contextos debemos buscar la manera de llevarnos a alguno de las 4 áreas con aprendizaje.

¿En qué nos ayuda?

Este modelo es útil para ayudarnos a entender el sistema sobre el cual estamos trabajando y el contexto en el que se encuentra para seleccionar la mejor estrategia, y herramientas que podemos usar.

Contexto	Estrategia	Actividades del líder	Herramientas
Sistemas Simples	Sentir Categorizar Responder	Monitorear el progreso Estandarizar Delegar Usar mejores prácticas	Mejores estrategias Cinco "¿por qué?" Manuales, Recetas, PMBOK
Sistemas Complicados	Sentir Analizar Responder	Formar un panel experto Mediar	Procesos, Estándares Paneles Expertos Kanban, PMBOK
Sistemas Complejos	Probar Sentir Responder	Dejar espacio para la auto organización del equipo Mejorar Comunicación Empoderar	Experimentos Discusiones Auto organización Agile, Kanban
Sistemas Caóticos	Actuar Sentir Responder	Actuar y concluir en esta base Comunicar de manera simple y sin ambigüedades Dirigir y controlar	Hacer según la ocasión Intuición



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

p. 23

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



3. Principios Lean

La llamada gestión “Lean” es un modelo de gestión que tiene sus orígenes en la manufactura japonesa (que también toma algunos conceptos previos), la cual está enfocada a la creación de flujo para poder entregar el máximo valor a los clientes utilizando los mínimos recursos necesarios. El desarrollo más profundo se dio en la post guerra con el llamado “Toyota Way” tomando conceptos del llamado Kayzen (‘cambiar para mejor’ en japonés, también conocido como mejora continua) que busca la optimización de los procesos⁷.

La palabra “Lean” proviene justamente del término magro o sin grasa, por la eliminación de “Muda” o desperdicio y su objetivo es trabajar para eliminarlo por completo y así no utilizar ningún recurso que no entregue o aporte valor.

El “Desperdicio” puede darse por los siguientes motivos:

- Transporte: Movimiento de productos que no son requeridos para el procesamiento.
- Inventario: Todo componente, trabajo en curso y productos finalizados que no están siendo procesados.
- Movimiento: Desplazamiento de personas o equipos más de lo requerido para el procesamiento.
- Espera: Tiempos ociosos entre fases del proceso.
- Sobre producción: Mayor producción de la demandada.
- Sobre procesamiento: Operaciones extras (tales como retrabajos, materiales innecesarios, actividades que no suman valor, etc).
- Defectos: Esfuerzo involucrado en la corrección e inspección.

Ciclo de mejora Lean

⁷ The Toyota Way : 14 Management Principles from the World's Greatest Manufacturer



1. Identificar

2. Mapear el
Stream

5. Buscar
perfección

3. Crear el
flujo

4. Establecer
"pull"



Valores a tener en cuenta al aplicar estos principios

1. Basar la gestión de decisiones en filosofía de largo plazo, incluso a expensas de objetivos financieros de corto plazo.
2. Crear un flujo continuo para traer los problemas a la superficie.
3. Usar sistemas de "tire" (pull) para evitar sobreproducción.
4. Nivelar la carga de trabajo (heijunka).
5. Construir una cultura de detenerse para arreglar problemas, con el objetivo de obtener calidad desde el principio.
6. La estandarización de tareas es fundamental para la mejora continua y el empoderamiento (empowerment) de los empleados.
7. Usar control visual para que no haya problemas ocultos (transparencia).
8. Usar solo tecnología confiable y fuertemente testeada que sirva a la gente u procesos.
9. Hacer crecer líderes que entienden el trabajo, viven la filosofía y enseñan a otros.
10. Desarrollar personas excepcionales y equipos que siguen la filosofía de la compañía.
11. Respeta la red de socios y proveedores a través de desafíos y ayuda para su mejora.
12. Ve y observa por ti mismo para entender la situación (genchi genbutsu o 'go & see').
13. Toma las decisiones despacio y por consenso, analizando y considerando todas las opciones; implementarlas rápidamente una vez tomadas.
14. Conviértete en una organización que aprende a través de profunda reflexión (hansei) y mejora continua (kaizen).

