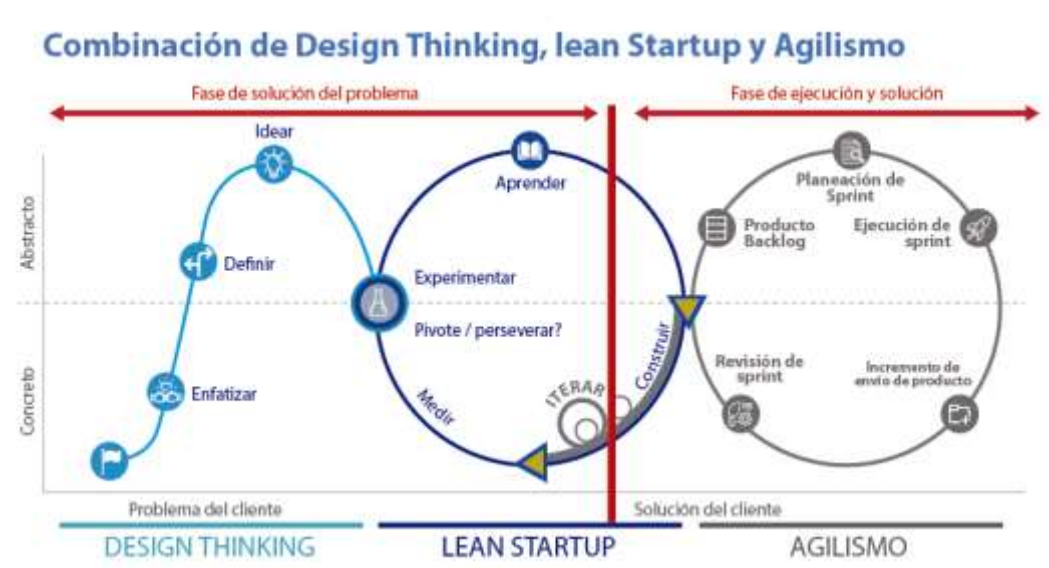
Módulo 4. Metodologías ágiles y *mindset* de experimentación

Unidad 4.1 Metodologías ágiles

Para comenzar, veamos qué es una metodología ágil.

> *Agile* es mucho más que una metodología para el desarrollo de proyectos que precisan de rapidez y flexibilidad. Es una filosofía que supone una forma distinta de trabajar y de organizarse, de tal forma que cada proyecto se divide en pequeñas partes, que tienen que completarse y entregarse en pocas semanas. El objetivo es desarrollar productos y servicios de calidad, que respondan a las necesidades de clientes, cuyas prioridades cambian a una velocidad cada vez mayor. (Tena, s.f., https://bbva.info/3SGHPOJ)

Figura 1: Metodologías ágiles



Fuente: Deloitte, 2021, https://bit.ly/3r91HhN

Cuando hablamos de metodologías ágiles no debemos limitarnos a pensar en una simple herramienta, sino en una estrategia integral que impulsa a las organizaciones a **gestionar los proyectos con rapidez y flexibilidad**.

La realidad es que el mercado cada día exige mayor flexibilidad ante un panorama incierto y cambiante, y las empresas deben responder con urgencia esta demanda.

La metodología *agile* ayuda en el desarrollo de proyectos que necesitan rapidez y flexibilidad para adecuarse a las necesidades del cliente. Siempre está enfocada a mejorar resultados.

A diferencia de la forma tradicional de gestionar los proyectos, las metodologías ágiles no necesitan definir al inicio de los proyectos la totalidad del alcance.

En efecto, definimos a las metodologías ágiles como una innovadora forma de trabajar y organizarse que "fragmenta" los proyectos en partes capaces de adaptarse sobre la marcha, complementarse y resolverse en poco tiempo. Es decir, no se planifica ni se diseña el proyecto por adelantado, sino que a medida que se desarrolla se va definiendo el proyecto, gracias a un *feedback* constante.

Otra característica muy particular es que se trabaja por periodos de tiempo durante el cual cada miembro del equipo debe ejecutar una serie de tareas.

Luego de ejecutar dichas tareas, se entregan los avances, se reciben devoluciones y comienza nuevamente el proceso, permitiendo implementar los cambios necesarios. (Aden, s.f., https://bit.ly/3RiJjh0)

Los 12 principios del Manifiesto Ágil

Los creadores del Manifiesto Ágil dispusieron de 12 principios para definir la filosofía de las metodologías ágiles.

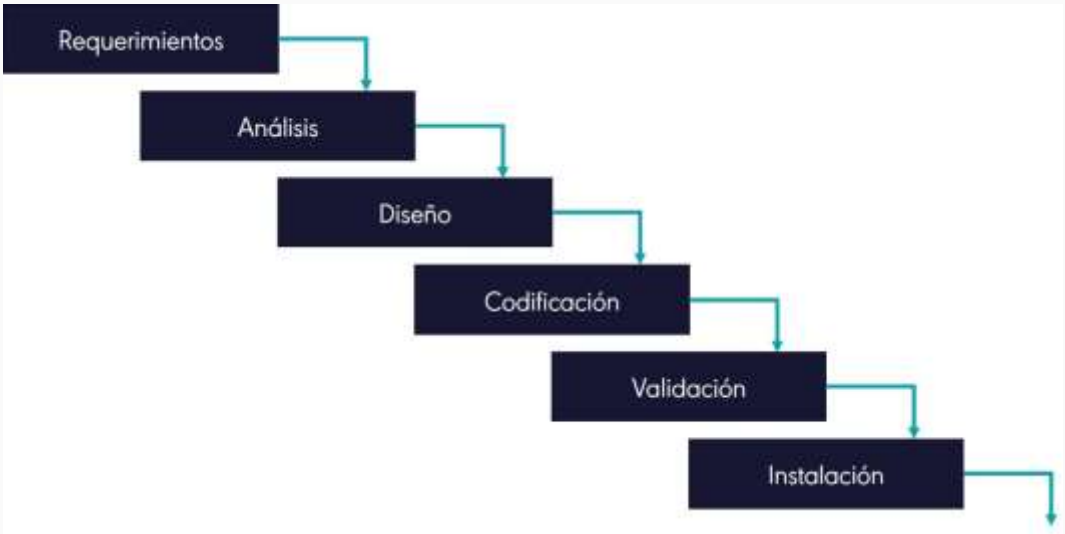
1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de *software* con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

3. Entregamos *software* funcional 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 *software* 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. (Manifiesto Ágil, s.f., https://bit.ly/30nXaIS)

Desarrollo tradicional

El desarrollo tradicional es conocido como el **desarrollo de cascada**, donde cada etapa inicia cuando se termina la anterior. Esta metodología no contempla la posibilidad de medir los resultados hasta el final del proyecto. Por un lado, porque recién ahí se dispondrá de un entregable. Por el otro, porque no considera necesario realizar ninguna medición antes del lanzamiento. Como dijimos, cada etapa comienza cuando se termina la anterior.

Figura 2: Desarrollo en cascada



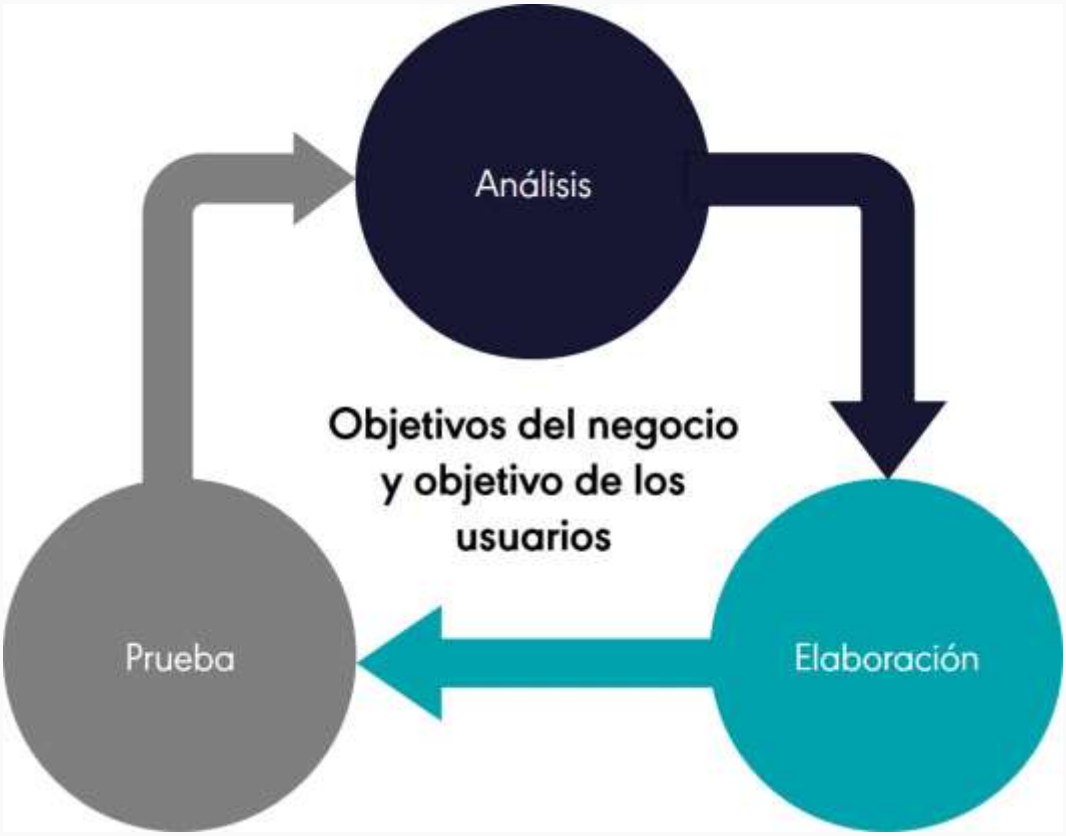
Fuente: elaboración propia

Desarrollo ágil

A diferencia del desarrollo en cascada, donde existe una única etapa de relevamiento, el desarrollo ágil posee tantas instancias de relevamiento como iteraciones.

En cada iteración se podrá revisar lo relevado en la anterior, a partir de los hallazgos en las pruebas realizadas. De esta forma, refinamos nuestro proyecto.