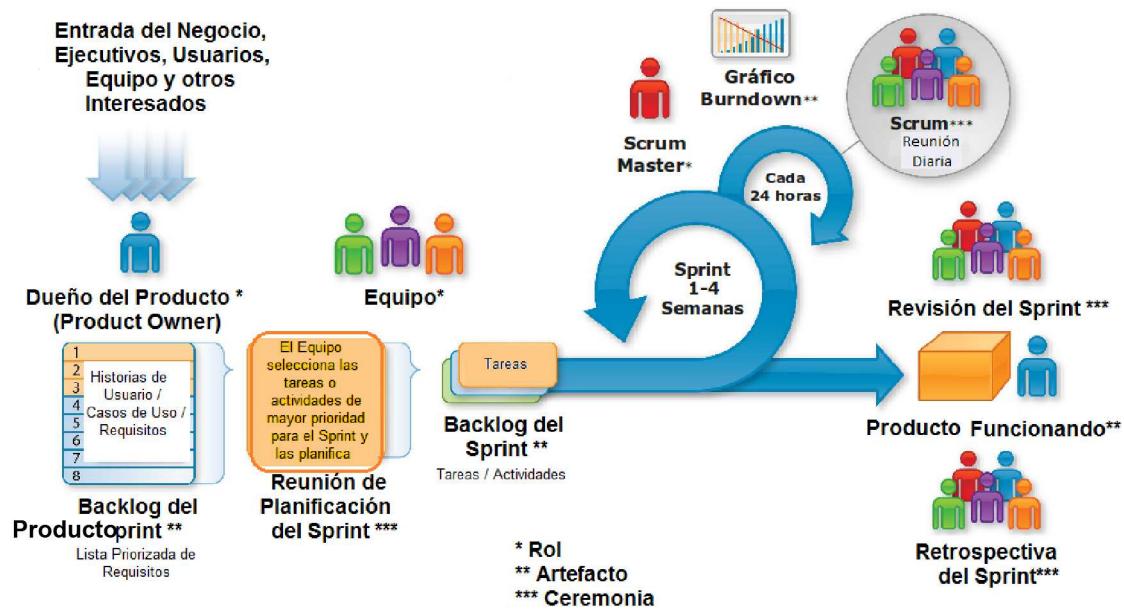


4. El Marco de Trabajo Scrum

¿Qué es Scrum?

Cuando Jeff Sutherland creó el proceso de Scrum en el 1993, utilizó ese término por una analogía presentada en un estudio de 1986 de Takeuchi y Nonaka⁸. En este estudio se comparan equipos de alto rendimiento y multidisciplinario a la formación del “Scrum” de los equipos de rugby.

A continuación, se presenta un diagrama y una explicación del ciclo de Scrum:



9

- El Dueño del Producto crea una lista priorizada de requisitos llamada Backlog de Producto.
- Durante la Planificación (2 - 4hs), el Equipo selecciona una pequeña porción de esta lista, un Backlog de Sprint, y decide cómo implementar estos requisitos.
- El equipo tiene un tiempo limitado, un Sprint (1 a 4 semanas), para terminar su trabajo, pero se reúne cada día para constatar su progreso en la Reunión Diaria (15 minutos).
- En este camino, el Scrum Master mantiene el equipo enfocado hacia su objetivo.
- Al final del sprint, el trabajo debería ser Potencialmente Entregable, o sea estar listo para enviar a un cliente, empaquetar para su distribución o presentar a un sponsor.
- El sprint termina con una Revisión de Sprint y una Retrospectiva.

8 Hirotaka Takeuchi e Ikujiro Nonaka (1986) - *New New Product Development Game* - Harvard Business Review

9 <https://www.scrumguides.org/>



- Cuando se inicia el próximo sprint, el equipo selecciona otra pequeña porción del backlog de producto y empieza a trabajar de nuevo.

Este ciclo se repite hasta que el backlog de producto se haya completado, que el presupuesto se haya gastado o que un hito definitivo haya llegado, según la especificidad de cada proyecto.

Roles en scrum

Scrum formaliza tres roles: el Product Owner (Dueño de Producto), el Scrum Master y el Equipo.

Este es un punto a destacar ya que explícitamente se remarca que no existen otros roles dentro del equipo de elaboración solo deben tener los diversos skills necesarios para ser un equipo autónomo.

Product Owner (Dueño de Producto)

Es la voz del cliente, el que establece una visión sólida del producto a desarrollar y prioriza continuamente los requisitos para alcanzar dicha visión.

Sus principales responsabilidades:

- Canalizar las necesidades del negocio
- Maximizar el valor para el negocio
- Inspeccionar y adaptar el producto
- Comunicar la visión de producto y negocio

Scrum Master

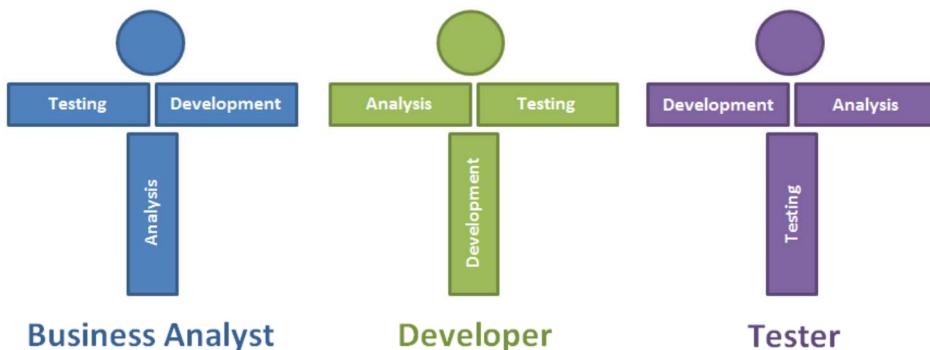
El Scrum Master es un líder servicial para el equipo, pero también un facilitador y agente de cambio para la aplicación adecuada de Scrum en el equipo y la organización.

Sus principales responsabilidades:

- Asegurar la correcta aplicación de Scrum.
- Facilitar los eventos y la mejora continua.
- Eliminar los impedimentos que frenan el trabajo del equipo.

Equipo

El equipo es un grupo auto-organizado, multidisciplinario y autónomo que desarrolla el producto. Su única responsabilidad es de transformar el Backlog de Producto en incrementos de funcionalidad potencialmente entregable en cada sprint.



Artefactos de Scrum

Scrum identifica tres artefactos o productos de trabajo: el Backlog de Producto, el Backlog de Sprint, y el Incremento de Funcionalidad Potencialmente Entregable.

Backlog de Producto

El Backlog de Producto es una lista única, pública y dinámica que recopila una secuencia priorizada de requerimientos para el producto. Algunos de estos requerimientos tienen una estimación de alto nivel de esfuerzo o complejidad. Se suele decir que el Backlog de Producto representa el “Qué” y el “Para Que” esperado del producto, sin preocuparse por el “Cómo”. Esta es una las principales herramientas del P.O.

Backlog de Sprint

El Backlog de Sprint es un conjunto reducido y negociado de ítems del Backlog de Producto que el equipo se compromete a completar durante el tiempo asignado al Sprint. El Backlog de Sprint es actualizado diariamente por el equipo, para mostrar los pendientes, en curso e ítems terminados. Se suele utilizar un tablero de sprint como forma de visualizar el backlog del sprint, o algún otro soporte que permite darle visibilidad (clave tal como en un Kanban).

Incremento de Funcionalidad Potencialmente Entregable

Al final de cada sprint el equipo debería entregar un incremento de funcionalidades del producto con una calidad productiva. El incremento del producto debe funcionar y si quedan tareas pendientes para su entrega, deberían requerir un esfuerzo mínimo.

Eventos de Scrum

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Scrum identifica cinco eventos (antes llamados ceremonias) o prácticas a ejecutar en cada sprint: el *Refinamiento*, la *Planificación*, la *Reunión Diaria (daily stand-up)*, la *Revisión (Review o Demo)* y la *Retrospectiva*.

Planificación

En este evento realizado al principio de cada sprint, se revisa el backlog junto al PO para entender y determinar el trabajo que el Equipo se compromete a terminar en el Sprint. El equipo setea un objetivo, entiende en detalle lo que se espera y define cómo se va a realizar este trabajo.

Reunión Diaria (Daily Stand-up Meeting)

En esta breve y ejecutiva reunión (debería durar unos 15 minutos) se espera que los integrantes del equipo puedan compartir el avance del trabajo comprometido para el Sprint, si se detectó algún impedimento y/o se requiere alguna colaboración. Comúnmente cada integrante responde a 3 preguntas:

- ¿Qué hice ayer?
- ¿Qué voy a hacer hoy?
- ¿Tengo algún impedimento?

En caso de haber temas puntuales se resuelven post reunión entre los involucrados para evitar que se extienda innecesariamente.

Revisión (Review)

La revisión es una presentación que realiza el equipo al finalizar el sprint para mostrar el resultado del trabajo realizado, con ella podemos obtener validación, mostrar avance concreto y obtener feedback sobre lo construido que sirva como input para el refinamiento del backlog. Habiendo dicho eso, no implica que no se pueda (o deba) buscar feedback en todo momento del sprint.

Retrospectiva

La retrospectiva es un evento en la cual el equipo repasa lo ocurrido en el sprint para identificar colaborativamente las causas de los principales problemas con el objetivo de consensuar y seleccionar acciones de mejora concretas para ejecutar en el próximo sprint.

Refinamiento

Este evento (el único opcional de scrum) normalmente se realiza durante el curso del sprint con los siguientes objetivos: Reducir la complejidad y duración de las reuniones de planificación de sprint, habilitar la discusión de nuevos ítems considerados para plantear cuestiones a investigar,



detectar dependencias y sobre todo preparar un slicing (termino que veremos en detalle más adelante) para desagregar los ítems del backlog del producto de cara a próximos sprint.