Figura 3: Desarrollo ágil



Fuente: elaboración propia

El desarrollo ágil propone un abordaje **iterativo**. Cada **iteración** se compone de tres instancias:

1. Análisis
2. Elaboración
3. Prueba/testeo

Cada proyecto puede componerse de determinada cantidad de iteraciones, de acuerdo con su complejidad. De esta manera, se puede ejecutar una prueba de usabilidad cada vez que se lo necesite, para ir validando el avance y con ello ir disminuyendo el riesgo del producto final.

Ventajas de la metodología ágil

En primer lugar, "es importante entender que las metodologías ágiles nacen para privilegiar la interacción en los procesos y facilitar la colaboración con el

cliente ante la necesidad de cambios durante el armado del proyecto" (Aden, s.f., https://bit.ly/3RiJjh0). Sabiendo esto, ahora sí veamos cuáles son las ventajas:

1. **Entregas rápidas y continuas.** Uno de los aspectos más importantes de las metodologías ágiles es que tienen como principal característica realizar entregas rápidas y continuas de *software* funcionando.
2. **Conciben al proyecto en partes homogéneas**. La capacidad de dividir al proyecto en partes capaces de adaptarse sobre la marcha, complementarse y resolverse en poco tiempo, ayuda a que, si hay que realizar cualquier modificación, solo se hacen cambios en la parte implicada y en poco tiempo.
3. **Promueven el trabajo colaborativo.** Además de los beneficios en los procesos, también fomenta el trabajo multidisciplinario, la autonomía y transparencia. Como todas las partes persiguen un objetivo en común y trabajan con fluidez y flexibilidad, permite que los equipos obtengan resultados más efectivos.
4. **Predicen el resultado y minimizan riesgos**. Gracias a las revisiones continuas y la adaptación al cambio, permite obtener una mirada predictiva sobre el resultado y esto por descarte minimiza los riesgos de cometer errores inmodificables.
5. **El cliente es un miembro más del equipo**. Claro está que, gracias a un vínculo fluido con los clientes y un trabajo multidisciplinario, se consiguen resultados realmente satisfactorios. El cliente se convierte en un miembro más del equipo y, de este modo, se logran proyectos

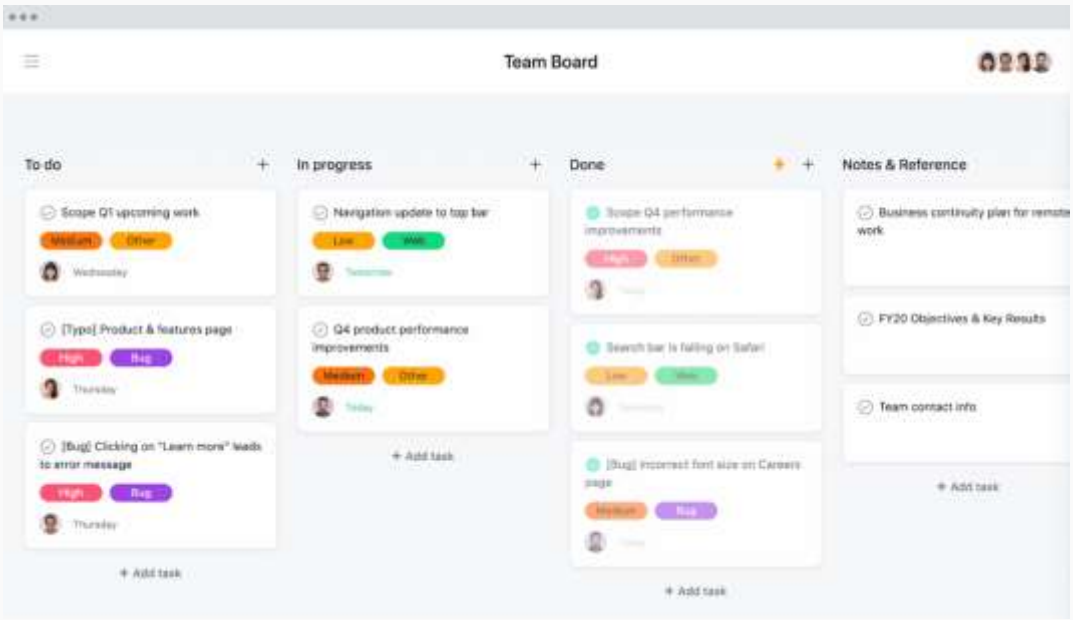
eficientes y con una gran experiencia de los clientes. (Aden, s.f., https://bit.ly/3RiJjh0)

Metodología *kanban*

Kanban es una forma de ayudar a los equipos a encontrar un equilibrio entre el trabajo que necesitan hacer y la disponibilidad de cada miembro del equipo. La metodología *kanban* se basa en una filosofía centrada en la mejora continua, donde las tareas se «extraen» de una lista de acciones pendientes en un flujo de trabajo constante.

La metodología *kanban* se implementa por medio de tableros *kanban*. Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo. En un tablero *kanban*, el trabajo se muestra en un proyecto en forma de tablero organizado por columnas. Tradicionalmente, cada columna representa una etapa del trabajo. El tablero *kanban* más básico puede presentar columnas como «Trabajo pendiente», «En progreso» y «Terminado». Las tareas individuales —representadas por tarjetas visuales en el tablero— avanzan a través de las diferentes columnas hasta que estén finalizadas. (Martins, 2022, https://bit.ly/3SiyooV)

Figura 4: Tablero *kanban*



Fuente: Martins, 2022, https://bit.ly/3SiyooV

> Hoy en día, los tableros *kanban* son en su mayoría tableros virtuales con columnas que representan las etapas del trabajo (aunque aún se pueden dibujar tableros *kanban* en pizarras blancas y dar seguimiento al trabajo con *post-it*). En un tablero, una tarjeta *kanban* representa una tarea, y esta tarjeta de tarea avanza a través de las etapas del trabajo a medida que se finaliza. Los equipos que usan un sistema *kanban* tienden a colaborar en un único tablero, aunque las tareas generalmente se asignan a miembros individuales del equipo. (Martins, 2022, https://bit.ly/3SiyooV)

Entonces, para usar esta metodología hay que crear un tablero visible y accesible para todos los miembros del equipo. En las columnas se anotará el estado del flujo de las tareas, siendo necesarias tantas como estados de las tareas existan desde su comienzo hasta su finalización. Así se determinará el estado de cada proyecto.

Al ser un método visual permite que con un simple vistazo se conozca el estado de los proyectos y se puedan asignar nuevas tareas de manera muy efectiva.

Es un tablero continuo en el que las tarjetas no se desplazan; sino que, a medida que se avanza por él, las nuevas funcionalidades, mejoras o incidencias se acumulan al inicio. Así, pueden priorizarse y colocarse en las secciones más oportunas.

Este método se basa en el desarrollo incremental, es decir, en la división del trabajo en diferentes partes. Por lo tanto, no se habla de una tarea en sí, sino que se agiliza el proceso de producción al dividir el trabajo en distintos pasos.

Cada tarjeta, *post-it* o casilla en la hoja de cálculo que se utilice para aplicar el método se añade al tablero en la fase que corresponda. Allí se añade la información necesaria para que el equipo conozca la carga total de trabajo que le va a suponer. Se describe la tarea y cualquier tipo de observación.

De esta forma, el trabajo es visualizado correctamente por cada departamento, así como las prioridades y objetivos. De nuevo, al tratarse de un método tan visual se gana en eficacia a todos los niveles. (Redacción APD, 2021, https://bit.ly/2Hp67JO)

Ventajas de *kanban*

En función de lo mencionado anteriormente, podemos decir que las ventajas de la metodología *kanban* son las siguientes:

1. **Transparencia**. Los tiempos de entrega son más cortos y hay una mayor fiabilidad en estos. Todo el mundo sabe cuál es su tarea y en qué momento está de su ciclo.

2. **Evita tareas ineficientes.** Se evita la sobreproducción y la limitación de los recursos, lo que supone contar con una mayor disponibilidad de materiales.
3. **Control de tareas.** El tiempo de producción es más rápido, por tanto, se reduce el control del esfuerzo y se mejora la planificación. Esto afecta directamente a la mayor productividad en el área de compras, abastecimiento y control. Aumenta la rotación de los inventarios y se necesita una menor capacidad de almacenamiento.
4. **Flexibilidad.** Como todo el equipo sabe perfectamente cuál es su tarea y la realiza con eficacia, si surge alguna tarea imprevista existe una capacidad de respuesta que permite atenderla. (Redacción APD, 2021, https://bit.ly/2Hp67JO)

En definitiva:

> La metodología *kanban* está indicada para las empresas que necesiten cierta flexibilidad a la hora de manejar nuevas entradas de tareas y poder realizar un buen seguimiento de estas. Además, permite priorizar, realizar informes precisos y supervisar adecuadamente el trabajo en equipo. (Redacción APD, 2021, https://bit.ly/2Hp67JO)

Metodología *scrum*

> *Scrum* es un *framework* que se utiliza dentro de equipos que manejan proyectos de alta incertidumbre. Se trata de un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregan productos del máximo valor posible productiva y

creativamente. *Scrum* es liviano y fácil de entender, pero, a la vez, difícil de dominar del todo. Este *framework* favorece el *time to market* y la entrega rápida de MVP (mínimos productos viables).

El Manifiesto Ágil y *scrum* van de la mano para ofrecer una guía clara a los equipos de desarrollo a la hora de afrontar un nuevo proyecto. En este sentido, hay una serie de valores que se asocian al *framework scrum* y que vale la pena tener en consideración:

- ✓ **Coraje**. Tener la valentía para hacer lo correcto y resolver problemas complejos.
- ✓ **Foco**. La organización del trabajo en *sprints* da certidumbre y permite centrar los esfuerzos en objetivos alcanzables.
- ✓ **Compromiso**. Para que el *framework scrum* funcione necesita el compromiso de todos los miembros involucrados.
- ✓ **Sinceridad**. El equipo *scrum* debe de ser transparente en cuanto al trabajo y los obstáculos que encuentra en el camino.
- ✓ **Respeto**. Tratarse de forma adecuada entre compañeros y valorar las ideas y esfuerzos de cada uno es imprescindible. (Ángel, 2022, https://bit.ly/3r8WxlT)

Los roles del equipo *scrum*

Si seguimos los valores y principios *scrum*, el equipo tiene como foco entregar valor y ofrecer resultados de calidad que permitan cumplir los objetivos de negocio del cliente.

Para ello, los equipos de *scrum* son autoorganizados y multifuncionales. Es decir, cada uno es responsable de unas tareas determinadas y de terminarlas en los tiempos acordados. Esto garantiza la entrega de valor del equipo completo, sin necesidad de ayuda o la supervisión minuciosa de otros miembros de la organización. (Ángel, 2022, https://bit.ly/3r8WxlT)

A continuación, desarrollaremos los tres roles más importantes en un equipo de trabajo *scrum*.

Product owner

Es el responsable de **maximizar el valor del trabajo** que viene de la mano de una buena gestión del ***product backlog***.

El *product owner* es el único perfil que habla constantemente con el cliente, lo que le obliga a tener muchos conocimientos sobre el negocio.

Para finalizar, **un equipo scrum debe tener solo un *product owner*** y este puede ser parte del equipo de desarrollo. (Ángel, 2022, https://bit.ly/3r8WxlT)

Scrum master

Es el **responsable de que las técnicas *scrum* sean comprendidas y aplicadas** en la organización. Es el ***manager* de *scrum***, un líder que se encarga de eliminar impedimentos o inconvenientes que tenga el equipo dentro de un *sprint*, aplicando las mejores técnicas para fortalecer el equipo.

Dentro de la organización, el *scrum master* tiene la labor de ayudar en la adopción del *framework* en todos los equipos. (Ángel, 2022, https://bit.ly/3r8WxlT)

Equipo de desarrollo

Son los encargados de **realizar las tareas priorizadas por el *product owner***. Es un equipo multifuncional y autoorganizado. Son los únicos que estiman las tareas del *product backlog*, sin dejarse influenciar por nadie.

Los equipos de desarrollo no tienen subequipos o especialistas. La finalidad de esto es transmitir la responsabilidad compartida si no se llegan a realizar todas las tareas de un *sprint*. (Ángel, 2022, https://bit.ly/3r8WxlT)

Eventos del equipo *scrum*

Figura 5: Eventos del equipo *scrum*

Timeboxes ceremonias scrum

Ceremonia	1 semana	2 semanas	3 semanas	4 semanas
Sprint Planning	2 horas	4 horas	6 horas	8 horas
Daily Scrums	15 minutos	15 minutos	15 minutos	15 minutos
Sprint Review	1 hora	2 horas	3 horas	4 horas
Sprint Retrospective	45 minutos	1:30 horas	2:15 horas	3 horas
Grooming	Entre el 5% y el 10% del tiempo disponible del sprint			

Fuente: Ángel, 2022, https://bit.ly/3r8WxlT

Este *timebox*:

> Refleja las diferentes ceremonias *scrum* y su peso en horas según se avanza en semanas de trabajo. El desarrollo iterativo se realiza en un *sprint* que contiene los siguientes eventos *scrum*: *sprint planning*, *daily meeting*, *sprint review*, *sprint retrospective* y *grooming*. (Ángel, 2022, https://bit.ly/3r8WxlT)

Antes de profundizar en los diferentes eventos, es importante definir qué es un *sprint* en *scrum*.

> El *sprint* es el corazón de este marco de trabajo, el contenedor de los demás hitos del proceso. Todo lo que ocurre en una iteración para entregar valor está dentro de un *sprint*. La duración máxima es de un mes, el tiempo se determina con base en el nivel de comunicación que el cliente quiere tener con el equipo. **Los *sprints* largos pueden hacer que se pierda *feedback* valioso** del cliente y poner en peligro el proyecto. (Ángel, 2022, https://bit.ly/3r8WxlT)

Sprint planning

> En esta reunión, todo el **equipo *scrum* define qué tareas se van a abordar y cuál será el objetivo del *sprint***. La primera reunión que se hace en el *sprint* puede llegar a tener una duración de 8 horas para *sprints* de un mes.

El equipo se hace las siguientes preguntas:

- ¿**Qué se va a hacer en el *sprint***? Con base en ello, se eligen tareas del *product backlog*.
- ¿**Cómo lo vamos a hacer**? El equipo de desarrollo define las tareas necesarias para completar cada ítem elegido del *product backlog*.

La definición de qué se va a hacer implica que el equipo tenga un objetivo y se encuentre comprometido con la entrega de valor que se hará al cliente al final del *sprint*. A esto se le llama ***sprint goal***.

El resultado de esta reunión es el *sprint goal* y un ***sprint backlog***. (Ángel, 2022, https://bit.ly/3r8WxlT)

Daily meeting

Es una reunión diaria dentro del *sprint* que tiene como **máximo 15 minutos de duración**. En ella deben participar, sí o sí, el equipo de desarrollo y el *scrum master*. El *product owner* no tiene la necesidad de estar presente.

En esta reunión diaria el equipo de desarrollo hace las siguientes **tres preguntas**:

- ✓ ¿Qué hice ayer?
- ✓ ¿Qué voy a hacer hoy?
- ✓ ¿Tengo algún impedimento que necesito que me solucionen?

Una ***daily scrum*** es el lugar más oportuno para poder **inspeccionar el trabajo** y poder adaptarse en caso de que haya cambio de tareas dentro de un *sprint*. (Ángel, 2022, https://bit.ly/3r8WxlT)

Sprint review

La *review* del valor que vamos a entregar al cliente se hace en esta reunión, al final de cada *sprint*. Su duración es de 4 horas para *sprints* de un mes y es la **única reunión de *scrum* a la que puede asistir el cliente**.

En esta reunión, **el *product owner* presenta lo desarrollado al cliente** y el equipo de desarrollo muestra su funcionamiento. El cliente valida los cambios realizados y además brinda *feedback* sobre nuevas tareas que el *product owner* tendrá que agregar al *product backlog*. (Ángel, 2022, https://bit.ly/3r8WxlT)

Sprint retrospectivo

La retrospectiva es el último evento de *scrum*. Tiene una duración de 3 horas para *sprints* de un mes, y es la reunión del equipo en la que se hace una **evaluación de cómo se ha implementado el *framework scrum*** en el último *sprint*.

Es una gran **oportunidad para el equipo *scrum*** de inspeccionarse a sí mismo, proponiendo mejoras para el siguiente *sprint*. El resultado es una lista de mejoras que debe aplicar el siguiente día, ya que, al finalizar la retrospectiva, inmediatamente comienza un nuevo *sprint*, que incluye los mismos eventos mencionados anteriormente. (Ángel, 2022, https://bit.ly/3r8WxlT)

Grooming

El ***grooming* en *scrum*** es el tiempo de revisión que se dedica dentro de los equipos durante los *sprints* para refinar los requerimientos. Es una reunión que **no debe ocupar más del 10 % del tiempo disponible del *sprint***. Las tareas de usuario se revisan con un par de *sprints* de ventaja para que cuando llegue el equipo ya esté todo listo para el desarrollo de la tarea. (Ángel, 2022, https://bit.ly/3r8WxlT)

Artefactos del scrum

"Los **eventos y artefactos de *scrum*** están definidos para maximizar la transparencia dentro del equipo; es decir, que todos tengan una misma visión de lo que está ocurriendo en el proyecto" (Ángel, 2022, https://bit.ly/3r8WxlT).

A continuación, veremos los tres artefactos *scrum* claves: *product backlog*, *sprint backlog* e incremento. (Ángel, 2022, https://bit.ly/3r8WxlT)

Product backlog

Básicamente, el *product backlog* es el **listado de tareas que engloba todo un proyecto**. Cualquier cosa que debamos hacer debe estar en el *product backlog* y con un tiempo estimado por el equipo de desarrollo.

La responsabilidad exclusiva de ordenar el *product backlog* es del **product owner**, que se encuentra en constante comunicación con el cliente para asegurarse de que las prioridades están bien establecidas.

La ordenación también es 100 % responsabilidad del *product owner*, por lo que las tareas que están más arriba deben de ser las de **mayor prioridad.** El equipo de desarrollo elige tareas del *product backlog* en el *sprint planning* para generar tanto el *sprint backlog* como el *sprint goal*. (Ángel, 2022, https://bit.ly/3r8WxlT)

Sprint backlog

Es el grupo de tareas del *product backlog* que el equipo de desarrollo elige en el *sprint planning* junto con el plan para poder desarrollarlas. Debe ser conocido por todo el equipo, para asegurarse de que el foco esté en este grupo de tareas.

El *sprint planning* no cambia durante el *sprint*, solo se permite cambiar el plan para poder desarrollarlas. (Ángel, 2022, https://bit.ly/3r8WxlT)

Incremento del producto

El *product increment* o incremento de *scrum* es el resultado de **sumar todos los elementos del *product backlog* completados** durante el *sprint* actual, añadiendo también el valor de todos los *sprints* anteriores.

La suma de los incrementos se presenta en cada *sprint review*, y se pueden crear varios incrementos de *scrum* dentro de un mismo *sprint*. (Ángel, 2022, https://bit.ly/3r8WxlT)

Ventajas y desventajas

Ahora que sabemos **cómo funciona *scrum***, hablemos de sus ventajas y desventajas.

Ventajas de la metodología *scrum*

- *Scrum* es muy fácil de aprender: los roles, hitos y artefactos son claros y tienen un objetivo, por lo que es un método muy relacionado con nuestra manera diaria de trabajar.
- El cliente puede comenzar a usar el producto rápidamente.
- Se agiliza el proceso, ya que la entrega de valor es muy frecuente.
- Menor probabilidad de sorpresas o imprevistos, porque el cliente está viendo frecuentemente el proyecto. (Ángel, 2022, https://bit.ly/3r8WxlT)

Desventajas de la metodología *scrum*

- Aunque *scrum* sea fácil de aprender, es muy difícil implementarlo. Esto supone una predisposición y un cambio de cultura de la organización que debe ir desde los altos mandos hasta los clientes.
- La necesidad de tener equipos multidisciplinares puede ser un problema, ya que es difícil encontrar personas que sean capaces de hacer todo el trabajo de un equipo.
- El equipo puede tender a realizar el camino más corto para conseguir el objetivo de un *sprint*, el cual no siempre ofrece resultados de calidad. (Ángel, 2022, https://bit.ly/3r8WxlT)

En resumen:

> **El marco de trabajo *scrum* es especialmente útil cuando se trabaja en entornos de alta incertidumbre**, en los que la probabilidad de que haya cambios durante el proyecto es alta. Si los requisitos no están claros, si el cliente tiene prisa en sacar un producto al mercado o necesita un MVP, *scrum* es el *framework* perfecto. Este marco nos permite entregar un proyecto en varias partes funcionales e independientes entre sí, a un ritmo rápido y con capacidad de corregir errores en el momento. (Ángel, 2022, https://bit.ly/3r8WxlT)

Agile vs. *scrum*: ¿cuál es la diferencia?