No hay un momento fijo en el sprint, ni es mandatorio que este todo el equipo, se recomienda al menos scrum master (quien la facilita) el product owner y alguien de cada skill requerida para el ítem con el objetivo de tener una visión completa. Cada equipo puede agendarlo de acuerdo a su contexto, aunque es una práctica común tener un espacio fijo en el calendario para contar con agenda y lugar donde realizarla periódicamente.



5. El Rol del Dueño del Producto (Product Owner)

Como ya vimos en párrafos anteriores, el Product Owner es uno de los 3 roles que constituye un equipo scrum. Es la persona con la responsabilidad final en el liderazgo del producto y que debe servir de medio para lograr que el equipo de elaboración construya el producto requerido por los interesados y los usuarios.

Debe participar de la planificación de Portfolio y de Producto para garantizar que estas planificaciones mantengan un horizonte alineado. Asegurarse de la realización de los procesos de incepción, mapeo de historias de usuario, definición de un producto mínimo viable –MVP– (términos a ver en detalle más adelante). Así como también mantener un backlog refinado, actualizado y validado para sostener un ritmo de construcción constante, manteniendo un contacto efectivo con el equipo de elaboración que permita responder consultas, mantenerse actualizado y dar feedback.

Responsabilidades de un PO en Scrum

Las responsabilidades pueden variar en alcance de acuerdo al producto y su madurez (profundizaremos el tema en próximas unidades). A medida que avancemos en la descripción incluso podría llegar a creerse que una sola persona puede encontrar dificultades para manejar todas esas responsabilidades o no tener todas las características, pero vale la pena destacar que nada dice que se debe realizar esto en soledad, sino que por el contrario es siempre deseable realizarlo en colaboración con el equipo y/o stakeholders.

Siguiendo los conceptos de Kenneth Rubin podemos expresar las siguientes áreas de responsabilidad:

Gestión económica

El Product Owner es responsable de asegurarse que se tomen buenas decisiones económicas, sabiendo aprovechar los recursos de la organización para maximizar la entrega de valor a nivel release y sprint.

- *A nivel Release:* Este punto es de fundamental importancia para definir los trade-off ("sacrificios") en alcance, fecha, presupuesto y calidad a medida que se obtiene información de avance y de contexto. Así como también definir el alcance del release, balanceando entre magnitud del gasto e impacto al negocio.
- *A nivel Sprint:* Además de la economía a nivel release, el Product Owner es el encargado de asegurarse que un buen retorno de la inversión (ROI) es devuelto al final de cada sprint.



Atacando siempre lo más prioritario o valioso. Conociendo el costo que tiene cada Sprint (tiempo y recursos fijos) debe analizar si es la mejor manera de invertir el dinero.

Planificación

El Product Owner es un participante clave en la planificación, debe clarificar las dudas que puedan surgir, llevar información o datos que se tengan sobre el uso y comportamiento del producto así como también transmitir claramente las prioridades de la organización. De estos eventos, debe llevarse la información resultante para compartir con los interesados.

Refinamiento, priorización y mantenimiento del backlog

Estas tareas pueden ocupar un gran porcentaje del tiempo, en ellas se construyen los ítems del Backlog del producto, refinan, negocian, se validan hipótesis y se priorizan. Esto cabe destacar que no significa que el escriba todos los ítems, pero si es el responsable final por el contenido.

Definición de criterio de aceptación y verificar su cumplimiento

El establecer criterios bajo los cuales serán validados los ítems facilita entender, preparar las pruebas que se realizarán, requerimientos técnicos, etc. La existencia de estos criterios son mandatorios al momento de planificar un ítem del product backlog en un sprint, y al finalizar el ítem debe validar su cumplimiento. Cabe destacar que a medida que un equipo madura, es posible que se relajen las restricciones, pero esto no es recomendable hasta ganar ritmo.

Colaborar con el equipo de elaboración

Como vimos al momento de repasar el manifiesto ágil, la colaboración con el equipo es fundamental, y un buen Product Owner debe tener una comunicación fluida con el equipo de scrum, aclarando dudas, informando y trabajando colaborativamente en el refinamiento. A diferencia del enfoque tradicional donde es posible que los esfuerzos se localizan principalmente al inicio y al final, en los marcos ágiles el esfuerzo es constante.

Colaborar con los interesados (stakeholders)

Siendo el Product Owner la voz del usuario, es su responsabilidad coordinar con los interesados internos y externos para entender sus necesidades y coordinar con el equipo de elaboración los trade-off que sean necesarios para la entrega de valor y comunicarlos para mantener transparencia y las expectativas alineadas.

Características de un Product Owner

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



En general la persona a ocupar este rol debe ser un visionario con capacidad para guiar al equipo para cumplir ese propósito. Esto requiere de un set bastante variado de características, las cuales siguiendo la línea de las propuestas de Rubin¹⁰ y Pichler¹¹ se pueden agrupar en las siguientes categorías:

Habilidades en el dominio

Es difícil ser un Producto Owner efectivo siendo nuevo en el dominio (y contexto), poder definir prioridades y/o adelantarse a riesgos y validaciones. Si no conoce puede tornarse en una tarea llena de incertidumbre, por eso es valioso que la persona con el rol posea experiencia que facilite la toma de decisiones y si conoce también el contexto en el que se encuentra podrá moverse con mayor facilidad. Igualmente, un PO valida sus decisiones con el comportamiento del usuario real y se adapta en consecuencia de manera de ganar certidumbre e incluso adaptarse en dominios no tan conocidos.

Habilidades de comunicación y negociación

Ser la voz del usuario, cual requiere de una buena relación con los stakeholders (los cuales pueden tener conflicto en las prioridades y requerimientos) y el equipo de elaboración, para lo cual debe tener una participación activa en la comunicación y negociación con el objetivo de obtener compromisos de manera clara generando confianza, transparencia y consenso. Esto lo beneficia teniendo múltiples puntos de vista y experiencias que generan mayor compromiso y robustez en el proceso.

Poder de decisión

Un Product Owner (justamente como su nombre lo indica) debe tomar "propiedad" del producto en cuestión y tener poder de decisión para definir y tomar decisiones. Si bien debe buscarse consenso, en ocasiones debe ser el encargado de establecer definiciones de acuerdo a lo que considera más prioritario o mediar para desatar conflictos de intereses y/o de alcance. También es el encargado de mantener un balance entre el negocio y el lado técnico para lograr maximizar la entrega de valor de manera sostenible.

Responsabilidad y compromiso

El Product Owner es el principal responsable de que los recursos se están utilizando de manera eficiente para entregar valor al usuario frecuentemente. Si bien esto no absuelve a los demás

10 Essential Scrum: A Practical Guide to the Most Popular Agile Process

11 Agile Product Management with Scrum: Creating Products that Customers Love

miembros del equipo scrum, es en el PO en quien cae esta responsabilidad, ya que él en su rol puede repriorizar y refinar el backlog para cada sprint e incluso cancelar el desarrollo por completo.

Entusiasta y crítico

Para cumplir el rol de manera efectiva, debe tener entusiasmo y motivación, sumados a una visión de un producto que crea que realmente beneficia al usuario y cliente y vale la pena invertir en él. Compartiendo esto con los involucrados para generar una activa participación y energía. La contracara de esto es que demasiado entusiasmo puede hacer que se tapen problemas u oportunidades, por lo tanto, la cuota de crítica es fundamental para el equilibrio.

Informado e intuitivo

Un buen Product Owner debería estar constantemente testeando las afirmaciones y procesando el feedback (en lugar de opiniones) para la toma de decisiones. Este accionar baja el riesgo y evita depender solo de su opinión e ideas. Obviamente siempre podemos tomar decisiones luego de probar cada opción, y como hemos visto en la historia de los productos exitosos, estos deben complementarse con intuición e imaginación para nuevas ideas.

Tipos de Product Owner

De acuerdo al nivel de responsabilidades asignadas por la organización, Roman Pichler realiza una diferenciación en dos tipos de PO que me resulta muy práctica: El "Gran Product Owner" y el "Pequeño Product Owner". Esta diferenciación por "tamaño" tiene que ver con los niveles de propiedad del producto, siendo un "gran" Product Owner realmente un "dueño" de todo el producto (visión, estrategia, presupuesto, etc.) en su máxima expresión, mientras que un "pequeño" Product Owner es más bien un responsable de los detalles del producto, cumpliendo una función más operativa.

La visión describe la razón o motivo por el cual existe el producto, la estrategia cubre la estrategia seguida para el producto (su roadmap y modelo de negocio), mientras que la táctica se refiere a los detalles del producto tales como épicas e historias de usuario, mock-ups, escenarios, diagramas y demás ítems que generalmente se incluyen en el backlog. Esta distinción, si bien no es nueva, es importante destacarla debido a lo habitual que se ha vuelto hacer que las asignaciones de Product Owner hagan foco solo en la táctica, lo cual no sería particularmente problemático si alguien más se ocupa de la visión y estrategia y se trabajara con una comunicación muy fluida.



Lamentablemente este no es siempre el caso y sin darnos cuenta podemos estar dejando librado al azar parte de las responsabilidades.

La ventaja de ser un gran Product Owner es que controla todos los aspectos del producto, al no haber necesidad de traspasos (hand-off) debería existir más consistencia en las decisiones. La dificultad se encuentra en el desafío que representa para una persona tomar todas estas responsabilidades ya que requiere de una variedad grande de habilidades y una carga considerable en casos de productos grandes y/o en gran crecimiento.