No es fácil comparar directamente *agile* y *scrum*, ya que *agile* es una metodología que se puede interpretar e implementar de varias maneras según las necesidades del producto.

Con sus eventos y artefactos, *scrum* proporciona una estructura para que los equipos la sigan con objetivos y procesos específicos. Dos equipos ágiles diferentes en dos organizaciones diferentes pueden hacer cosas completamente diferentes entre sí. Mientras que dos equipos *scrum* opuestos seguirán realizando *sprints* y retrospectivas, etc.

Los dos métodos también difieren en algunas otras formas:

- ✓ La jerarquía es clave para crear productos ágiles, mientras que los equipos de *scrum* son en gran medida autónomos.
- ✓ Las personas en equipos ágiles colaboran constantemente con equipos multifuncionales, mientras que la mayor parte de la colaboración de *scrum* se lleva a cabo durante las reuniones de quince minutos.
- ✓ *Scrum* se adapta mejor a proyectos en los que los requisitos cambian con frecuencia, ya que los equipos pueden adaptar sus tareas diarias para satisfacer las nuevas necesidades del producto.
- ✓ *Scrum* se enfoca más en brindar valor comercial a lo largo de todo el proceso, no solo al final. (Merryweather, 2020, https://bit.ly/3CcG6LC)

Entonces, ¿*scrum* es ágil?

Sí y no. Scrum es una forma de implementar la metodología ágil en el desarrollo de *software*. Pero todo ágil no es *scrum*.

Un equipo ágil podría implementar algunas de las características de *scrum*, como una reunión diaria, pero sin cada pieza del rompecabezas, simplemente, no se trata de *scrum*. (Merryweather, 2020, https://bit.ly/3CcG6LC)

Unidad 4.2 *Growth marketing*

El *growth marketing* es:

La articulación de diferentes técnicas de *marketing* digital con análisis de datos y método científico. Se puede llegar a una definición entendiendo cómo se aborda el trabajo en esta disciplina. Su propósito es elaborar y experimentar pequeñas hipótesis para luego repetir y escalar. Una forma de hacer crecer un negocio o marca en este mundo digital.

En inglés, *growth* es crecimiento, pero podemos pensarlo también como aprendizaje. En esta nueva concepción del *marketing*, lo que importa es aprender, saber qué es lo que hay que hacer para ser más grandes.

Por eso, es importante la práctica de pensar hipótesis, testear, analizar datos, iterar, interpretar y aprender. Si la idea inicial tuvo éxito, se repite y se multiplica. Así, crece el negocio, se aprende y se aplican los aprendizajes.

El impacto del *growth marketing* alcanza todas las etapas del *funnel* de conversión, que es el camino que recorre un cliente hasta llegar a adquirir un bien o servicio. Las fases de la conversión son descubrimiento, consideración, conversión y fidelización.

Un ejemplo de este camino puede ser cuando una persona quiere comprar un auto y, mientras busca «mejores autos familiares», descubre un modelo que no había visto. Luego, comienza a considerar si es necesario comprar o no, analizando decenas de variables.

Cuando se decide, convierte, compra el auto. Y, si le resultó satisfactoria la experiencia, recomienda. Una concesionaria puede crear

contenido para esta persona, publicando un artículo sobre «cómo comprar un auto familiar» en su blog y así contactar a un posible cliente.

El *growth marketing* se puede aplicar experimentando en la etapa de descubrimiento, trabajando canales de adquisición, o creando contenido para la etapa de consideración o búsqueda de información. Incluso, al final, en la etapa de fidelización. (Rojas, 2022, https://bit.ly/3ShAKo3)

¿Qué diferencia hay entre *growth marketing* y *growth hacking*?

El *growth hacking* es un término acuñado por Sean Ellis, quien trabajó en diferentes empresas, como Eventbrite, Dropbox, empresas que valen millones de dólares. Es conocido como «el primer *growth hacker*».

Básicamente, este concepto hace referencia a una forma de trabajo que busca tácticas de crecimiento rápido, haciendo uso y análisis de datos combinados con canales de *marketing*.

Con respecto al *growth marketing*, es similar, está dentro, pero no es lo mismo. En el *growth hacking*, los esfuerzos apuntan a tener un resultado a corto plazo. Son dos conceptos que están íntimamente relacionados, pero no hablan de lo mismo.

El *growth marketing* está orientado a mirar todo el proceso de la venta. El *growth hacker* solo busca experimentar, probar y vender más.

Esto quiere decir que lo que busca un *growth hacker* es pensar la forma de analizar un problema y solucionarlo. *Hacker* es una palabra que puede estar relacionada con algo «malo», pero en realidad hace referencia a conseguir un «truco» o una «vueltita» para usar un recurso inteligentemente. (Rojas, 2022, https://bit.ly/3ShAKo3)

¿Cómo se puede utilizar el *growth marketing* en una empresa?

Dependiendo del tamaño de la empresa, el *growth marketing* se puede aplicar de distintas formas. Pero siempre partiendo desde el consumidor, al usuario o *buyer* persona, quien va a usar nuestra marca.

Hay que analizar primero el producto que tenemos, qué características tiene y cómo lo presentamos al consumidor. Luego, pensar en el *buyer* persona: qué características tiene, cómo vive, qué está buscando. Por último, ir a buscar a ese perfil en redes sociales y en otros canales *online*, probando la forma de comunicar y viralizar.

Algunas estrategias son muy similares a las del *growth hacking* y pueden ser:

- Crear perfiles en redes sociales, para estar en contacto con el público objetivo.
- Seguir a las empresas y marcas que siguen para aprender a comunicar mejor.

- Realizar *test* A/B en pequeños grupos de páginas o con un universo acotado de clientes, para después repetir y escalar el éxito, en caso de que funcione.

- Gamificación u otorgar puntos por uso de *apps* o plataformas. Puntos por ventas, por compartir con amigos, etc.

- Ofrecer servicios gratuitos. Si se ofrece un servicio, dar una parte gratis. O un periodo de tiempo, para dar a conocer el servicio.

- Lanzar pruebas cerradas o invitaciones exclusivas. Consiste en ofrecer un servicio a pocas personas para que lo prueben y generar ese sentido de pertenencia y exclusividad. (Rojas, 2022, https://bit.ly/3ShAKo3)

Implementar una estrategia de *growth marketing*

El punto de partida es la persona que va a usar el producto, quien va a consumir la marca. Se debe investigar en dónde está, qué redes sociales usa, cómo es, qué problemas tiene. Luego, usar esa información para realizar el contacto.

Inmediatamente después hay que pensar qué canales usar para llegar a esa persona. Pueden ser redes sociales o un blog. También se puede trabajar con SEO para estar en resultados de búsqueda de Google. Allí donde busca la audiencia que todavía no nos conoce.

La idea es crear los contenidos ideales para llegar a ese usuario. Analizar los resultados de la comunicación para saber si está funcionando o no, si existen oportunidades de mejora.

Hasta acá se puede decir que es una estrategia de comunicación, no hay nada nuevo. Pero en *growth marketing* tenemos la posibilidad de usar los datos que obtenemos en cada contacto con el cliente. Esto nos permite probar diferentes acciones para cada parte de la estrategia. Si las pruebas funcionan, se pueden repetir, y así seguir creciendo.

Por ejemplo, una marca de electrodomésticos puede testear comunicaciones sobre recetas de cocina en su blog. Si funcionan y traen tráfico, significa más clientes. También puede crear un libro descargable o videos y potenciar su adquisición orgánica de clientes. (Rojas, 2022, https://bit.ly/3ShAKo3)

Adquisición de usuarios

La estrategia de captación de usuarios de un servicio debe regirse por los principios de *push* y *pull*, en función del enfoque de nuestro negocio. Veamos qué significan estos principios.

El término *push* se refiere a la publicidad obstructiva y se ofrece al cliente mientras este realiza una actividad de su interés, por ejemplo, hacen los anuncios en televisión o en YouTube. Las estrategias *pull*, por su parte, comunican un mensaje que será encontrado por su público objetivo, que consideramos como un grupo de usuarios interesados en esa materia y en búsqueda activa al respecto. Como nos podemos imaginar, en la

publicidad *push* de internet existen mecanismos de segmentación para que, aun siendo obstructiva, sea lo menos intrusiva posible y más afín al usuario. (Gómez, 2016, https://bit.ly/3UK0v1D)

La adquisición de usuarios, entonces:

Consiste en la estrategia a implementar para conseguir atraer visitantes a nuestro servicio. Y aquí viene uno de los mantras del *growth hacking*: **la calidad siempre es mejor que la cantidad**.

Normalmente, tenemos una tendencia natural a pensar que necesitamos siempre, mucho y de todo, pero no es cierto. Internet es tan apabullantemente grande que, con una buena estrategia de adquisición de usuarios, siempre se va a poder mantener un flujo de usuarios suficiente sin detrimento de su calidad.

Bien, y ¿qué son exactamente usuarios de calidad? Usuarios con un interés previo y real en el producto o servicio ofrecido. Cualquier otra cosa que no sea exactamente así juega en contra. Por ejemplo, llevar hacia la web miles de usuarios de forma «casi» engañosa hará descender drásticamente las medias de *ratio* de rebote (es decir, visitantes que abandonan la página nada más acceder a ella, sin realizar ninguna otra acción destacable) y el tiempo medio por visita, lo cual es muy negativo para el posicionamiento en motores de búsqueda. (Gómez, 2016, https://bit.ly/3UK0v1D)

Embudo de ventas

El embudo de ventas es un modelo estratégico utilizado por las áreas de *marketing* y ventas de las empresas para mapear los momentos clave en el flujo de captación de un cliente. También es conocido como *funnel* de ventas.

Se trata de una de las herramientas más importantes para un negocio, ya que es similar a un mapa del tesoro que hay que seguir para alcanzar los objetivos.

Un embudo o *funnel* de ventas es un concepto del *marketing* digital que representa las etapas que, idealmente, atraviesa un potencial cliente para completar su compra.

[…]

Es decir, es un mapa de los momentos clave por los que pasa un consumidor y, en general, se compone de tres partes:

- ToFu (*top of the funnel* o parte superior del embudo de ventas)
- MoFu (*middle of the funnel* o parte media del embudo de ventas)
- BoFu (*bottom of the funnel* o parte inferior del embudo de ventas)

El embudo de ventas es una manera gráfica de ilustrar cómo un cliente interactúa con una marca, comenzando con un gran número de interesados hasta el grupo de clientes que en realidad compra el producto o servicio que se ofrece. (León, 2022, https://bit.ly/3RkUnKt)

Figura 6: Embudo de ventas

Fuente: León, 2022, https://bit.ly/3RkUnKt

El objetivo de la implementación del embudo de ventas es ofrecer un entendimiento de los procesos que atraviesa el consumidor en su paso a la adquisición del bien o servicio que se ofrece. Por esta misma razón, la ejecución de los pasos puede diferir un poco de empresa a empresa. No se trata de una plantilla que podemos copiar y pegar.

Los embudos de venta son adaptables al público meta, al servicio ofrecido y según los objetivos de venta de la empresa. (Rodríguez, 2021, https://bit.ly/3rfPg3H)

Sin embargo, en todos los casos, el esquema del embudo de ventas tiene cinco fases en común aplicable a cualquier empresa.

1. **Atracción**. Es la etapa en la que el consumidor nota una necesidad y busca satisfacerla. Si un negocio puede cubrirla, la estrategia se concentra en demostrar cómo satisfacer esta necesidad, incluso si los consumidores no están conscientes de que la tienen.
2. **Interacción**. En esta etapa, el consumidor se encuentra en la evaluación sobre cómo el producto ofrecido satisface sus necesidades. Incluye la evaluación de las ofertas, tanto nuestra como de la competencia.

   Por esta razón, será importante establecer un acercamiento sólido. Interacción temprana a través de un *ventabot*, encuestas, videos y cualquier tipo de atención de parte de la empresa demostrará que tenemos lo que el cliente necesita en la medida justa.
3. **Conversión**. Al haber mantenido una interacción satisfactoria, el cliente muestra una inclinación lógica o emocional sobre la solución que desea adquirir. Para esto será necesario el ofrecimiento de contenido interesante con el ganaremos su confianza.
4. **Compra**. Para muchos especialistas en *marketing*, este es el punto final del embudo. Sin embargo, una venta no garantiza un cliente. La compra es la demostración tangible de que las estrategias han funcionado y hemos ganado la confianza del consumidor.
5. **Fidelización**. Es la garantía de haber generado en el cliente un proceso emocional y lógico que dirige lo dirige por una segunda

compra. Luego de esto sí podremos decir que ¡tenemos un cliente! (Rodríguez, 2021, https://bit.ly/3rfPg3H)

En resumen:

El embudo de ventas sirve para predecir y seguir el camino que recorrerá el potencial cliente, desde el primer contacto con la marca hasta realizar la compra en la tienda *online*.

A través del embudo de ventas es posible crear acciones específicas para cada etapa del recorrido, resolviendo los miedos y objeciones que pueda tener el futuro comprador, otorgándole la información que necesita y así llevarlo de la mano hasta que concrete su compra. (León, 2022, https://bit.ly/3RkUnKt)

Email marketing

El *email marketing* define el envío de correos electrónicos a una base de datos o a contactos que son clientes potenciales. Si se mandan los correos a los clientes actuales, entonces el *emailing* es con objetivo de fidelización.

Actualmente, gracias a la automatización y la IA, se pueden personalizar de forma increíble estos *emails* y así mejorar los resultados.

Se ha predicho su decaimiento desde hace mucho, pero esta técnica veterana de *marketing* sigue tan vigente como años atrás. El *email*

marketing de captación o de fidelización continúa siendo una de las acciones más rentables y efectivas para los anunciantes en términos de retorno.

Se llega fácilmente y de forma eficaz a los clientes potenciales. Ya no solo a las bandejas de entrada, sino directamente a la mano del cliente gracias a los dispositivos móviles. Muchos usuarios tienen configurado el correo personal y el profesional en su *smartphone* o *tablet*. Cerca de la mitad de los *emails* son abiertos desde un móvil y esto nos obliga a que sus diseños sean *responsive*.

Pensar hoy en día en *email marketing* es pensar en formato *mobile*, si no lo hacemos así, nuestro mensaje estará muerto incluso antes de llegar a manos del consumidor. De aquí que se hable muchas veces que la evolución del *email marketing* es el *mobile email marketing*. (Cyberclick, s.f., https://bit.ly/3SywL62)

A/B *testing*

El A/B *test* consiste en desarrollar y lanzar dos versiones de un mismo elemento y medir cuál funciona mejor. Es una prueba que nos sirve para optimizar una estrategia de *email marketing* o mejorar la efectividad de una *landing page*.

Este método consiste en desarrollar dos versiones de un mismo elemento que vamos a lanzar al mercado y luego utilizar las métricas de cada variación para evaluar cuál funciona mejor.

Al contrario de lo que pueda parecer, hacer muchas variaciones no tiene por qué producir impactos negativos; son cambios incrementales que mantendrán a tus usuarios conectados y más cerca del último eslabón del ciclo de compra.

Para utilizar bien el A/B *test*, debemos centrar nuestra atención en esos elementos que influyen en el *ratio* de apertura de un email y en los clics que el usuario hace en una *landing page*. Estos son algunos de esos elementos que podemos testear en un A/B *test*:

- ✓ Las palabras, colores, tamaños.
- ✓ Los titulares y cuerpos de la descripción de los productos.
- ✓ La extensión de un formulario y tipos de campos.
- ✓ El *layout* o estructura visual de la página web.
- ✓ El modo de presentación del precio de los productos y ofertas promocionales.
- ✓ Las imágenes (ubicación, propósito, contenido y cantidad) de las *landings* y páginas de tu producto.
- ✓ Cantidad de texto en una página web o en un blog *post*. (40 de Fiebre, s.f., https://bit.ly/2DJudP4)

En otras palabras:

El A/B *testing* ayuda a evitar la necesidad de formular conjeturas, a mejorar la optimización del sitio web y a que los expertos en CRO tomen decisiones respaldadas por datos. En cuanto a la nomenclatura, la «A» se refiere a la

versión original o «de control» de la página web, mientras que la «B» representa la nueva versión o la «variación».

La alternativa que mejores métricas genera para el negocio se denomina «versión ganadora», ya que implementar los cambios incluidos en ella a las páginas o elementos testeados puede ayudar a optimizar el sitio web e incrementar el ROI de tu empresa.

Las métricas de conversión son diferentes para cada empresa. En el caso del *ecommerce*, por ejemplo, podría tratarse de las ventas de los productos. Sin embargo, en un negocio B2B podría estar determinado por la generación de *leads* cualificados.

El A/B *testing* forma parte del proceso de optimización de la tasa de conversión (CRO) y ayuda a recopilar *insights* cualitativos y cuantitativos. Los resultados obtenidos se pueden utilizar para comprender el comportamiento de los usuarios, la tasa de *engagement*, los puntos de fricción e incluso el nivel de satisfacción con distintas características del sitio web, como nuevas funcionalidades, secciones remodeladas, etc. Por tanto, el A/B *testing* de un sitio web permite evitar la pérdida de una gran cantidad de potenciales beneficios para la empresa.

Hoy en día, una de las principales preocupaciones de los negocios B2B es la cantidad *leads* no cualificados que generan cada mes, mientras que el punto débil de las empresas de *ecommerce* tiene más que ver con la elevada tasa de abandono del carrito. Las editoriales y las empresas de medios, por su parte, se tienen que enfrentar a unos niveles de *engagement*

reducidos. Estas métricas de conversión principales se ven perjudicadas, en todos los casos, por problemas como fugas en el *funnel* de conversión, abandonos en la página de pago, etc. (VWO, s.f., https://bit.ly/3CeuGXx)

Veamos, a continuación, qué ventajas aporta el A/B *testing*.

Figura 7: A/B *testing*

Fuente: VWO, s.f., https://bit.ly/3CeuGXx

Resolución de puntos de fricción

Los visitantes llegan a un sitio web con un propósito específico, ya sea obtener más información sobre productos o servicios, comprar un artículo concreto, leer o profundizar sobre un tema en particular o, simplemente, echar un vistazo. Sea cual sea el objetivo, es posible que se enfrenten a una serie de puntos de fricción comunes, como un *copy* redactado de forma confusa, un CTA de compra o una solicitud de demo ubicados en un lugar difícil de localizar, etc.

El hecho de que los usuarios no sean capaces conseguir el objetivo con el que llegaron implica que su experiencia ha sido mala, lo cual incrementa el nivel de fricción y acaba por repercutir en las tasas de

conversión. Por eso, es importante utilizar los datos recopilados mediante herramientas de análisis del comportamiento de los visitantes, como los *heatmaps*, Google Analytics y las encuestas en el sitio web para solucionar las áreas en las que los visitantes tienden a encontrar más problemas. Este procedimiento es aplicable a empresas de todo tipo: e*commerce*, turísticas, SaaS, educativas, de medios de comunicación y editoriales. (VWO, s.f., https://bit.ly/3CeuGXx)

Mejora el ROI a partir del tráfico existente

Tal y como se ha venido comprobando con el paso del tiempo, conseguir tráfico de calidad es esencial para los sitios web. El A/B *testing* permite sacar el máximo partido al flujo de visitantes ya existente e incrementar las conversiones sin necesidad de realizar inversiones adicionales ni de atraer a más usuarios. El A/B *testing* puede ayudar a obtener un ROI elevado, ya que, en ocasiones, incluso el más mínimo cambio acaba por resultar en un incremento significativo de la tasa general de conversión de una empresa. (VWO, s.f., https://bit.ly/3CeuGXx)

Reducción de tasas de rebote

Una de las métricas más importantes a tener en cuenta para conocer el desempeño de un sitio web es la tasa de rebote. Una tasa de rebote elevada puede deberse a múltiples factores, como una oferta excesiva de opciones a elegir, incompatibilidad con las expectativas, una navegación confusa, el uso de un lenguaje demasiado técnico, etc.