Ser un pequeño Product Owner permite enfocarse en el Product backlog y los detalles del producto, esto también puede ayudar a evitar el exceso de carga. Pero las responsabilidades de llevar adelante la estrategia y visión deben ser tomadas por alguien y tener una colaboración muy efectiva para alinear con la táctica, de otro modo se puede sufrir los clásicos problemas de los traspasos (hand-off): tener demoras, pérdida de conocimientos, decisiones inconsistentes, confusiones y solapamientos.

Product Owner	Beneficios	Contras
Grande	Rapidez y consistencia en la toma de decisiones	Requiere una diversa variedad de habilidades y conocimientos. Potencialmente gran carga laboral.
Pequeño	Mayor foco y grado de especialización en las tareas	Transferencias de conocimientos y objetivos. Potenciales retrasos y pérdida de conocimiento Inconsistencia en la toma de decisiones.

Como puede verse ambas variantes de rol tienen ventajas y desventajas, y depende de la necesidad la conveniencia de cada uno. Suele entenderse que un Product Owner “grande” es beneficioso cuando tenemos un producto joven que está evolucionando rápidamente. Una vez maduro y/o estable suele ser útil la utilización de un Product Owner “pequeño” para compartir la responsabilidad de carga laboral y facilitar el escalamiento. Cabe destacar que la utilización de esta desagregación requiere de una fuerte comunicación entre las personas que ocupan estos cargos, recordando siempre que la estrategia debe dirigir la táctica, y esta última influenciar a la primera.

Product Owner vs Product Manager

Es común escuchar estos términos por separado, no sabiendo cuáles son las responsabilidades de cada uno; si son roles completamente distintos, lo mismo o posiciones jerárquicas.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Nos limitaremos a decir que a nivel del curso los veremos cómo sinónimos, sabiendo que el puesto original era de justamente realizar la gestión del producto (Product Manager) que llevado a scrum es asimilado por las responsabilidades dadas al rol de Product Owner y con la gran popularidad del marco consideramos llevó a la incorporación del título como más común incluso sin usar scrum.



6. Escalamiento del Rol de Product Owner

Como vinimos hablando en puntos anteriores, el Product Owner es un rol que se espera cumpla una persona, pero en la práctica cuando tenemos un producto grande y complejo es normalmente un esfuerzo compartido. Esto nos lleva a pensar en cómo resolver esto sin perder cohesión o inconsistencia, la cual en línea con los autores antes mencionados^{12 13} podemos mencionar las siguientes prácticas de acuerdo al tipo de situación en la que nos encontramos.

Escalando con el ciclo de vida del producto

La etapa en que se encuentra el producto suele ser un buen indicador de si requerimos escalar el rol. Es natural que en un producto nuevo que aún no alcanzo un punto de salida al mercado y/o está en un nivel más bien de prueba conceptual, lo más efectivo es que una sola persona cumpla con el rol. La razón es que estos productos tienden a requerir experimentación y retrabajo para poder adaptarlo luego del feedback y poder lanzarlo al mercado.

Este motivo requiere de una toma de decisiones efectiva y el tener una sola persona ocupando el rol facilita esto. El hecho de que estos productos suelen tener equipos de elaboración relativamente chicos lo vuelven más viable.

Una vez que el producto empieza a crecer y suma mayor cantidad de funcionalidades y crece el equipo de elaboración requerido para realizar el trabajo una sola persona normalmente empieza a tener dificultades para realizar un buen análisis y mantener la comunicación efectiva con todos los involucrados. Además, durante esta etapa de crecimiento se suele agregar cosas sin hacer cambios radicales en el funcionamiento core del producto por lo tanto compartir las responsabilidades se vuelve más fácil.

Escalando por dueños de funcionalidades y componentes

Una vez que comprobamos la viabilidad del producto y empieza a crecer, es común que podamos diferenciar con cierta facilidad una arquitectura con los bloques que lo componen, teniendo ya la relación que cada bloque tiene con el resto para conformar un producto, una persona puede ser asignada a cada bloque en particular independizando un tanto la evolución del mismo como si

12 <http://www.romanpichler.com/blog/scaling-the-product-owner/>

13 <https://www.mountaingoatsoftware.com/blog/the-chief-product-owner-on-large-agile-projects>



fueran un producto en sí mismo, dejando a un Product Owner que tenga la visión general. Esto lo podemos ver en productos como Facebook (que separaron la administración de imágenes, muro, etc.), así como también en Spotify por nombrar algunos. En la industria financiera esta la separación del producto, “prestamos”, préstamos personales, préstamos hipotecarios, prestamos inmediatos, otros.