Puesto que cada sitio web tiene su propio cometido único y está dirigido a un público objetivo concreto, no existe una solución universal para conseguir una mejora en este aspecto. No obstante, los *test* A/B pueden resultar de gran ayuda, ya que permiten testear múltiples variaciones de un elemento para dar con la mejor versión posible. Esto no solo ayuda a identificar puntos de fricción y conflicto, sino que permite optimizar la experiencia general de los usuarios para que pasen más tiempo en el sitio web e incluso acaben por convertirse en clientes habituales. (VWO, s.f., https://bit.ly/3CeuGXx)

Toma de decisiones con riesgo reducido

El A/B *testing* permite introducir pequeños cambios de forma gradual en un sitio web para no tener que rediseñarlo por completo, lo cual reduce el riesgo de poner en peligro la tasa de conversión ya existente.

Gracias al A/B *testing* podremos aprovechar al máximo los recursos, obtener grandes resultados con modificaciones mínimas e incrementar el ROI. Un ejemplo de ello serían los cambios en las descripciones de productos. Podemos ejecutar *test* A/B siempre que tengamos pensado eliminar o actualizar la descripción de alguno de los productos para saber cómo reaccionarán los visitantes a dicha modificación y qué versión reportará más beneficios a la empresa.

Otro ejemplo de variación de bajo riesgo sería la introducción de una nueva funcionalidad. Recurrir al A/B *testing* antes de publicar una nueva

funcionalidad puede ayudar a saber qué acogida le darán los visitantes del sitio web.

El hecho de implementar cambios sin testearlos previamente hace que resulte imposible prever si funcionarán o no a corto o a largo plazo, mientras que los *test* aportan un mayor grado de seguridad. (VWO, s.f., https://bit.ly/3CeuGXx)

Obtención de mejoras estadísticas

El A/B *testing* ofrece resultados totalmente basados en datos, lo cual elimina la necesidad de realizar conjeturas o fiarse del propio instinto a la hora de tomar decisiones. Por tanto, resulta sencillo determinar qué versión es la ganadora y cuál la perdedora a través de métricas, como el tiempo que pasan los visitantes en la página, el número de solicitudes de demos, la tasa de abandono del carrito, el CTR, etc. (VWO, s.f., https://bit.ly/3CeuGXx)

Rediseño web para aumentar ventas futuras

La remodelación de un sitio web puede incluir desde pequeños cambios, como el color o el texto de un CTA, hasta otros más importantes, como la modificación total de ciertas páginas web o incluso de todo el sitio. Cuando se realiza el A/B *testing*, las decisiones sobre qué versión implementar deberían basarse por completo en los datos obtenidos. Es más, este procedimiento de experimentación resulta útil incluso cuando ya se ha lanzado el nuevo diseño, dado que permite probar distintas versiones de ciertos elementos para garantizar que se está ofreciendo la alternativa que genera más *engagement*. (VWO, s.f., https://bit.ly/3CeuGXx)

Unidad 4.3 Experimentar e iterar

> La experimentación, método común de las ciencias experimentales y las tecnologías, consiste en el estudio de un fenómeno, reproducido generalmente en un laboratorio, en las condiciones particulares de estudio que interesan, eliminando o introduciendo aquellas variables que pueden influir en él. (Pérez Prado, s.f., https://bit.ly/3DZ60DH)

Experimentar es una forma de aprendizaje; es un método para obtener nuevos conocimientos.

Figura 8: *Framework* de experimentación en 4 pasos para equipos de productos

Fuente: elaboración propia

Distintas fases, mismo *framework* de trabajo

Necesidad del cliente y *discovery*

"En esta fase, la hipótesis que necesitamos validar es la necesidad del cliente/usuario, priorizar las necesidades y entender cuáles son las oportunidades de mercado. En esta etapa, el MVP tendrá funciones limitadas y de baja fidelidad (*mock-ups*, prototipos simples)" (Pérez Prado, s.f., https://bit.ly/3DZ60DH).

Solución del problema

En esta fase, la hipótesis a validar es si hay encaje entre el producto y solución y su viabilidad técnico-funcional. También es el momento de validar el grado de satisfacción versus los sustitutos.

En este estadio, el MVP comenzará a tener una fidelidad media, con interacción simulada (*landing page*, *mock-up* con interacción). (Pérez Prado, s.f., https://bit.ly/3DZ60DH)

Product- market fit

Esta fase es una extensión de la anterior, donde se continúa con la validación del encaje producto–solución y su viabilidad. La diferencia es que el MVP comienza a pasar de una fidelidad media a alta con una interacción real. (Pérez Prado, s.f., https://bit.ly/3DZ60DH)

Product- channel fit y optimización de conversión en *funnel*

En esta última fase, debemos validar el encaje entre el producto y el canal y los *gaps* en el *funnel* de conversión: adquisición, activación, recurrencia, pago, *referral*.

Por el lado del producto, ya comienza a dejar de ser un MVP y se comienza a transformar en un producto funcionando. (Pérez Prado, s.f., https://bit.ly/3DZ60DH)

***Framework* para una cultura de experimentación**

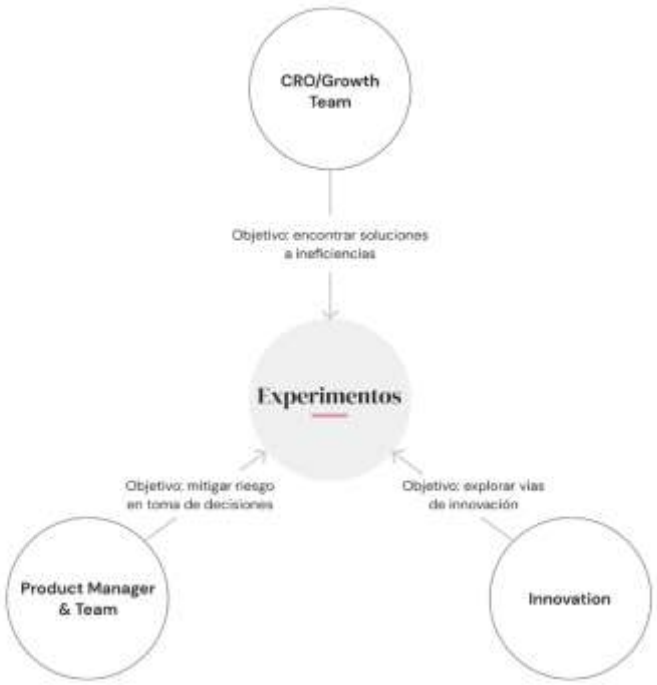
Experimentar es, ni más ni menos, que un método de investigación que busca demostrar causalidad, un método de validación de hipótesis, parte fundamental

del método científico. Antes de crear un *framework* para desarrollar la cultura de la experimentación, debemos saber que podemos investigar con diferentes objetivos, principalmente tres:

1. Encontrar mejores soluciones a una propuesta ineficiente (propio de equipos de CRO o *growth*).
2. Explorar propuestas de innovación (propio de equipos de I+D).
3. Mitigar el riesgo de afectar negativamente al rendimiento de la experiencia con nuevas *features* o diseños (propio de equipos de producto).

Desde CRO o *growth*, habitualmente, experimentamos para encontrar palancas de optimización y maximización del rendimiento de una experiencia digital. Mejorando el rendimiento de la experiencia, impulsaremos el crecimiento de un negocio. Nosotros lo llamamos «hacer *growth*». Previamente a la creación de un *framework* de experimentación, tenemos que saber cómo incorporar la experimentación. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK)

Figura 9: Experimentos



Fuente: Lipúzcoa, s.f., https://bit.ly/3SkfkGK

¿Cómo incorporar la experimentación?

Podemos distinguir —mínimo— tres enfoques desde los que enfrentarse a la experimentación aplicada al diseño, creación, mantenimiento y escalado de un *funnel* o experiencia digital:

1. **Experimentación como *mindset*:** experimentar está interiorizado por el equipo. Permea de arriba a abajo en el organigrama y viceversa.

2. **Experimentación táctica:** ciertos expertos o equipos dominan la técnica de la experimentación, que no tiene por qué estar estructurada. Se experimenta con objetivos dispares. Es un método de investigación utilizado por equipos diferentes en diferentes fases de los proyectos.

3. **Experimentación estratégica**: en busca de la maximización u optimización de una métrica de negocio, se crea un *roadmap* priorizado de experimentos que un equipo se encarga de ejecutar.

Para escalar una cultura de la experimentación eficiente, tienen que convivir los tres enfoques, aunque cada uno de ellos solo sobrevive si coexiste con, al menos, otro.

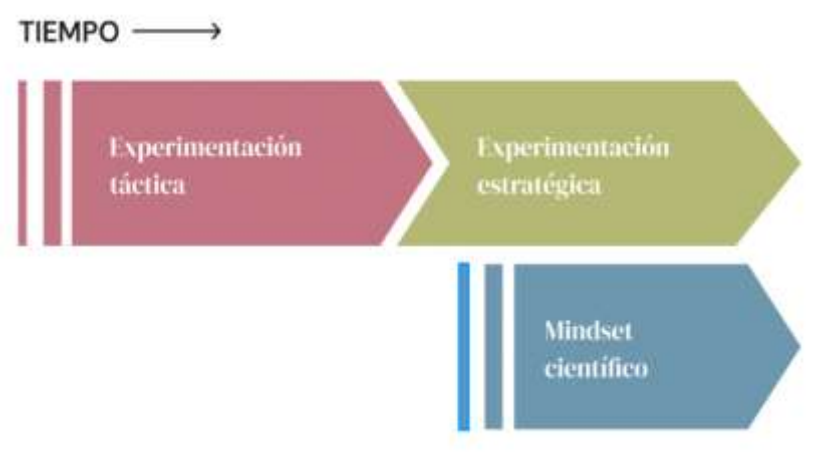
Lo ideal es que convivan los tres, ahí radica el máximo potencial de la experimentación. Pero dependerá de las necesidades del negocio, de las capacidades del equipo y de la madurez de ambos (equipo y negocio). No hay un camino válido para todo el mundo. Algunos empiezan por una estrategia para alcanzar la excelencia táctica y otros empiezan por la instalación de un *mindset* para elaborar una estrategia. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK)

Figura 10: Máximo potencial de la experimentación

Fuente: Lipúzcoa, s.f., https://bit.ly/3SkfkGK

"Una empresa podría empezar con un enfoque muy táctico mientras traza una estrategia robusta e inculca el método científico como marco de trabajo" (Lipúzcoa, s.f., https://bit.ly/3SkfkGK).