En la industria de la construcción, la construcción de un barrio privado de viviendas está separada de la construcción individual de cada casa. En un edificio hace un tiempo se está empezando a ver la construcción, por un lado, y liberando la construcción interna de cada departamento dando opción a la definición de la estructura interna, divisiones, pisos, muebles, cerámicas, etc, a cada propietario. En publicidad está el responsable de una cuenta, que a su vez puede tener varios responsables de producto para cada una de sus líneas, por ejemplo, Nike, tiene una variedad de productos con una campaña central y una individual para sus productos.

Escalando por variantes del producto

Otro enfoque para escalar el producto es dividir el producto. Esto puede hacerse por haber tenido un crecimiento muy grande que genera la necesidad de crear un nuevo producto a partir de él que reduce el alcance del producto original con un nuevo Product Owner y equipo de elaboración (por ejemplo, como Facebook hizo con su aplicación Messenger). Así como también puede hacerse para cubrir distintos nichos del mercado (como ha realizado Apple con el Ipod en sus versiones Shuffle y Touch).



7. ¿Quién debería ocupar el rol según el tipo de proyecto?

La selección de la persona a ocupar el rol es una tarea determinante en el resultado del proyecto, de acuerdo al tipo de contexto que estemos manejando autores como Roman Pitcher mencionan las siguientes opciones:

Desarrollos internos

En estos casos donde el objetivo es satisfacer un requerimiento interno de la organización con un equipo propio, es deseable seleccionar una persona del grupo que se beneficiará del desarrollo, usualmente alguien del negocio que tiene conocimientos del día a día de la operación y poder de decisión e influencia para concentrar las distintas visiones y priorizar entre ellas. En los casos de organizaciones con estructuras más tradicionales o que no tienen una participación activa de los usuarios (lo cual es algo no deseable) es posible asignar alguna persona las responsabilidades operativas del rol de Product Owner relevando y facilitando la generación de un backlog priorizado con las necesidades que van siendo detectadas.

Desarrollos comerciales

En los casos donde desarrollamos para un cliente externo que compra nuestro servicio, el Product Owner debería ser una persona de la organización que actúa como la voz de los clientes reales y esa persona puede hacer las priorizaciones y compromisos que fuesen necesarios acompañado por el equipo para asistirlo en las tareas del rol. Es frecuente que se asigne una persona de la empresa contratista para suplir al cliente en el rol y poder trabajar de manera directa con el equipo de elaboración abstrayendo de las potenciales demoras o desconocimientos técnicos. Si es el caso debemos tener en cuenta esta situación (véase PO Proxy).

Desarrollo tercerizado

En este tipo de desarrollo en el cual se contrata un tercero para la construcción, tenemos justamente la versión opuesta a la anterior, y en este caso un representante de la compañía contratante debería ser asignado como Product Owner. Al igual que en el anterior la compañía contratada podría asignar una persona que interactúe de manera directa y liberando bloqueantes, pero no delegando esta responsabilidad.



Este funcionamiento puede ser un tanto problemático en los casos de contrato a precio fijo, en los cuales el empleado de la empresa contratada puede sentirse en la responsabilidad de tomar el rol para controlar el alcance.

Desarrollo de componentes

Algunas organizaciones pueden tener una estructura orientada a componentes, las cuales construyen solo partes de la solución. Estas partes deben ser integradas con otras desarrolladas por otros equipos para dar una solución completa al usuario. El foco de estos equipos es el componente técnico, por lo tanto, sus Product Owners son típicamente personas de perfil técnico que son capaces de definir y priorizar estos tipos de ítems. En estos casos se requiere un tanto más de esfuerzo en la gestión a entre equipo para la integración y sincronización.



8. Problemas comunes

Al momento de implementar Scrum en las organizaciones pueden darse múltiples problemas, vamos a categorizar los más comunes que indica Pichler¹⁴ y que también fuimos experimentando en la práctica:

Product Owner sin poder de decisión

Este tipo de Product Owner carece de autoridad, lo cual puede deberse a diversos motivos, por ejemplo; no tiene la atención en la gestión, no tiene la confianza de la gerencia, recibió el rol delegado del responsable solo para seguimiento, pero sin la responsabilidad, etc. La consecuencia es que se dificulta cumplir el rol de manera efectiva para alinear al equipo, transmitir la visión, gestión de stakeholders o la priorización o exclusión de requerimientos del backlog del release.