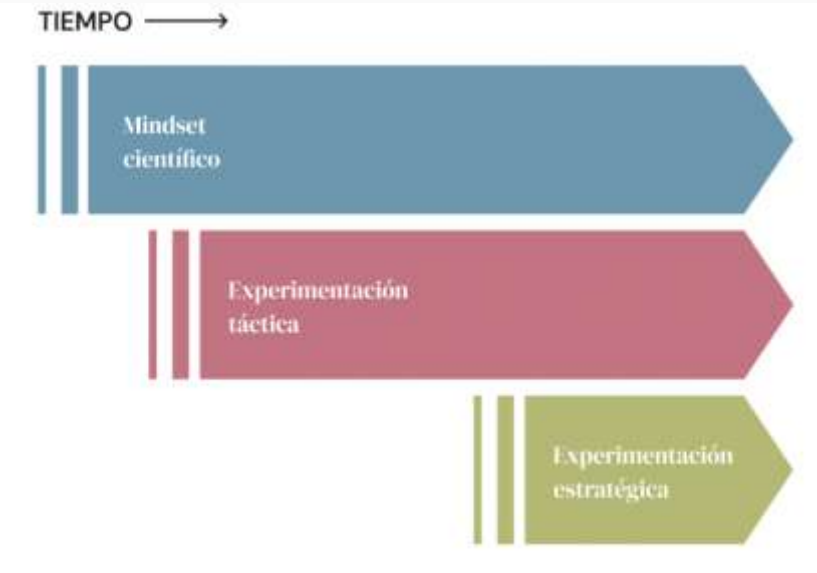
Figura 11: Panorama de la experimentación en las empresas



Fuente: Lipúzcoa, s.f., https://bit.ly/3SkfkGK

"Otra empresa podría tener dentro del equipo perfiles muy *data-informed* y muy cercanos a una filosofía de prueba y error, empezar a experimentar de forma aislada y acabar formando uno o varios equipos con objetivos y estrategias definidos" (Lipúzcoa, s.f., https://bit.ly/3SkfkGK).

Figura 12: Panorama de la experimentación en las empresas



Fuente: Lipúzcoa, s.f., https://bit.ly/3SkfkGK

Factores claves del *framework* de experimentación

En un flujo de experimentación óptimo, ya sea estratégico o puramente táctico, hay cuatro variables que hay que mantener en equilibrio en la maquinaria de optimización y crecimiento de un producto digital:

1. Velocidad de experimentación
2. Confianza en el dato
3. Democratización
4. Solidez de las hipótesis

Hay un quinto factor variable que ayudará en los otros cuatro: **automatización**, que es intrínseco a todos ellos. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK).

Velocidad de experimentación

Un experimento es un aprendizaje. **Cien experimentos, cien aprendizajes.** Con un *ratio* promedio de *test* positivos del 20 % y una mejora observada de 0,5 %, en promedio, en ***revenue***: cada 10 experimentos, accionaremos un potencial incremento del 1 %. Cada 100, un 10 % de crecimiento.

La velocidad en la experimentación **estratégica** es un *plus*.

Cada nuevo *feature*, rediseño o ajuste del equipo de producto supone una toma de decisión con su riesgo asociado. Un *test* antes de un *rollout* permite mitigar el riesgo de una potencial pérdida de rendimiento con esa nueva

feature. Es lo más "*lean*" que podemos hacer. Pero eso, requiere velocidad y mucha agilidad.

La velocidad en la experimentación **táctica** también es un *plus*. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK).

Confianza en el dato

No hay nada más efectivo para aniquilar a un equipo de analítica, *testing*… Que un flujo de datos poco confiable.

«Yo no me fiaría mucho de ese dato», «los datos de Analytics no corresponden con los del CRM», «Esta métrica nunca estuvo bien medida», se suele escuchar decir.

En este sentido, hay tres claves para asegurar un flujo que genere confianza:

1. La analítica de la experimentación no es independiente de la analítica de producto o negocio. Usa los mismos modelos y se apoya en las mismas métricas, medidas de la misma manera.
2. Se pueden triangular los datos. Se recogen, como mínimo, en dos plataformas de forma que siempre podamos comparar.
3. Existe en el equipo un conocimiento mínimo sobre estadística inferencial y contraste de hipótesis. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK).

Democratización

Para que exista una cultura hay que convertir la experimentación en un valor intrínseco a la forma de actuar y pensar de todos los equipos de la compañía.

Hay algunas claves para desbloquear este logro:

1. **Accesibilidad del dato**: cualquiera puede comprobar los datos de los experimentos, sus resultados y sus conclusiones. Además, cualquiera puede entender los datos.
2. **Educar, evangelizar, generar y mantener la curiosidad**.
3. **Cualquier equipo tiene acceso a la experimentación**: cualquier equipo puede ejecutar o, al menos, solicitar la ejecución de un test para validar una hipótesis.
4. **Dar valor a los beneficios de la experimentación** una y otra vez. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK)

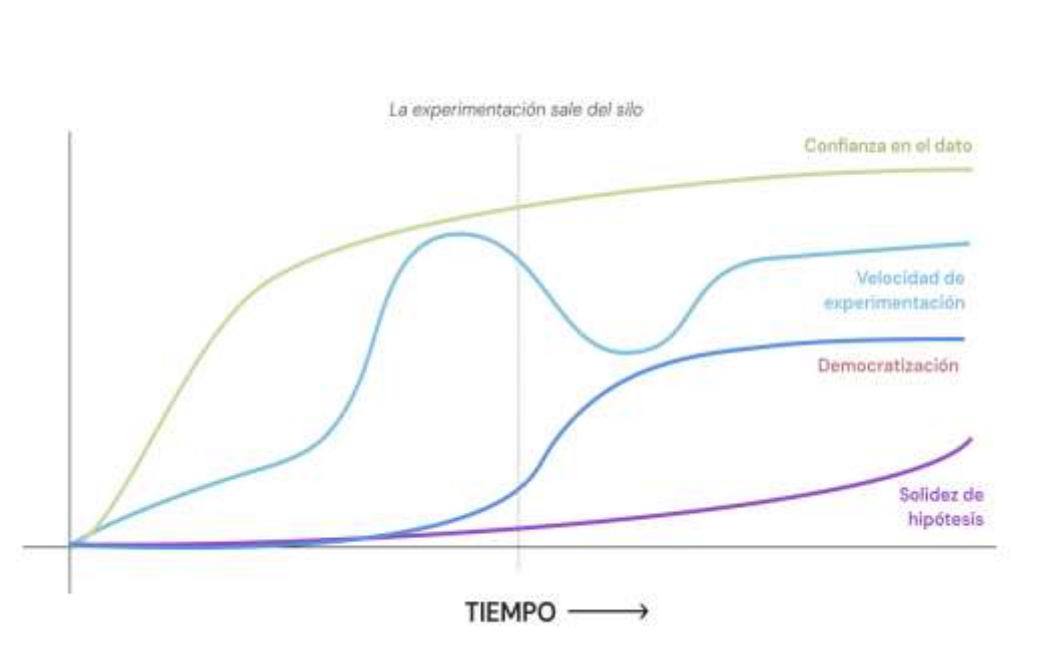
Solidez de la hipótesis

Una de las grandes ineficiencias de un programa de experimentación es la elaboración de hipótesis poco **argumentadas**, sin base sólida, sin KPI bien definidos. Esto dificulta el diseño del *test* y la interpretación de los resultados, porque si no sabemos exactamente qué queremos investigar, no sabemos qué ver en los datos.

Crear buenas hipótesis es algo que se puede entrenar y es parte fundamental de una cultura de la experimentación robusta y escalable. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK)

En el siguiente gráfico se ve cómo evolucionan habitualmente estos cuatro factores a medida que madura una cultura de experimentación dentro de un equipo.

Figura 13: Evolución de los factores de la cultura de la experimentación



Fuente: Lipúzcoa, s.f., https://bit.ly/3SkfkGK

Lo que habilita el crecimiento del resto de variables es la **confianza en el dato.** Sin esto, la cultura de la experimentación no escala.

Y aquí es donde entra en juego el *framework* de experimentación. Llega un momento en el que se desbloquean aspectos técnicos, se superan curvas de aprendizaje y la velocidad se incrementa rápidamente. Pasamos de un experimento al mes a cinco sin darnos cuenta, **los experimentos ya no solo son cosa de un equipo**. ¿Qué ocurrirá inevitablemente? Que a medida que la experimentación se democratiza, el

equipo de experimentación no tendrá los recursos necesarios para dar soporte al resto de equipos. Pero una vez superado este bache, podemos estabilizar el crecimiento de la cultura de experimentación.

De forma intrínseca, a medida que ganemos madurez, crecerá la solidez de nuestras hipótesis y, con ella, el *ratio* de resultados positivos. (Lipúzcoa, s.f., https://bit.ly/3SkfkGK)

Iterar

Iterar es un concepto que se ha vuelto clave para definir una metodología empresarial. Y que, desde hace algunos años, se ha incorporado al mundo de los negocios. Consiste en, precisamente, repetir una y otra vez un proceso, tomando en cuenta los resultados previos, con el objetivo de mejorarlo hasta llegar a donde se desea.

Su principal utilización en los negocios viene desde que Erick Ries la usó en *Lean Startup*, el libro que, desde hace unos años, se convirtió en una referencia para emprender.

En el caso de Ries, él aseguraba que su método era una nueva forma de mirar el desarrollo de productos innovadores, enfocado en la rápida iteración y la percepción del cliente.

Cada iteración de una *startup* es un intento de acelerar su motor para ver si avanza. Una vez que está funcionando, el proceso se repite, mejorando

cada vez más sus engranajes. Es decir, que empezar de nuevo una y otra vez se ha convertido en una metodología exitosa.

> Pero ¿por qué es importante que las empresas incorporen la iteración en su estrategia de negocios? Entre otras cosas, porque hacerlo es un camino a la innovación. (Ilab, 2020, https://bit.ly/3Chm9n0)

En este sentido, Sosa (2019) explica:

> El proceso de diseño iterativo, siempre tendrá como aliado a la investigación de usuarios. La información que obtengamos de nuestros clientes, nos ayudará a saber el camino qué queremos tomar, es decir, un montón de ideas que después nos permitan crear un prototipo. Y a su vez, este será probado con los usuarios para poder hacer las modificaciones necesarias.
>
> […] La iteración nos permite desarrollar rápidamente un prototipo del producto y luego ponerlo a prueba con usuarios reales. ¿Y eso cómo ayuda? Pues esto nos ayuda a determinar cómo se enfrentan al producto, si es fácil de usar, si es comprensible, si les agrada, etc.
>
> Ya que resulta más sencillo (y económico) crear un prototipo, es relativamente más rápido ponerlo al alcance de los usuarios para aprender sobre él. Además, resulta ideal para el presupuesto, aprender de «fracasos económicos» que aprender de «fracasos costosos».
>
> Diseñar de esta forma tiene varias ventajas, que se mencionan a continuación. Pero sobre todas ellas, iterar significa estar activo constantemente, estar al pendiente de la mejora de nuestro producto para

los usuarios; hasta que lleguemos al punto en que no se pueda aprender nada más. No se trata de una cuestión instantánea y mágica, donde se resuelven los problemas de un momento a otro.

Se trata de un paso a paso, de un camino en el que se va a perfeccionando el producto o servicio, un proceso donde se aprende de los usuarios, del producto mismo y hasta del equipo que ejecute. (Sosa, 2019, https://bit.ly/3LS0jJJ)

Iterativo e incremental

En un entorno tan cambiante como el actual, las organizaciones necesitan disponer de una visión más amplia de lo que sucede a su alrededor. Los cambios en tecnología evolucionan rápidamente y ello afecta directamente en el ecosistema de los mercados y su exposición en el medio.

Se requiere, por tanto, que la estrategia de negocio pueda adaptarse rápidamente sobre el entorno volátil en el que vivimos. Es volátil, hay incertidumbre y no es predictivo. Nos obliga a disponer de la máxima información más rápidamente, para anticiparnos y poder formular estrategias acordes al ritmo cardíaco del mercado. Un pronóstico constante bajo un proceso empírico.

Ello implica que, si queremos lanzar un producto al mercado en 12 meses, no parece tener mucho sentido irnos a un sistema de desarrollo tradicional, ya que el tiempo que transcurre desde la necesidad planteada hasta la entrega puede generar varios problemas:

- Alguno o parte de los requisitos no son correctos y desafortunadamente no nos damos cuenta hasta la fecha de la entrega, pasados los 12 meses.
- Puede que el negocio ya no esté interesado en los *features* implementados y tengamos los ya conocidos «*change requests*», lo cual incrementa la duración del proyecto y los costes totales; ya que, después de 12 meses, puede ocurrir que hay que tirar a la basura parte del desarrollo ya finalizado.

¿Y si los usuarios no tienen claro el objetivo final? Siendo un sistema de desarrollo tradicional orientado a documentación, quizás no sea la mejor opción para arrancar.

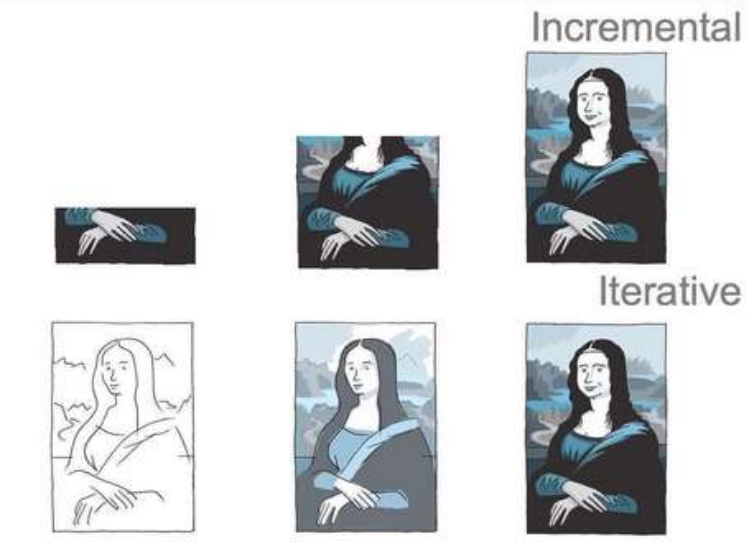
Una alternativa para mitigar los riesgos detectados es iterar por entregas, por ejemplo, 4 iteraciones de 3 meses. Es decir, trabajar en versiones completas en cada ciclo en lugar de trabajar las partes en su máximo detalle. Ello nos acerca al punto de obtener información sobre el trabajo realizado en menos tiempo y evaluar posibles cambios para la siguiente iteración. El objetivo es aquel definido a nivel global y nos permite evaluar el avance con respecto a la totalidad del proyecto.

Pero si dijimos que el entorno es cambiante, no parece que 3 meses pueda ser lo suficientemente adaptativo para nosotros.

Para ello, vamos a aumentar la frecuencia de las iteraciones y, además, vamos a ajustar la visión global del proyecto a objetivos más cortos, de tal forma que iremos realizando entregas acordes a esos objetivos y que se

irán moldeando a lo largo del tiempo. En otras palabras, iteramos e incrementamos con una periodicidad mucho más corta y elegimos en qué partes aportamos qué nivel de detalle. (Herrera, 2020, https://bit.ly/3BNFAlH)

Figura 14: Iteraciones



Fuente: Herrera, 2020, https://bit.ly/3BNFAlH

Para adaptarse rápidamente al mercado, la colaboración entre las personas es esencial para identificar aquellas funcionalidades que permitan potenciar el posible valor añadido y, en consecuencia, la capacidad para monetizar una característica del producto. Es decir, iteramos e incrementamos con valor. Pero ¿esto qué significa?

- Iteramos para buscar una mejor solución a través del *feedback* de los usuarios, no solo para buscar un cambio o modular la frecuencia con la que sucede. Con un desarrollo iterativo, construimos añadiendo nueva

funcionalidad incrementalmente e iteramos para mejorar, cambiar o suprimir funcionalidad existente.

- Añadimos *software* incrementalmente para gradualmente construir funcionalidad. No obtenemos ROI hasta que el usuario empieza a consumir lo que hemos construido. Por tanto, el valor de negocio hasta ese momento es una estimación, al igual que lo es el desarrollo del *software*. (Herrera, 2020, https://bit.ly/3BNFAlH)

Referencias

40 de Fiebre, (s.f.). *¿Qué es el A/B Testing?* https://www.40defiebre.com/que-es/ab-test

Aden, (s.f.). *Metodologías ágiles: ¿Qué son y cuáles son las más utilizadas?* https://blog.aden.org/metodologias-agiles-que-son-y-cuales-son-las-mas-utilizadas

Ángel, M. (2022). Scrum: qué es y cómo funciona este marco de trabajo. *We Are Marketing*. https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html#:~:text=Scrum%20es%20un%20framework%20que,valor%20posible%20productiva%20y%20creativamente.

Cyberclick, (s.f.). *Email Marketing. ¿Qué es el email marketing? Definición y cómo diseñar una estrategia efectiva*. https://www.cyberclick.es/marketing/email-marketing#:~:text=Se%20ha%20predicho%20su%20decaimiento,anunciantes%20en%20t%C3%A9rminos%20de%20retorno

Deloitte, (2021). *Cuál es la diferencia entre Agile, Lean Startup y Design Thinking*. https://www2.deloitte.com/es/es/blog/todo-tecnologia/2021/diferencias-agile-lean-startup-design-thinking.html

Gómez, J. L. (2016). Growth Hacking 3: Adquisición de usuarios. *Sarenet*. https://blog.sarenet.es/growth-hacking-adquisicion-usuarios/

Herrera, J. (2020). ¿Por qué iterativo e incremental en productos digitales? *Paradigma Digital*. https://www.paradigmadigital.com/techbiz/iterativo-e-incremental-en-productos-digitales/

Ilab, (2020). *¿Qué es iterar y por qué las empresas deberían implementarlo?* https://ilab.net/que-es-iterar-y-por-que-las-empresas-deberian-implementarlo/

León, M. (2022). Embudo de ventas: qué es y cómo aplicarlo a tu negocio. *Mi Tienda Nube*. https://www.tiendanube.com/blog/que-es-el-embudo-de-ventas/

Lipúzcoa, J. (s.f.). Framework para desarrollar una cultura de experimentación. *Product Hackers*. https://producthackers.com/es/blog/framework-cultura-experimentacion

Manifiesto Ágil, (s.f.). *Principios del Manifiesto Ágil*. https://agilemanifesto.org/iso/es/principles.html

Merryweather, E. (2020). La diferencia: Agile vs. Scrum. *Product School*. https://productschool.com/blog/product-management-2/difference-agile-vs-scrum/

Pérez Prado, J. M. (s.f.). *Experimentación en equipos de producto*. https://es.slideshare.net/Product_Hackers/experimentacin-en-equipos-de-producto

Redacción APD, (2021). *¿En qué consiste la metodología Kanban y cómo utilizarla?* https://www.apd.es/metodologia-kanban/

Rojas, K. (2022). ¿Qué es growth marketing? *Mi Tienda Nube*. https://www.tiendanube.com/blog/growth-marketing/

Sosa, P. (2019). La iteración en el diseño de productos. *Medium*. https://medium.com/@slikmx/la-iteraci%C3%B3n-en-el-dise%C3%B1o-de-productos-50e47af7382a

Tena, M. (s.f.). *¿Qué es la metodología 'agile'? BBVA*. https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/

VWO, (s.f.). *Guía Sobre A/B Testing*. https://vwo.com/es/ab-testing/