Product Owner sobre-exigido

Este tipo de Product Owner rápidamente se convierte en un cuello de botella y frena el avance del proyecto. El síntoma más común es no estar disponible para atender sus responsabilidades con el equipo de elaboración, participando de las ceremonias, clarificando dudas y compartiendo la visión que tiene del producto. Una causa común de esta falta de disponibilidad se da cuando el rol Product Owner para ese producto es solo una de tantas tareas asignadas compitiendo por tiempo y atención o cuando el Product Owner tiene a su cargo demasiados productos o equipos. En este caso por un lado se debe entender a nivel organizacional las implicancias del rol y la carga que tiene para tener una asignación acorde, y por otro el equipo también debe recordar la necesidad de autogestionarse para no requerir por demás del PO asumir su participación en la propiedad del producto.

Product Owner Parcial

En muchas organizaciones se dividen las responsabilidades del PO entre varias personas, una manera común en la suele dividirse es teniendo una jerarquía, teniendo alguien encargado de la visión y estrategia y por debajo alguien encargado de la táctica o día a día del equipo (describimos esto en detalle con anterioridad), si bien útil suele acarrear problemas en el traspaso (hand-off) y en la falta de autoridad (que ya describimos).

Product Owner Distante

14 Libro: Agile product management with scrum, Autor: Roman Pichler



El Product Owner puede estar distante del equipo de distintas maneras, en otra habitación del mismo edificio, en otra localidad, país o incluso continente. Pero el impacto puede ser el mismo en todos los casos, en los cuales no existe confianza, alineamiento, se pierde información y resulta en avance lento o pobre (si lo hay). En IT es práctica común hasta en la actualidad el envío de definiciones documentadas a un equipo de construcción ubicado en cualquier otro lugar del mundo esperando que luego de eso se obtenga software funcionando tal como se esperaba. Si recordamos uno de los principios ágiles dice "La manera más efectiva y eficiente de comunicarse en cara a cara". La presencia, confianza y buena comunicación cara a cara es fundamental para alcanzar los objetivos de manera eficiente.

Product Owner Proxy

Este tipo de Product Owner se da cuando la persona cumpliendo las tareas del rol no es realmente la persona que toma estas decisiones ya sea por tener un Product Owner distante, sobrecargado, parcial, no participativo, etc. La persona asignada como "proxy" puede ser muy aplicada en sus tareas, pero esta separación suele traer acarreadas demoras en las decisiones, retrabajo ante diferencia de opiniones o criterios, malas comunicaciones o fallas en la priorización lo cual impacta en menor productividad, desmotivación y conflictos.

Comité de Product Owner

Este tipo de caso sucede cuando tenemos un grupo de Product Owner sin nadie a cargo del producto en sí que se encargue de generar una meta común y facilitar la toma de decisiones. Hay una vieja frase que dice "si se quiere que algo no se haga, creen un comité", y en este caso aplica ya que este tipo de Product Owner suele resultar en reunión eternas con conflictos de intereses constantes, idas y vueltas que implican que en general no haya un avance real, y la estrategia pueda ser cambiante generando desmotivación y perdida de la visión (si es que alguna vez existió).

Product Owner sin conocimiento de las técnicas de construcción

En las metodologías agiles insistimos por la inclusión del negocio en los equipos, por lo tanto la participación de ellos en los mismos no debe verse como un problema sino como una oportunidad que el equipo debe aprovechar y asumir el compromiso de colaborar con él para facilitar la aplicación de las prácticas requeridas para la generación de las historias y mantenimiento del backlog con el coaching correspondiente (muy importante aquí el scrum master/agile coach en el acompañamiento).

Product Owner sin conocimiento del negocio

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Este caso es más grave que el anterior, suele darse cuando se generan nuevos equipos y el rol se asigna sin un claro conocimiento de la implicancia del rol, a alguien que no posee las características necesarias y particularmente la manera de trabajo del usuario. En este caso puede mitigarse teniendo contacto fluido con las partes del negocio siendo su funcionamiento más similar a la del tipo proxy



Bibliografía utilizada y sugerida

Pichler, Roman, Agile Product Management with Scrum; 1ra Edición; USA; Addison-Wesley; 2010.

Pichler, Roman, Strategyze; 1ra Edición; USA; Addison-Wesley; 2016.

Rubin, Kenneth, Essential Scrum; 1ra Edicion; USA; Addison-Wesley; 2012.

Schwaber, Ken, Agile Project Management with Scrum, 1ra Edición; USA; Microsoft Press; 2004.

Shore, James, The Art of Agile Development, 1ra Edición; USA; O'Reilly, 2008.

Lo que vimos:

En esta Unidad hicimos foco en el conocimiento del paradigma ágil y el marco de trabajo Scrum, para luego detallar las responsabilidades del Product Owner (dueño del producto), sus tareas y problemas más comunes.





Lo que viene:

En la próxima Unidad profundizaremos en el establecimiento de la estrategia, buenas prácticas y herramientas para iniciar el proceso de construcción del producto.

